Signal, Image Processing and Embedded Systems

David C. Wyld,
Dhinaharan Nagamalai (Eds)

Computer Science & Information Technology

- 8th International Conference on Signal, Image Processing and Embedded Systems (SIGEM 2022)
- 3rd International Conference on Machine Learning Techniques (MLTEC 2022)
- 11th International Conference on Software Engineering and Applications (SEAPP 2022)
- 8th International Conference on Information Technology Converge Services (ITCON 2022)
- 8th International Conference on Natural Language Computing (NATL 2022)
- 8th International Conference on Fuzzy Logic Systems (Fuzzy 2022)
- 8th International Conference on Computer Science, Engineering And Applications (CSEA 2022)

Published By

AIRCC Publishing Corporation

Volume Editors

David C. Wyld,
Southeastern Louisiana University, USA
E-mail: David.Wyld@selu.edu

Dhinaharan Nagamalai (Eds),
Wireilla Net Solutions, Australia
E-mail: dhinthia@yahoo.com

ISSN: 2231 - 5403
ISBN: 978-1-925953-80-0
DOI: 10.5121/csit.2022.122001 - 10.5121/csit.2022.122016

Typesetting: Camera-ready by author, data conversion by NnN Net Solutions Private Ltd., Chennai, India

Preface

8th International Conference on Signal, Image Processing and Embedded Systems (SIGEM 2022), November 19-20, 2022, Zurich, Switzerland, 3rd International Conference on Machine Learning Techniques (MLTEC 2022), 11th International Conference on Software Engineering and Applications (SEAPP 2022), 8th International Conference on Information Technology Converge Services (ITCON 2022), 8th International Conference on Natural Language Computing (NATL 2022), 8th International Conference on Fuzzy Logic Systems (Fuzzy 2022), 8th International Conference on Computer Science, Engineering And Applications (CSEA 2022) was collocated with 8th International Conference on Signal, Image Processing and Embedded Systems (SIGEM 2022). The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The SIGEM 2022, MLTEC 2022, SEAPP 2022, ITCON 2022, NATL 2022, FUZZY 2022, and CSEA 2022. Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically.

In closing, SIGEM 2022, MLTEC 2022, SEAPP 2022, ITCON 2022, NATL 2022, FUZZY 2022 and CSEA 2022 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. The book is organized as a collection of papers from the SIGEM 2022, MLTEC 2022, SEAPP 2022, ITCON 2022, NATL 2022, FUZZY 2022 and CSEA 2022

We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research. It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

David C. Wyld,
Dhinaharan Nagamalai (Eds)

General Chair

Organization

David C. Wyld, Southeastern Louisiana University, USA
Dhinaharan Nagamalai (Eds) Wireilla Net Solutions, Australia

Program Committee Members

A. Raja Basha, K. L. University, India
A. S. M. Sanwar Hosen, Jeonbuk National University, South Korea
A.Ramesh babu, Ramco Institute of Technology, India
Abdel-Badeeh M. Salem, Ain Shams University, Egypt
Abdelhadi Assir, Hassan 1st University, Morocco
Abdelhameed Ibrahim, Mansoura University, Egypt
Abderrahmane Ez-Zahout, Mohamed V University, Morocco
Abdulraqeb Alhammadi, Universiti Teknologi Malaysia, Malaysia
Abhilash, Cyrielle Castle, India
Abhishek Chakraborty, University of Calcutta, India
Addisson Salazar, Universitat Politècnica de València, Spain
Adil Bashir, Islamic University of Science and Technology, India
Adil Bouhouch, Mohammed V University, Morocco
Adrian Olaru, University Politehnica of Bucharest, Romania
Afaq Ahmad, Sultan Qaboos University, Oman
Ahmad Fakharian, Islamic Azad University, Iran
Ajay Anil Gurjar, Sipna College of Engineering and Technology, India
Ajay Kumara, Lenoir-Rhyne University, USA
Ali Asif, Harbin Engineering University, China
Ali H. Wheeb, University of Baghdad, Iraq
Alireza Valipour Baboli, University Technical and Vocational, Iran
Allel Hadjali, LIAS/ENSMA, France
Amal Azeroual, Mohammed V University, Morocco
Amando P. Singun Jr, University of Technology and Applied Sciences, Oman
Amel Ourici, University Badji Mokhtar Annaba, Algeria
Amine Khaldi, Universite Kasdi Merbah Ouargla, Algeria
Ammar Almasri, Al-Balqa Applied University, Jordan
Ana Luisa V. Leal, University of Macau, China
Anala M R, RV College of Engineering, India
Anita Dixit, SDM College of Engineering and Technology, India
Anjan Kumar, GLA University, India
Anouar Abtoy, Abdelmalek Essaadi University, Morocco
Anuj Singal, GJU S &T, HISAR, India
Archit Yajnik, Sikkim Manipal Institute of Technology, India
Aridj Mohamed, Hassiba Benbouali University Chlef, Algeria
Artika Singh, NMIMS University, India
Arun Malik, Lovely Professional University, India
Aruna Kumar S V, Malnad College of Engineering, India
Aruna Pavate, St. Francis Institute of Technology, India
Arunkumar B R, BMS Institute of Technology and Management, India
Ashkan Ebadi, National Research Council Canada, Canada
Asif Irshad Khan, King Abdulaziz University, Saudi Arabia
Assia Djenouhat, University of Algiers 3, Algeria
Atlas Haddadi Avval, Mashhad University of medical sciences, Iran

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Artificial Intelligence Community (AIC)

Soft Computing Community (SCC)

Digital Signal & Image Processing Community (DSIPC)

8th International Conference on Signal, Image Processing and Embedded Systems (SIGEM 2022)

3rd International Conference on Machine Learning Techniques (MLTEC 2022)

11th International Conference on Software Engineering and Applications (SEAPP 2022)

8th International Conference on Information Technology Converge Services (ITCON 2022)

8th International Conference on Natural Language Computing (NATL 2022)

8th International Conference on Fuzzy Logic Systems (Fuzzy 2022)

8th International Conference on Computer Science, Engineering and Applications (CSEA 2022)

CLASSIFICATION OF VIRAL, BACTERIAL, AND COVID-19 PNEUMONIA USING DEEP LEARNING FRAMEWORK FROM CHEST X-RAY IMAGES

Muhammad E. H. Chowdhury, Tawsifur Rahman,
Amith Khandakar and Sakib Mahmud

Department of Electrical Engineering, Qatar University, Doha, Qatar

ABSTRACT

The novel coronavirus disease (COVID-19) is a highly contagious infectious disease. Even though there is a large pool of articles that showed the potential of using chest X-ray images in COVID-19 detection, a detailed study using a wide range of pre-trained convolutional neural network (CNN) encoders-based deep learning framework in screening viral, bacterial, and COVID-19 pneumonia are still missing. Deep learning network training is challenging without a properly annotated huge database. Transfer learning is a crucial technique for transferring knowledge from real-world object classification tasks to domain-specific tasks, and it may offer a viable answer. Although COVID-19 infection on the lungs and bacterial and viral pneumonia shares many similarities, they are treated differently. Therefore, it is crucial to appropriately diagnose them. The authors have compiled a large X-ray dataset (QU-MLG-COV) consisting of 16,712 CXR images with 8851 normal, 3616 COVID-19, 1485 viral, and 2740 bacterial pneumonia CXR images. We employed image pre-processing methods and 21 deep pre-trained CNN encoders to extract features, which were then dimensionality reduced using principal component analysis (PCA) and classified into 4-classes. We trained and evaluated every cutting-edge pre-trained network to extract features to improve performance. CheXNet surpasses other networks for identifying COVID-19, Bacterial, Viral, and Normal, with an accuracy of 98.89 percent, 97.87 percent, 97.55 percent, and 99.09 percent, respectively. The deep layer network found significant overlaps between viral and bacterial images. The paper validates the network learning from the relevant area of the images by Score-CAM visualization. The performance of the various pre-trained networks is also thoroughly examined in the paper in terms of both inference time and well-known performance criteria.

KEYWORDS

Novel Coronavirus disease, COVID-19, viral pneumonia, bacterial pneumonia, deep learning, Convolutional neural network, Principal component analysis.

1. INTRODUCTION

The COVID-19 pandemic struck the world, and it has severely overrun healthcare systems worldwide. It had an impact on social, economic, and all facets of human life [1, 2]. As of August 2022, there were more than 6 million fatalities and more than 579 million active cases worldwide [3]. Reverse transcription-polymerase chain reaction (RT-PCR), which recognizes viral nucleic acid, is the gold standard for COVID-19 diagnosis. Low viral load and sample

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 01-21, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122001

mistakes might lead to inaccurate RT-PCR results [4, 5]. Antigen testing is quick but not particularly accurate [6-8].

Therefore, it has become necessary to search for more readily available, dependable, and easily accessible diagnostic equipment. The authors have used deep learning in a variety of areas, including food [9-11], renewable energy [12, 13], education [14], communication [15, 16], and others. The authors investigated the use of machine learning for biomedical solutions, such as reliable heart sound diagnosis in smart digital stethoscopes [17], real-time heart attack detection in reducing road accidents [18], for estimating blood pressure from Photoplethysmogram signal and demographic features [19], etc. The current advances in artificial intelligence have been a lifesaver in various biomedical abnormalities detection [20]. Radiological tests can be useful in the diagnosis and assessment of disease progression by assessing the severity of pneumonia because pneumonia has been found in the majority of COVID-19 patients. As a result, typical diagnosis methods for COVID-19 include Chest X-ray (CXR) and Chest computed tomography (CT)-scan [21-23]. The procedures used in CT scans can be costly and occasionally readily can contaminate the equipment due to the nature of this contagious disease, endangering the safety of the subsequent patients. The American College of Radiology does not endorse it either [24, 25]. However, X-ray machines are more readily accessible, more affordable, and portable (as opposed to CT machines), making them a more affordable option for treating lung-related conditions. The impact of Chest X-rays and artificial intelligence on lung-related disorders including tuberculosis [26], pneumonia [27], and even COVID-19 [22, 23, 28, 29] has been the subject of numerous investigations by the authors. Portable tools used in solitary spaces can also help to lower the risk of infection [30-32]. Therefore, if artificial intelligence (AI) on CXR can be made more dependable with the aid of more research, it would be a more cost-effective solution.

Convolutional neural networks (CNNs), one type of deep learning artificial intelligence network, need large training data. Unfortunately, in the initial phase of the pandemic, CXR images are scarce for deep neural network training [33-36]. It is challenging to gather enough information from the small number of CXR images. In several investigations, it was suggested that increasing synthetic images for training might lessen the drawbacks. CovidGAN, an Auxiliary Classifier Generative Adversarial Network (ACGAN) based model, was utilized by Wang et al. in [37] to generate artificial data. A dataset of 403 COVID-CXR images and 721 normal images was used for their investigation. The accuracy of CNN's COVID-19 detection (using CXRs) has increased from 85% to 95%; thanks to the use of CovidGAN to generate synthetic data [37]. To achieve an accuracy of 96.58 percent, Chowdhury et al. in [38] built a unique framework called PDCOVIDNet using a dilated convolution in the parallel stack. Decompose, transfer, and compose (DeTraC) is a network proposed by Abbas et al. in [39] that checked anomalies by examining class borders of the images and reported an accuracy of 93.1 percent and sensitivity of 100 percent in COVID-19 detection. In [35], Wang and Wong proposed a deep COVID19 detection (COVID-Net) model, which classified normal, non-COVID pneumonia, and COVID-19 groups with 92.4 percent accuracy.

Some studies looked into how well learning of CNN models can be transferred to CXRs related learning. Transfer learning using CNN models has significantly helped in the process by using pre-trained networks for the task of differentiating between CXR images with normal and COVID-19-affected CXR by varying parameters like weights and biases of the pre-trained model. Azemin et al. in [40] used the ResNet-101 model to stratify COVID-19 with an accuracy of only 71.9 percent while Khan et al. [41] investigated a couple of pre-trained deep learning models like ResNet50, VGG16, VGG19, and DensNet121 and found that VGG16 and VGG19 had the highest performance with 99.3 percent accuracy in COVID-19 detection. The pre-trained networks AlexNet, GoogLeNet, and ResNet18 were examined by Loey et al. in [42] using a

dataset of 307 images divided into four classes: COVID-19, normal, pneumonia bacterial, and pneumonia virus. According to the authors, GoogLeNet achieved 99.9% validation accuracy and 100% testing accuracy. However, the study was carried out on a very small dataset and therefore, the performance reported in this study cannot be generalized on a large dataset.

Using a modified version of VGG-16, Brunese et al. [43] analyzed a dataset of 6,523 chest X-rays from patients with COVID-19, other lung illnesses, and healthy individuals. A 97 percent accuracy rate has been reported. In [44], the authors utilized the Xception model to categorize pneumonia, positive COVID-19, and negative COVID-19 on a dataset acquired from [45]. Training accuracy was stated to be 99.5% while the testing accuracy was 97.4%. Using a COVID-19 dataset made up of 2,951 CXR images and annotated ground-truth infection segmentation masks, Degerli et al. developed a novel method for COVID-19 infection map development in [46]. On the generated dataset, several encode-decoder (E-D) CNNs were trained and tested, with the best network achieving an F1 score for infection localization of 85.81 percent.

Deep CNN models were employed by Chowdhury et al. in [28] to classify images of normal, viral pneumonia, and COVID-19 chest X-rays into binary and three-class categories. Transfer learning was investigated on the generated dataset using pre-trained Squeezenet, Mobilenetv2, Inceptionv3, CheXNet, ResNet, and Densenet201 models. Three-class classification tasks produced an accuracy score of 97.9 percent while binary classification had a score of 99.7 percent. However, the study did not include bacterial pneumonia which is often challenging to classify using CXR images. Moreover, the dataset size was much smaller compared to the current study. Additionally, few deep learning models were investigated in this and other similar transfer learning-based works.

To the best of the authors' knowledge, an in-depth analysis of the performance of a large pool of state-of-the-art transfer learning models has not yet been demonstrated, even though a lot of research has been done on the use of artificial intelligence for COVID-19 detection from CXR images. The research community will benefit from this paper's addition to the body of knowledge about transfer learning's role in COVID-19 detection because it will help them to decide whether to deploy a particular network given that networks vary not only in terms of performance but also in terms of size, parameters, and inference time. These will be covered in the paper's remaining section. The network's performance's dependability as a visualization tool is tested, and since it uses one of the largest datasets accessible, it can also be regarded as trustworthy. The next sections of the chapter are organized as follows: Section II explored the technique in detail and gave a description of each transfer learning model, Section III presented the results and analyses and discussed them, and Section IV reported the conclusion.

2. METHODOLOGY

Figure 1 depicts the details of the proposed methodology used in this investigation. As can be observed, the authors investigated how well state-of-the-art pre-trained CNN models perform in useful feature extraction as a CNN encoder and then principal component analysis (PCA) was used to reduce the dimensionality to avoid overfitting of the classifier.

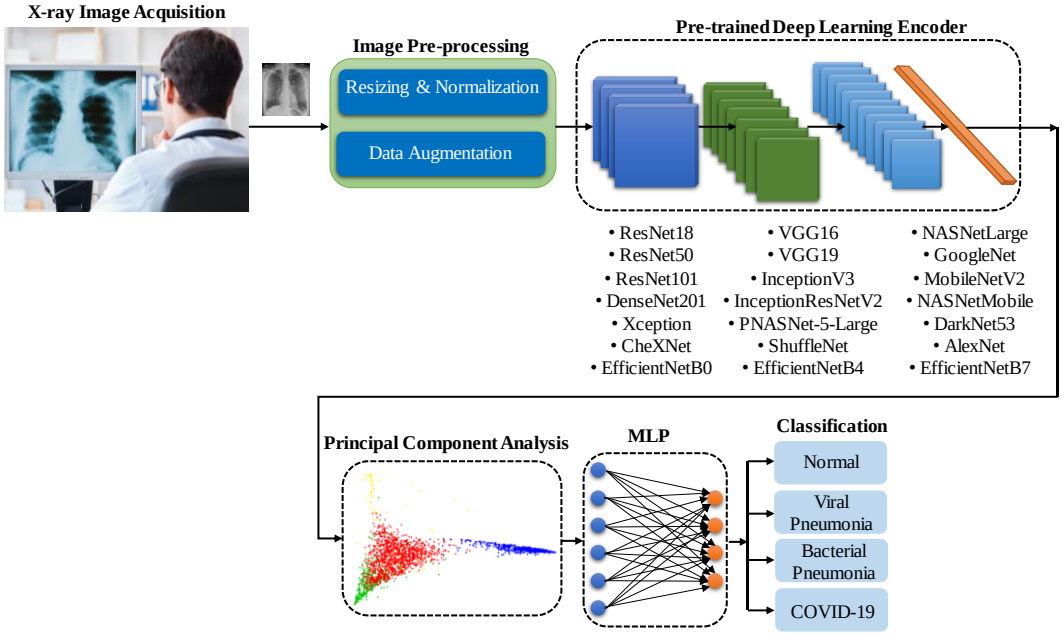


Figure 1. Framework of the proposed methodology.

Finally, a 3-layer multilayer-perceptron (MLP) classifier is used in identifying a dataset made up of COVID-19, Bacterial Pneumonia, Viral Pneumonia, and Normal CXR images. In MLP, one input, one hidden layer and one output layer, where the rectilinear unit (ReLU) activation function is used in the input and hidden layers and sigmoid function is used in the output and Interpretable maps are used afterwards to describe the usefulness and reliability of the proposed framework. This section will go into detail on the cutting-edge pre-trained model that was investigated in this study, the specifics of the dataset that was used, as well as the pre-processing and training aspects.

2.1. Deep Learning Pre-Trained Models

Several pre-trained CNN models were trained on a very large dataset, ImageNet, which has demonstrated cutting-edge performance [47]. These pre-trained networks have been shown to perform consistently well in different computer vision problems and can be trained on new databases to adjust their weights and biases for that dataset or application. The introduction section already showed examples of such applications. Network depth, which is the maximum number of consecutive convolutional layers of fully connected layers along the path from the input layer to the output layer, varies between these networks in terms of size and the number of parameters in millions. The parameters utilized in the investigation are described in detail in Table 1.

Table 1 compares the several cutting-edge models that were used in the study.

Network	*Network Depth*	*Parameters [Millions]*
ResNet18	72	11.7
ResNet50	107	25.6
ResNet101	209	44.7
VGG16	16	138.4
VGG19	19	143.7
InceptionV3	189	23.9
InceptionResNetV2	449	55.9
NASNetLarge	533	88.9
PNASNet-5-Large	*	86.1
Xception	81	22.9
CheXNet (DenseNet 121)	242	8.1
DenseNet201	402	20.2
ShuffleNet	50	1.4
GoogLeNet	22	7
MobileNetV2	105	3.5
NASNetMobile	132	5.3
DarkNet53	53	41
AlexNet	8	61.1
EfficientNetB0	132	5.3
EfficientNetB4	258	19.5
EfficientNetB7	438	66.7

*Values are not known

All the experiments were carried out in Python by importing all the models from the Pytorch library in Google ColabPro. Through transfer learning, specific characteristics from the X-ray images of the COVID-infected pneumonia patients were extracted using the rich set of features that these networks had learned from the ImageNet dataset. To categorize the X-ray images into one of the following classes: Normal, Bacterial Pneumonia, Viral Pneumonia, and COVID-19, the dense layers of each network were dropped and after flattening features were extracted from each CNN encoder. The dimensionality of the feature vector was reduced using principal component analysis (PCA) with 90% variance to avoid over-fitting of the models on a comparatively small dataset, then the multi-layer perceptron (MLP) model with SoftMax layer with four neurons was used as a classifier to classify the 4-class problem. Below are further specifics about the networks used in this study:

AlexNet - In the AlexNet, convolution processes were performed several times between max-pooling operations, which allows the network to acquire richer features at all spatial scales. AlexNet placed first with a top-5 test error rate of 15.3% [48] in the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

VGG - VGG uses a standard CNN design and simply varies the depth: one network has 11 weight layers (8 convolutional and 3 fully connected layers), while another network has 19

weight layers (16 convolutional and 3 fully connected layers). Convolutional layers have a relatively small width (number of channels), starting at 64 in the first layer and rising by a factor of 2 after each max-pooling layer to reach 512 [49].

DarkNet 53 - Like the VGG models, it primarily makes use of filters and doubles the number of channels after each phase of pooling. Batch normalization is used to regularize the model batch, speed up convergence, and stabilize training. It serves as the foundation for the well-known localization network, YOLOv3 [50].

ResNet - Overfitting, a well-known paradigm for deep networks trained on small datasets, can significantly reduce the generalization performance. When a lot of training epochs are run, the "vanishing gradient" problem eventually leads to network saturation, especially at the initially hidden layers, making the problem worse. By incorporating the idea of shortcut connections, where the activations of one layer that are given to the next layer are fed to the deeper layers as well, which is the core concept of a residual network (ResNet). This solves the vanishing gradient problem with deep CNN networks. ResNet is made up of 8 residual blocks, each of which has two convolutional layers with three kernels on each layer [51] and it has a couple of variants: ResNet 18, 50, 101, and 152. Going farther into the network causes the layer depth to rise every two blocks, with layer sizes of 64, 128, 256, and 512 kernels, respectively. In addition, a 7×7 Conv layer is utilized in the network's beginning, followed by a pooling layer of stride 2, and a SoftMax classification layer at its end.

GoogleNet - The classification of many types of problems performed better with GoogLeNet-Inception networks. Smaller kernels are typically preferred for an area-specific feature that is dispersed over an image frame, whereas larger kernels are typically selected for global characteristics that are distributed over a vast region of images. This gave rise to the concept of inception layers, where kernels of various sizes - such as 1×1, 3×3, and 5×5 were combined within the same layer rather than moving further into the network [52, 53]. The Inception network begins with several traditional layers of 3×3 kernel, and 3 inception blocks, and culminates with an 8×8 global average pooling layer, followed by a SoftMax classifier. This architecture expands the network space where training can choose the best features.

InceptionV3 - This network suggested several improvements over version InceptionV1, which improved accuracy and decreased computational cost. High-quality networks can be trained on relatively small training sets thanks to the combination of a decreased parameter count, extra regularization, batch-normalized auxiliary classifiers, and label smoothing [54].

InceptionResNetV2 – It is the result of combining the most recent revision of the Inception architecture [54] with the residual connections, reported in [51]. It was claimed that the combination would keep the computational efficiency of the Inception network architecture while gaining all the advantages of the residual technique. Although it is a more expensive hybrid version of Inception, it has been demonstrated to have better performance [55].

Xception - In neural computer vision architectures, this network substitutes depthwise separable convolutions for Inception modules. The feature extraction base of the network in the Xception architecture is composed of 36 convolutional layers. The Xception architecture is a linear stack of residually connected depthwise separable convolution layers [56].

DenseNet - Contrary to residual networks, DenseNet concatenates all feature maps as opposed to simply adding up residuals [57]. All layers inside a thick block are closely connected to one another, allowing for more monitoring between levels. The four dense blocks that make up

DenseNet, each have numerous convolution layers with 1×1 and 3×3 filters. Transition layers made up of a batch normalization layer, a 1×1 convolutional layer, and a 2×2 average pooling layer are used to divide the dense blocks. The network begins with a 7×7 convolutional layer, then moves on to a 3×3 max-pooling layer, both with a stride of 2 and concludes with a 7×7 global average pooling layer, then a SoftMax layer. DenseNet has several variants, such as DenseNet121, DenseNet169, and DenseNet201.

ChexNet - One of the largest publicly accessible Chest X-ray datasets is ChestX-ray14 [58], which has over 100,000 frontal view X-ray images with 14 disease classes. CheXNet [59] is a DenseNet (DenseNet121) model which was re-trained on this dataset. Therefore, CheXNet is the only pre-trained network, which is already trained on a large X-ray dataset, unlike other pre-trained models.

MobileNetV2 – There are real-time applications like robots, self-driving cars, augmented reality, etc., which need compact networks (lightweight networks). MobileNet is built on a simplified architecture that uses depthwise separable convolutions to construct compact deep neural networks. It was created for mobile and embedded vision applications. The model builder can select the appropriate model size for their application based on the constraints of the problem using two straightforward global hyperparameters (width multiplier and resolution multiplier) [60]. For MobileNet, Depthwise Separable Convolutions are a crucial component. The MobileNetV2 design starts with a fully convolutional layer with 32 filters, followed by 19 bottleneck layers with residual connections between point wise convolutional layers. Except for the last fully connected layer, which has no nonlinearity and feeds into a SoftMax layer for classification, all Conv layers are followed by batch normalization and rectilinear unit (ReLU) nonlinear activation function [61].

ShuffleNet - The architecture of ShuffleNet makes use of two novel operations – point wise group convolution and channel shuffle - to significantly lower computation costs while retaining accuracy, it was also created with mobile phone deployment in mind. It has been improved to achieve lower complexity and is based on Residual network design [62].

NASNet - The NASNetLarge and NASNetMobile models' generalization has been greatly improved thanks to a new regularization method called ScheduledDropPath [63].

EfficientNet - Tan et al. presented EfficientNet in [64], which, in contrast to existing CNN scaling algorithms that use one-dimension scaling, balances the network's width, depth, and resolution.

2.2. Description of the Database

The authors have compiled a large dataset called, QU-MLG-COV in this study, which consists of 16,712 CXR images with 8851 normal, 3616 COVID-19, 1485 viral pneumonia, and 2740 bacterial pneumonia CXR images. The authors created this dataset by using and modifying various open-access databases for four different types of CXR images (COVID-19, normal (healthy), viral pneumonia, and bacterial pneumonia). The QU-MLG-COV dataset merged the COVID-19 dataset, the CXR dataset from the Radiological Society of North America (RSNA) [65], and the Chest X-Ray Images (Pneumonia) Kaggle dataset.

COVID-19 dataset

COVID-19 CXR images that make up the COVID-19 dataset were gathered from several publically accessible datasets,

have various lung abnormalities. We have taken 8851 healthy (normal) CXR images from the RSNA dataset for this study. Radiologists with the necessary training assessed the CXRs in the dataset, and clinical history, vital signs, and laboratory tests were used to confirm online sources and published studies. A total of 3616 X-ray images were collected; 2473 of them came from the BIMCV-COVID19+ dataset [66], 183 from a German medical school [67], 559 from the Italian Society of Medical Radiology (SIRM), GitHub, Kaggle, and Twitter [68-71], and 400 from another COVID-19 CXR repository [72]. The BIMCV-COVID19+ dataset, which includes 2473 CXR pictures of COVID-19 patients obtained from digital X-ray (DX) and computerized X-ray (CX) equipment, is the single largest available dataset.

RSNA Chest X-ray dataset

About 26,684 CXR DICOM images make up the RSNA pneumonia detection challenge dataset [65], of which 8851 images are normal, and 17842 images are the condition. To categorize the CXR images into healthy control (normal) and lung infections, they were connected with clinical symptoms and history.

Chest X-Ray Images (Pneumonia)

On Kaggle, 5824 chest X-ray images of bacterial, viral, and normal pneumonia were found with resolutions ranging from 400p to 2000p. Out of 5824 chest X-ray images, 2760 images with bacterial pneumonia and 1485 with viral pneumonia are used in this study. Figure 2 provides some examples of the Chest X-Ray images used in this investigation.

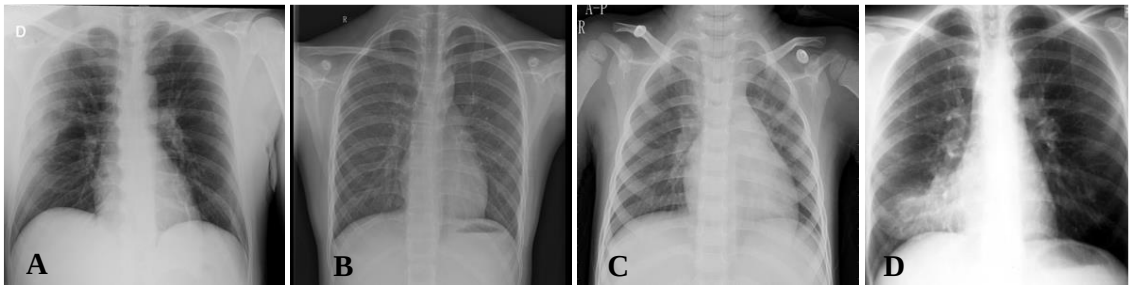


Figure 2. Sample CXR images from the dataset for COVID-19 (A), Normal (B), Viral Pneumonia (C), and Bacterial Pneumonia (D).

2.3. Experimental Setup

The dataset's distribution of labelled images across classes was unbalanced, which could have influenced training results. Data augmentation is a well-liked remedy for this unbalanced dataset [20, 27, 28], which may also be applied to expand the dataset because CNN models learn best from large databases. With 80% of the data used for training and 20% of the data being unseen for testing, the performance of the experiment conducted in this study was evaluated using five-fold cross-validation. Additionally, to prevent overfitting, 20% of the training data are used as a validation set.

Table 2. Before and after data augmentation, the number of photos per class and per fold Class

Class	*# of Samples*	*Training Samples*	*Augmented Training Samples*	*Validation Samples*	*Test Samples*
COVID-19	3616	2893	2314×2=4628	579	723
Normal	8851	7081	5665	1416	1770
Viral	1485	1188	950×6 = 5700	238	297
Bacterial	2760	2208	1766×3 = 5298	442	552

Data augmentation was used to balance the dataset by applying rotations of 5 and 10 degrees to avoid unrealistic rotations of the images. Additionally, image translations in both the horizontal and vertical directions were applied within the range [-0.15, +0.15], as shown in Figure 3. The number of CXR images per class utilized for training, validation, and testing at each fold is listed in Table 2.

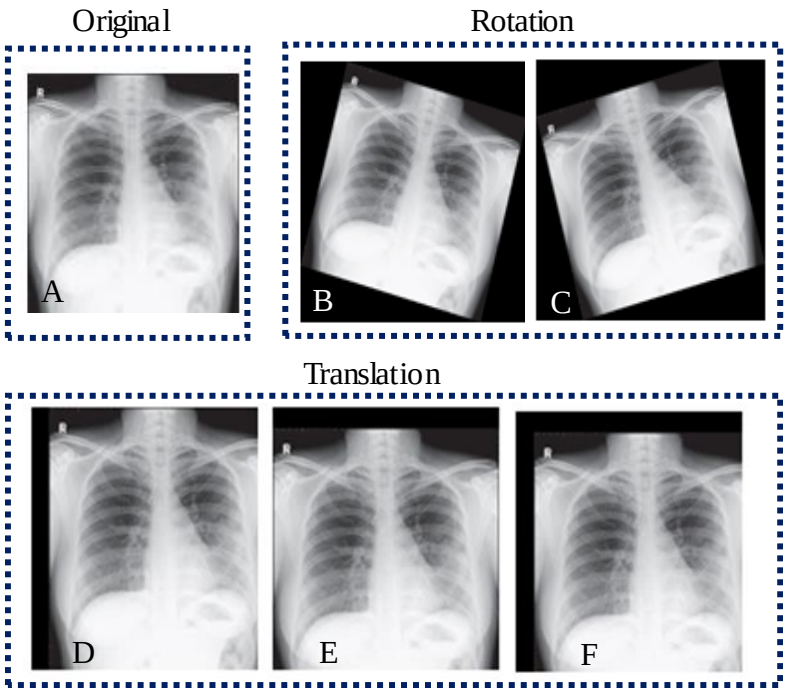


Figure 3. Image augmentation: clockwise and anticlockwise rotation and horizontal and vertical translation.

The experiment was conducted on ColabPro using the Pytorch library. Table 3 displays the training experiment's specifics. To create the final receiver operating characteristic (ROC) curve, confusion matrix, and evaluation matrices, a fivefold cross-validation result was averaged.

Table 3. Details of the experiment hyper-parameters

Training parameter	*Value*
batch size	16
learning rate	0.001
epochs	15
epochs patience	8
stopping criteria	8
Loss function	BCE
optimizer	ADAM

Four evaluation metrics - Accuracy, Precision, Sensitivity, and F1-score with 95 percent confidence intervals (CIs) - were used to evaluate the performance of the deep CNN-based encoders. The overall confusion matrix, which compiles all test fold results from the 5-fold cross-validation, was used to derive per-class values.

$$Accuracy_{class\_i} = \frac{TP_{class\_i} + TN_{class\_i}}{TP_{class\_i} + TN_{class\_i} + FP_{class\_i} + FN_{class\_i}} \quad (1)$$

$$Precision_{class\_i} = \frac{TP_{class\_i}}{TP_{class\_i} + FP_{class\_i}} \quad (2)$$

$$Sensitivity_{class_i} = \frac{TP_{class_i}}{TP_{class_i} + FN_{class_i}} \quad (3)$$

$$F1\_score_{class_i} = 2\frac{Precision_{class_i} \times Sensitivity_{class_i}}{Precision_{class_i} + Sensitivity_{class_i}} \quad (4)$$

where $class_i = COVID-19, normal, viral\ and\ bacterial.$

The weighted average values of each class were used to calculate the overall performance. Since class frequencies differ for the given task, the weighted average provides a better indication of overall performance.

$$Precision = \frac{n1(Precision_{COVID}) + n2(Precision_{normal}) + n3(Precision_{viral}) + n4(Precision_{bacterial})}{n1 + n2 + n3 + n4} \quad (5)$$

$$Sensitivity = \frac{n1(Sensitivity_{COVID}) + n2(Sensitivity_{normal}) + n3(Sensitivity_{viral}) + n4(Sensitivity_{bacterial})}{n1 + n2 + n3 + n4} \quad (6)$$

$$F1\_score = \frac{n1(F1\_score_{COVID}) + n2(F1\_score_{normal}) + n3(F1\_score_{viral}) + n4(F1\_score_{bacterial})}{n1 + n2 + n3 + n4} \quad (7)$$

$$Accuracy = \frac{n1(Accuracy_{COVID}) + n2(Accuracy_{normal}) + n3(Accuracy_{viral}) + n4(Accuracy_{bacterial})}{n1 + n2 + n3 + n4} \quad (8)$$

Where $n1, n2, n3\ and\ n4$ are the total number of COVID-19, normal, viral, and bacterial cases respectively.

The authors also used a different metric, known as inference time, $\delta(t)$, to measure how long it took the framework to classify the image. The trade-off between accuracy and decision-making time can also be better understood by plotting inference time against the F1 score. To ensure that the best-performing network is picking up knowledge from important areas of the image, the authors have additionally verified the validity of the best-trained network using the well-known Score-CAM visualization technique [73].

3. RESULTS AND DISCUSSION

Table 3 shows the overall performance of the various pre-trained network-based encoders in descending order of the F1-score. As can be observed, the performance of all different CNN encoder-based frameworks is comparable. It demonstrates that all networks, regardless of size, perform well in classifying the CXR images into various classes. This merely verifies how well pre-trained models on CXRs are, as described in numerous earlier literature. It is also noteworthy that EfficientNet B7 has the longest inference time—roughly 42.8 milliseconds—while AlexNet has the shortest—roughly 1.8 milliseconds. However, the best performing model is CheXNet, which is the only network pre-trained on a large CXR dataset along with already trained on ImageNet. There is a clear performance gap of more than 2% compared to other pre-trained models which are only trained on ImageNet. This highlights the importance of retraining the pre-trained model on domain data, which can improve the model's performance.

The best option initially can be appeared to be ResNet18 and MobileNetV2 because of their excellent performance and quick inference times. The greatest option for even deploying for mobile and smart devices would be MobileNetV2. CheXNet is DenseNet121 trained on large CXR images and performed better than the DenseNet201 variant of DenseNet. This also clearly reflects the importance of domain knowledge of the model during re-training. However, it is worth mentioning that the pre-trained model-based framework overall performs close to each other as they are already trained on a large image database and these models are very good at extracting useful image features.

Table 3. Overall network performance metrics in descending order of F1-Score.

Encoders	***Inference time $\delta(t)$(ms)***	***Accuracy (%)***	***Precision (%)***	***Sensitivity (%)***	***F1-score (%)***
CheXNet (DenseNet 121)	17.6	96.7	96.51	97.18	96.81
EfficientNet B0	10.7	94.24	94.32	94.42	94.18
ResNet 18	5.5	93.72	93.76	93.54	93.99
MobileNet V2	2.5	93.19	93.29	93.07	92.99
DenseNet201	25.8	93.02	92.67	93.07	92.99
InceptionV3	23.8	92.52	92.26	92.52	92.3
ResNet 101	10.8	92.81	91.45	91.85	91.45
ResNet 50	6.1	91.71	91.23	91.71	91.35
EfficientNet B7	42.8	91.47	91.47	91.47	91.47
VGG 16	11	91.42	91.12	91.42	91.22
AlexNet	1.8	91.16	91.06	91.37	91.16
PNASNET-5-Large	28.8	91.15	91.04	91.15	91.09
EfficientNet B4	28.7	90.96	90.86	90.96	90.91
GoogLeNet	3.3	90.03	90.79	90.03	90.89
Darknet53	7.2	90.88	90.56	90.88	90.66
Xception	4.6	90.96	90.52	90.96	90.62
VGG 19	12.1	90.91	90.47	90.92	90.61
NASNetLarge	30.1	89.66	89.58	89.58	89.57
ShuffleNet	6.6	89.77	89.46	89.77	89.51
InceptionResNet	29.8	88.07	88.78	88.07	88.89
NASNetMobile	4.8	87.24	87.11	87.25	87.17

Table 4 shows the class-wise performance comparison of the top performing 5 networks, namely, CheXNet, EfficientNet B0, ResNet18, MobileNetV2, and DenseNet201. It is further confirmed by the class-specific performance that CheXNet outperforms all other models as it performs well not only overall but also in classes. It is also evident from the table that the networks may become perplexed when attempting to differentiate between viral and bacterial pneumonia CXRs. This is supported by the literature as the signature of infection from community-acquired viral pneumonia and bacterial pneumonia has some overlapping features, which is confusing the networks. It is clear from Tables 3 and 4 that the performance of CheXNet is higher than the other four top-performing models. This explains that CheXNet can extract more useful CXR features from the CXR images compared to other top-performing models.

Table 4. Class-wise performance comparison for the Top-5 networks

Encoder	*Inference time $\delta(t)$(ms)*	*Class*	*Accuracy (%)*	*Precision (%)*	*Sensitivity (%)*	*F1-score (%)*
CheXNet (DenseNet 121)	17.6	Bacterial Pneumonia	97.87	93	95	94
		COVID-19	98.89	97	99	98
		Normal	99.09	99	99	99
		Viral Pneumonia	97.55	87	86	86
		Overall	96.7	96.51	97.18	96.81
EfficientNet B0	10.7	Bacterial Pneumonia	96.74	89	92	90
		COVID-19	97.19	94	93	93
		Normal	97.75	98	98	98
		Viral Pneumonia	96.81	83	81	82
		Overall	94.24	94.32	94.42	94.18
ResNet18	5.5	Bacterial Pneumonia	96.51	88	91	90
		COVID-19	96.91	93	93	93
		Normal	97.5	98	97	98
		Viral Pneumonia	96.52	81	79	80
		Overall	93.72	93.76	93.54	93.99
MobileNetV2	2.5	Bacterial Pneumonia	96.28	87	90	89
		COVID-19	96.61	92	92	92
		Normal	97.24	98	97	97
		Viral Pneumonia	96.25	80	78	79
		Overall	93.19	93.29	93.07	92.99
DenseNet201	25.8	Bacterial Pneumonia	96.17	87	90	89
		COVID-19	96.53	92	92	92
		Normal	97.09	97	97	97
		Viral Pneumonia	96.25	79	78	79
		Overall	93.02	92.67	93.07	92.91

The confusion matrix depicted in Figure 4 can be used to further confirm CheXNet's effectiveness. It is clear that the network does a good job of differentiating between normal and pathological behaviour (i.e. Viral Pneumonia, Bacterial Pneumonia, and COVID -19). It is doing an excellent job of determining the COVID-19 patients and healthy control. Although understandably, bacterial and viral pneumonia are both atypical forms of pneumonia, it can be difficult to distinguish them in the early stage as mentioned earlier. There is a good number miss-classification between bacterial and viral pneumonia. The framework is missing some of the control patients to the unhealthy group, which could be due to the early stage CXR images, where the signature of infection is not evident.

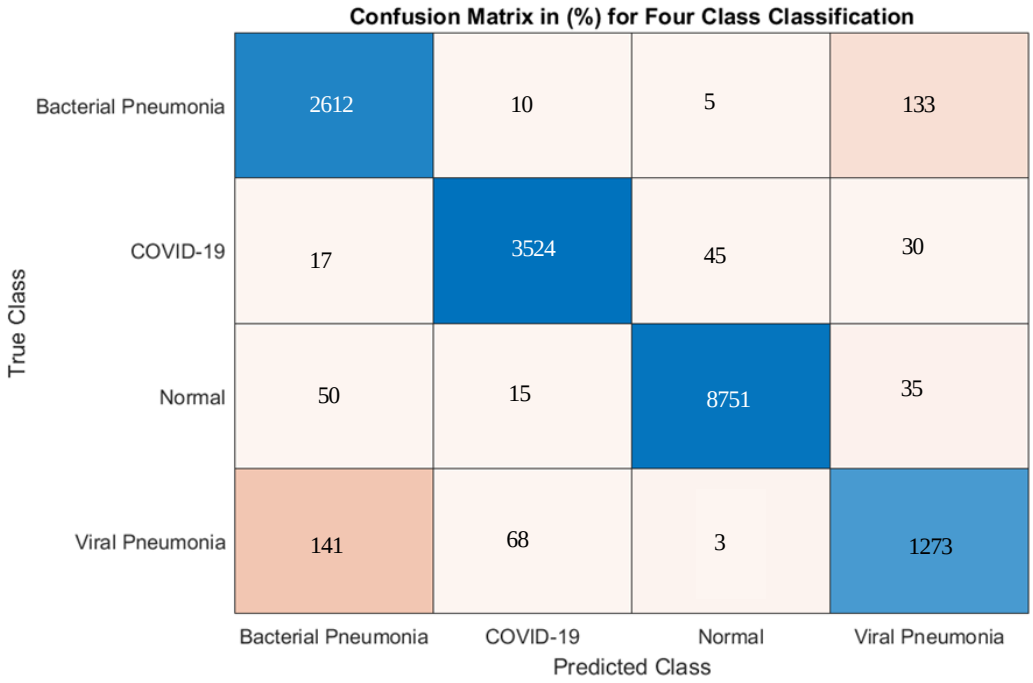


Figure 4. Confusion Matrix for the top-performing model, CheXNet.

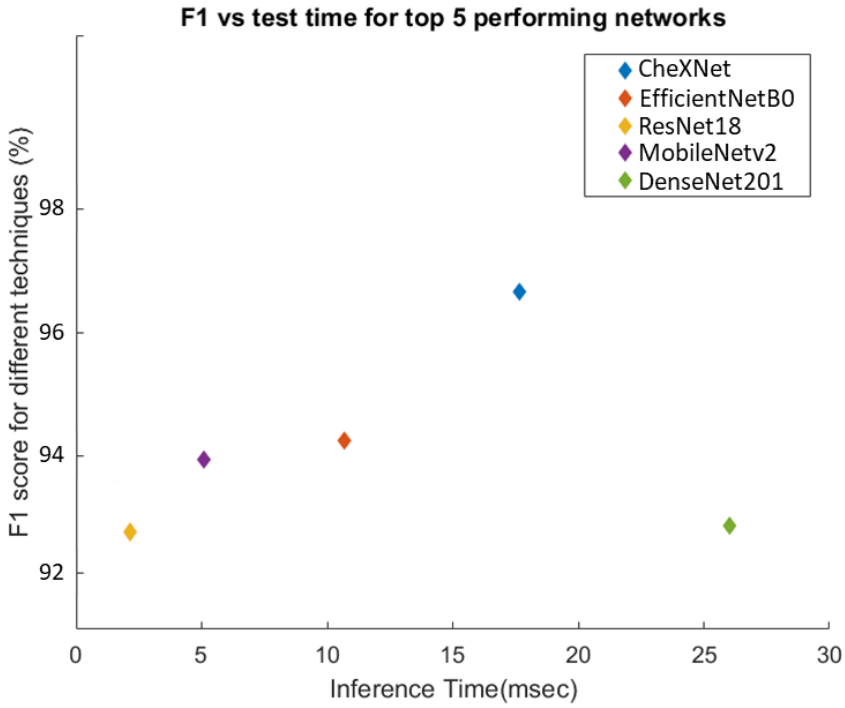


Figure 5. F1-Score versus Inference time for the top-performing 5 models.

Figure 5 shows a comparison of the Top-5 networks in terms of F1-Score and inference time. It is clear that while the networks' overall performances are comparable, the inference times vary significantly. The authors have further examined whether the networks are genuinely picking up

knowledge from the problem's "lungs," or region of interest. The Score-CAM visualization results displayed in Figure 6 can be used to confirm this. The network choosing the lung area for all classes can be verified by the Score-CAM visualization for the top-performing CheXNet network.

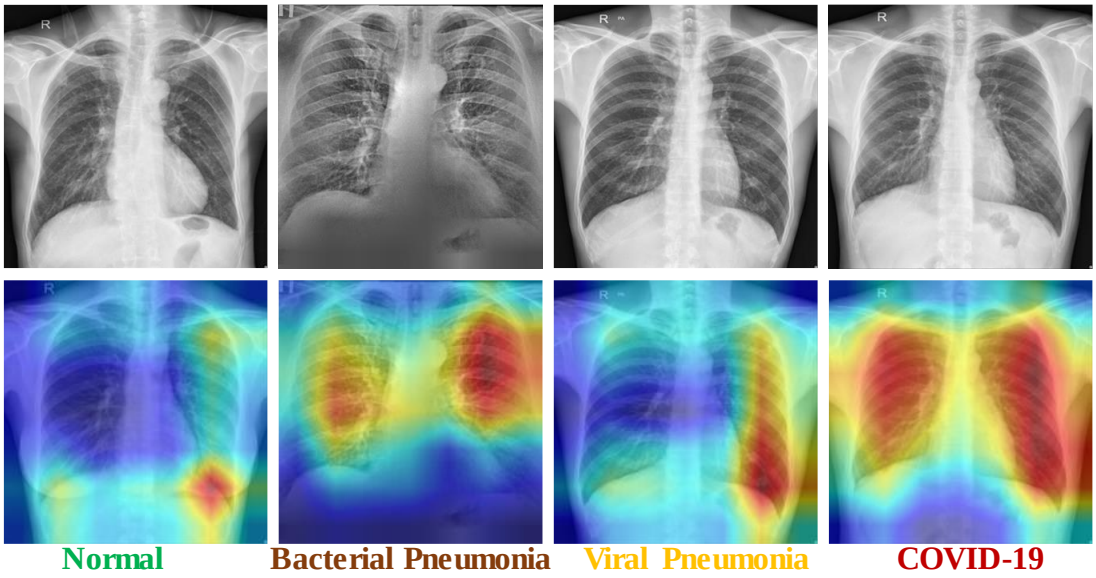


Figure 6. Score-CAM-based heat-map for the best performing CheXNet model, showing where the network is learning more to take the decision.

Table 5 summarizes the recent works on multi-class classification using deep learning algorithms. In all the cases, apart from the studies from our group, the size of the COVID dataset is small. However, the performance of this work is superior compared to the relevant literature.

Table 5. Comparison with the current state-of-art/relevant studies.

Articles	Techniques	Dataset	Performance
Tsung et al. [75]	CNN (ResNet50)	15478 chest X-ray images (473 COVID)	accuracy, sensitivity, and specificity obtained are 93%, 90.1%, and 89.6%
Abbas et al. [39]	CNN (DeTraC)	1768 chest X-ray images (949 COVID)	Accuracy-93.1%
Jain et al. [76]	CNN (Inception V3, Xception, and ResNet)	6432 chest X-ray images (490 COVID)	Accuracy-96% and Recall-92%
Ohata et al. [77]	Transfer learning + machine learning method (DenseNet201 + MLP)	388 chest X-ray images (194 COVID)	Acc: 95.641%, F1-score: 95.633%, FPR: 4.103%
Ioannis et al. [78]	CNN	1427 chest X-ray images (224 COVID)	accuracy, sensitivity, and specificity obtained are 96%, 96.66%, and 96.46%
Chowdhury et al. [28]	Seven different deep CNN networks for classification	423 COVID-19, 1485 viral pneumonia, and 1579 normal chest X-ray	The classification accuracy, precision, sensitivity, and specificity

Articles	Techniques	Dataset	Performance
		images	were 99.55% and 97.9%, 97.95%, 97.9%, and 98.8%, respectively
Rahman et al. [29]	Seven different deep CNN networks for classification and a modified Unet network for segmentation	18479 chest x-ray images (3616 COVID)	accuracy of 96.29%, sensitivity of 97.28%, and the F1-score of 96.28%. In segmentation, Accuracy of 98.63%, and Dice score of 96.94%
Proposed study	Many different deep CNN networks and PCA-based framework for classification	16,712 CXR images with 8851 normal, 3616 COVID-19, 1485 viral, and 2740 bacterial pneumonia	Overall accuracy of 96.7% for 4-class problem

4. CONCLUSION

To the best of the authors' knowledge, an in-depth analysis of the performance of the popular twenty-one state-of-the-art transfer learning model has not yet been demonstrated, even though a lot of research has been done on the use of artificial intelligence for COVID-19 detection from CXR images. The research community will benefit from this paper's addition to the body of knowledge about transfer learning's role in COVID-19 detection because it will help them decide whether to deploy a particular network given that networks vary not only in terms of performance but also in terms of size, parameters, and inference time. Overall performance measurements reveal that all the networks perform well, however, CheXNet is found to perform better than all the other networks. It is found that the CheXNet is picking up information from the lung areas in the reliability test conducted utilizing Score-CAM visualization. The results show that CheXNet outperforms with an overall accuracy of 96.7%, with an inference time of 17.6 ms, which is inline with the findings reported in the previous work of the authors [74]. In future, the authors will deploy such framework in cloud platform to carryout a multi-centre study and to improve the model generalizability.

ACKNOWLEDGEMENTS

This work was supported by the Qatar National Research Grant NPRP11S-0102-180178 and UREP28-144-3-046. The statements made herein are solely the responsibility of the authors.

REFERENCES

[1] S. A. Harmon, T. H. Sanford, S. Xu, E. B. Turkbey, H. Roth, Z. Xu, et al., "Artificial intelligence for the detection of COVID-19 pneumonia on chest CT using multinational datasets," Nature communications, vol. 11 (1), pp. 1-7(2020)

[2] N. Abughanam, S. S. M. Gaben, M. E. Chowdhury, and A. Khandakar, "Investigating the effect of materials and structures for negative pressure ventilators suitable for pandemic situation," Emergent Materials, vol. 4 (1), pp. 313-327(2021)

[3] M. Black, A. Lee, and J. Ford, "Vaccination against COVID-19 and inequalities–avoiding making a bad situation worse," Public Health in Practice (Oxford, England), 2021)

[4] I. Arevalo-Rodriguez, D. Buitrago-Garcia, D. Simancas-Racines, P. Zambrano-Achig, R. Del Campo, A. Ciapponi, et al., "False-negative results of initial RT-PCR assays for COVID-19: a systematic review," PloS one, vol. 15 (12), p. e0242958(2020)

[5] A. Tahamtan and A. Ardebili, "Real-time RT-PCR in COVID-19 detection: issues affecting the results," Expert review of molecular diagnostics, vol. 20 (5), pp. 453-454(2020)
[6] A. Afzal, "Molecular diagnostic technologies for COVID-19: Limitations and challenges," Journal of advanced research, 2020)
[7] W. H. Organization, "Use of chest imaging in COVID-19: a rapid advice guide, 11 June 2020," World Health Organization2020.
[8] W. Wang, Y. Xu, R. Gao, R. Lu, K. Han, G. Wu, et al., "Detection of SARS-CoV-2 in different types of clinical specimens," Jama, vol. 323 (18), pp. 1843-1844(2020)
[9] M. E. Chowdhury, A. Khandakar, S. Ahmed, F. Al-Khuzaei, J. Hamdalla, F. Haque, et al., "Design, construction and testing of iot based automated indoor vertical hydroponics farming test-bed in qatar," Sensors, vol. 20 (19), p. 5637(2020)
[10] M. E. Chowdhury, T. Rahman, A. Khandakar, M. A. Ayari, A. U. Khan, M. S. Khan, et al., "Automatic and Reliable Leaf Disease Detection Using Deep Learning Techniques," AgriEngineering, vol. 3 (2), pp. 294-312(2021)
[11] M. E. Chowdhury, T. Rahman, A. Khandakar, N. Ibtehaz, A. U. Khan, M. S. Khan, et al., "Tomato Leaf Diseases Detection Using Deep Learning Technique," 2021)
[12] A. Khandakar, M. EH Chowdhury, M. Khoda Kazi, K. Benhmed, F. Touati, M. Al-Hitmi, et al., "Machine learning based photovoltaics (PV) power prediction using different environmental parameters of Qatar," energies, vol. 12 (14), p. 2782(2019)
[13] A. Khandakar, A. Rizqullah, A. Ashraf Abdou Berbar, M. Rafi Ahmed, A. Iqbal, M. E. Chowdhury, et al., "A Case Study to Identify the Hindrances to Widespread Adoption of Electric Vehicles in Qatar," Energies, vol. 13 (15), p. 3994(2020)
[14] A. Khandakar, M. E. H. Chowdhury, A. Gonzales Jr, S. Pedro, F. Touati, N. A. Emadi, et al., "Case study to analyze the impact of multi-course project-based learning approach on education for sustainable development," Sustainability, vol. 12 (2), p. 480(2020)
[15] A. Khandakar and A. Mahmoud Salem Mohamed, "Understanding probabilistic cognitive relaying communication with experimental implementation and performance analysis," Sensors, vol. 19 (1), p. 179(2019)
[16] A. Khandakar, A. Touati, F. Touati, A. Abdaoui, and A. Bouallegue, "Experimental setup to validate the effects of major environmental parameters on the performance of FSO communication link in Qatar," Applied Sciences, vol. 8 (12), p. 2599(2018)
[17] M. E. Chowdhury, A. Khandakar, K. Alzoubi, S. Mansoor, A. M Tahir, M. B. I. Reaz, et al., "Real-time smart-digital stethoscope system for heart diseases monitoring," Sensors, vol. 19 (12), p. 2781(2019)
[18] M. E. Chowdhury, K. Alzoubi, A. Khandakar, R. Khallifa, R. Abouhasera, S. Koubaa, et al., "Wearable real-time heart attack detection and warning system to reduce road accidents," Sensors, vol. 19 (12), p. 2780(2019)
[19] M. H. Chowdhury, M. N. I. Shuzan, M. E. Chowdhury, Z. B. Mahbub, M. M. Uddin, A. Khandakar, et al., "Estimating blood pressure from the photoplethysmogram signal and demographic features using machine learning techniques," Sensors, vol. 20 (11), p. 3127(2020)
[20] M. E. Chowdhury, A. Khandakar, Y. Qiblawey, M. B. I. Reaz, M. T. Islam, and F. Touati, "Machine Learning in Wearable Biomedical Systems," in Sports Science and Human Health-Different Approaches, ed: IntechOpen, 2020.
[21] Y. Qiblawey, A. Tahir, M. E. Chowdhury, A. Khandakar, S. Kiranyaz, T. Rahman, et al., "Detection and severity classification of COVID-19 in CT images using deep learning," Diagnostics, vol. 11 (5), p. 893(2021)
[22] A. M. Tahir, M. E. Chowdhury, A. Khandakar, T. Rahman, Y. Qiblawey, U. Khurshid, et al., "COVID-19 Infection Localization and Severity Grading from Chest X-ray Images," arXiv preprint arXiv:2103.07985, 2021)
[23] A. Tahir, Y. Qiblawey, A. Khandakar, T. Rahman, U. Khurshid, F. Musharavati, et al., "Coronavirus: Comparing COVID-19, SARS and MERS in the eyes of AI," arXiv preprint arXiv:2005.11524, 2020)
[24] A. Jacobi, M. Chung, A. Bernheim, and C. Eber, "Portable chest X-ray in coronavirus disease-19 (COVID-19): A pictorial review," Clinical imaging, 2020)
[25] S. Kooraki, M. Hosseiny, L. Myers, and A. Gholamrezanezhad, "Coronavirus (COVID-19) outbreak: what the department of radiology should know," Journal of the American college of radiology, vol. 17 (4), pp. 447-451(2020)

[26] T. Rahman, A. Khandakar, M. A. Kadir, K. R. Islam, K. F. Islam, R. Mazhar, et al., "Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization," IEEE Access, vol. 8 pp. 191586-191601(2020)
[27] T. Rahman, M. E. Chowdhury, A. Khandakar, K. R. Islam, K. F. Islam, Z. B. Mahbub, et al., "Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection using Chest X-ray," Applied Sciences, vol. 10 (9), p. 3233(2020)
[28] M. E. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub, et al., "Can AI help in screening viral and COVID-19 pneumonia?," IEEE Access, vol. 8 pp. 132665-132676(2020)
[29] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. A. Kashem, et al., "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images," Computers in biology and medicine, vol. 132 p. 104319(2021)
[30] E. S. Amis Jr, P. F. Butler, K. E. Applegate, S. B. Birnbaum, L. F. Brateman, J. M. Hevezi, et al., "American College of Radiology white paper on radiation dose in medicine," Journal of the american college of radiology, vol. 4 (5), pp. 272-284(2007)
[31] E. Baratella, P. Crivelli, C. Marrocchio, A. M. Bozzato, A. D. Vito, G. Madeddu, et al., "Severity of lung involvement on chest X-rays in SARS-coronavirus-2 infected patients as a possible tool to predict clinical progression: an observational retrospective analysis of the relationship between radiological, clinical, and laboratory data," Jornal Brasileiro de Pneumologia, vol. 46 (5), 2020)
[32] G. D. Rubin, C. J. Ryerson, L. B. Haramati, N. Sverzellati, J. P. Kanne, S. Raoof, et al., "The role of chest imaging in patient management during the COVID-19 pandemic: a multinational consensus statement from the Fleischner Society," Chest, vol. 158 (1), pp. 106-116(2020)
[33] T. Ai, Z. Yang, H. Hou, C. Zhan, C. Chen, W. Lv, et al., "Correlation of chest CT and RT-PCR testing for coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases," Radiology, vol. 296 (2), pp. E32-E40(2020)
[34] L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang, B. Kong, et al., "Artificial intelligence distinguishes COVID-19 from community acquired pneumonia on chest CT," Radiology, 2020)
[35] L. Wang, Z. Q. Lin, and A. Wong, "Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images," Scientific Reports, vol. 10 (1), pp. 1-12(2020)
[36] P. R. Bassi and R. Attux, "A deep convolutional neural network for covid-19 detection using chest x-rays," Research on Biomedical Engineering, pp. 1-10(2021)
[37] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, and P. R. Pinheiro, "Covidgan: data augmentation using auxiliary classifier gan for improved covid-19 detection," Ieee Access, vol. 8 pp. 91916-91923(2020)
[38] N. K. Chowdhury, M. M. Rahman, and M. A. Kabir, "PDCOVIDNet: a parallel-dilated convolutional neural network architecture for detecting COVID-19 from chest X-ray images," Health information science and systems, vol. 8 (1), pp. 1-14(2020)
[39] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network," Applied Intelligence, vol. 51 (2), pp. 854-864(2021)
[40] M. Z. Che Azemin, R. Hassan, M. I. Mohd Tamrin, and M. A. Md Ali, "COVID-19 deep learning prediction model using publicly available radiologist-adjudicated chest X-ray images as training data: preliminary findings," International Journal of Biomedical Imaging, vol. 2020 2020)
[41] I. U. Khan and N. Aslam, "A deep-learning-based framework for automated diagnosis of COVID-19 using X-ray images," Information, vol. 11 (9), p. 419(2020)
[42] M. Loey, F. Smarandache, and N. E. M Khalifa, "Within the lack of chest COVID-19 X-ray dataset: a novel detection model based on GAN and deep transfer learning," Symmetry, vol. 12 (4), p. 651(2020)
[43] L. Brunese, F. Mercaldo, A. Reginelli, and A. Santone, "Explainable deep learning for pulmonary disease and coronavirus COVID-19 detection from X-rays," Computer Methods and Programs in Biomedicine, vol. 196 p. 105608(2020)
[44] N. N. Das, N. Kumar, M. Kaur, V. Kumar, and D. Singh, "Automated deep transfer learning-based approach for detection of COVID-19 infection in chest X-rays," Irbm, 2020)
[45] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. R. Acharya, "Automated detection of COVID-19 cases using deep neural networks with X-ray images," Computers in biology and medicine, vol. 121 p. 103792(2020)

[46] A. Degerli, M. Ahishali, M. Yamac, S. Kiranyaz, M. E. Chowdhury, K. Hameed, et al., "COVID-19 Infection Map Generation and Detection from Chest X-Ray Images," arXiv preprint arXiv:2009.12698, 2020)
[47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE conference on computer vision and pattern recognition, 2009, pp. 248-255.
[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, vol. 25 pp. 1097-1105(2012)
[49] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014)
[50] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018)
[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.
[52] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., "Going deeper with convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1-9.
[53] B. Raj, "A simple Guide to the versions of the Inception network," Retrieved from Towards Data Science, 2018)
[54] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818-2826.
[55] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in Proceedings of the AAAI Conference on Artificial Intelligence, 2017.
[56] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251-1258.
[57] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700-4708.
[58] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2097-2106.
[59] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, et al., "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," arXiv preprint arXiv:1711.05225, 2017)
[60] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017)
[61] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510-4520.
[62] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6848-6856.
[63] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8697-8710.
[64] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in International Conference on Machine Learning, 2019, pp. 6105-6114.
[65] kaggle. RSNA Pneumonia Detection Challenge [Online]. Available: https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data. [Accessed on 09-June-2020]
[66] (2020). BIMCV-COVID19, Datasets related to COVID19's pathology course [Online]. Available: https://bimcv.cipf.es/bimcv-projects/bimcv-covid19/#1590858128006-9e640421-6711. [Accessed on: 06 August 2020]
[67] (2020). covid-19-image-repository [Online]. Available: https://github.com/ml-workgroup/covid-19-image-repository/tree/master/png. [Accessed on: 06 August 2020]

[68] R. Chen, W. Liang, M. Jiang, W. Guan, C. Zhan, T. Wang, et al., "Risk factors of fatal outcome in hospitalized subjects with coronavirus disease 2019 from a nationwide analysis in China," Chest, 2020)
[69] Z. Weng, Q. Chen, S. Li, H. Li, Q. Zhang, S. Lu, et al., "ANDC: an early warning score to predict mortality risk for patients with Coronavirus Disease 2019," 2020)
[70] J. Liu, Y. Liu, P. Xiang, L. Pu, H. Xiong, C. Li, et al., "Neutrophil-to-lymphocyte ratio predicts severe illness patients with 2019 novel coronavirus in the early stage," MedRxiv, 2020)
[71] I. Huang and R. Pranata, "Lymphopenia in severe coronavirus disease-2019 (COVID-19): systematic review and meta-analysis," Journal of Intensive Care, vol. 8 (1), pp. 1-10(2020)
[72] (2020). COVID-CXNet [Online]. Available: https://github.com/armiro/COVID-CXNet. [Accessed on: 06 August 2020]
[73] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, et al., "Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 24-25.
[74] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. A. Kashem, et al., "Exploring the Effect of Image Enhancement Techniques on COVID-19 Detection using Chest X-rays Images," Computers in Biology and Medicine, p. 104319(2021).
[75] Lin, T. C., & Lee, H. C. (2020, August). Covid-19 chest radiography images analysis based on integration of image preprocess, guided grad-CAM, machine learning and risk management. In Proceedings of the 4th International Conference on Medical and Health Informatics (pp. 281-288).
[76] Jain, R., Gupta, M., Taneja, S., & Hemanth, D. J. (2021). Deep learning-based detection and analysis of COVID-19 on chest X-ray images. Applied Intelligence, 51(3), 1690-1700.
[77] Ohata, Elene Firmeza, et al. "Automatic detection of COVID-19 infection using chest X-ray images through transfer learning." IEEE/CAA Journal of Automatica Sinica 8.1 (2020): 239-248.
[78] Apostolopoulos, Ioannis D., and Tzani A. Mpesiana. "Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks." Physical and engineering sciences in medicine 43.2 (2020): 635-640.

AUTHORS

Muhammad E. H. Chowdhury received his PhD degree from the University of Nottingham, U.K., in 2014. He worked as a Postdoctoral Research Fellow at the Sir Peter Mansfield Imaging Centre, University of Nottingham. He is currently working as an Assistant Professor with the Department of Electrical Engineering, Qatar University. He has filed several patents and published more than 100 peer-reviewed journal articles, conference papers, and several book chapters. His current research interests include biomedical instrumentation, signal processing, wearable sensors, medical image analysis, machine learning and computer vision, embedded system design, and simultaneous EEG/fMRI. He is currently running several NPRP, UREP, and HSREP grants from Qatar National Research Fund (QNRF) and internal grants (IRCC and HIG) from Qatar University along with academic projects from HBKU and HMC. He has been involved in EPSRC, ISIF, and EPSRC-ACC grants along with different national and international projects during his tenure at the University of Nottingham. He is a Senior Member of IEEE, and a member of British Radiology, ISMRM, and HBM. He is serving as an Associate Editor for IEEE Access and a Topic Editor and Review Editor for Frontiers in Neuroscience. He has recently won the COVID-19 Dataset Award, AHS Award from HMC and National AI Competition awards for his contribution to the fight against COVID-19.

Amith Khandakar received a B.Sc. degree in electronics and telecommunication engineering from North South University, Bangladesh, and a master's degree in computing (networking concentration) from Qatar University, in 2014. He graduated as the Valedictorian (President Gold Medal Recipient) of North South University. He is an IEEE Senior Member. He is also a certified Project Management Professional and the Cisco Certified Network Administrator. He has 2 patents and published around 60 peer-reviewed journal articles, conference papers, and four book chapters. His current research interests include biomedical instrumentation, wearable sensors, medical image analysis, machine learning and Engineering Education. He is also running UREP grants from QNRF and internal grants from Qatar University.

Tawsifur Rahman is currently working as a research assistant in the machine learning group, at Qatar University. He received his B.Sc. degree in Electrical and Electronic Engineering from the University of Chittagong (Bangladesh), and his M.Sc. degree in Biomedical Engineering from the University of Dhaka (Bangladesh). His research interests include medical image analysis, big clinical data processing, computer vision, and deep learning. He is the author and co-author of about 30+ research journal articles, and a few book chapters on medical imaging, and clinical bio-markers data using artificial intelligence. Thus far, his publications have been cited 1700+ times (Source: Google Scholar). He has received 'An ICT fellowship 2019-2020' from the Information Communication & Technology (ICT) ministry of Bangladesh for his research. Moreover, he and his research team have received the 'Kaggle COVID-19 dataset award' from the Kaggle community.

Sakib Mahmud received his Bachelor of Science (BSc.) degree in Electrical Engineering with Honors from Qatar University (QU) in June 2020 and currently pursuing his Master of Science (MSc.) in Electrical Engineering from the same department. As an undergraduate student, he received the Dean's Award for six semesters during 2016-2019. As an undergraduate student at QU, He has been a part of two UREP projects funded by the QNRF. Upon graduation, he joined QU's master's program to pursue a postgraduate degree in electrical engineering and work as a Graduate Research Assistant (GRA) at the same time, under the same department. As a GRA, and a member of QU Machine Learning Group, he was hired in two QNRF funder NPRP grants and three High Impact grants from QU. Currently, he is hired in a High Impact grant funded by QU. Apart from his research work, he has been an Electrical Engineering Intern in the Electrical Consultant Group (E.C.G.) Qatar Branch, Data Science Intern in Data Glacier and has been selected as an Artificial Intelligence (AI) Intern in AI Ready Academy, a joint training program for Machine Learning and Data Science by Microsoft and ZAKA, among 2000 competitive applicants around the Middle East and North Africa (MENA) region, He has expertise in Electronics Circuitry design, 3D Modeling, Biomedical Signal and Image Processing, Computer Vision, Machine and Deep Learning, and Data Analytics and Visualization in various platforms. Currently, so far, he has published 10 articles in peer reviewed Journals, one conference paper and one patent, many under review, thus accumulating around 60 citations in Google Scholar.

RESEARCH ON WIRELESS POWERED COMMUNICATION NETWORKS SUM RATE MAXIMIZATION BASED ON TIME REVERSAL OFDM

Wei Liu[1]、Fang Wei Li[2]、Hai Bo Zhang[3]、Bo Li[4]

[1] School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China

[2] Chongqing Key Laboratory of Public Big Data Security Technology, Chongqing, China

[3] School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China

[4]Chongqing Key Lab of Mobile Communications Technology，Chongqing University of Post and Communications

ABSTRACT

This paper studies a wireless power communication network(WPCN) based on orthogonal frequency division multiplexing (OFDM) with time reversal(TR). In this paper, the " Harvest Then Transmit " protocol is adopted, and the transmission time block is divided into three stages, the first stage is for power transmission, the second stage is for TR detection, and the third stage is for information transmission. The energy limited access point (AP) and the terminal node obtain energy from the radio frequency signal sent by the power beacon (PB) to assist the terminal data transmission. The energy limited AP and the terminal node obtain energy from the radio frequency signal sent by the PB to assist the terminal data transmission. In the TR phase and the wireless information transmission (WIT) phase, the terminal transmits the TR detection signal to the AP using the collected energy, and the AP uses the collected energy to transmit independent signals to a plurality of terminals through OFDM. In order to maximize the sum rate of WPCN, the energy collection time and AP power allocation are jointly optimized. Under the energy causal constraint, the subcarrier allocation, power allocation and time allocation of the whole process are studied, and because of the binary variables involved in the subcarrier allocation, the problem belongs to the mixed integer non-convex programming problem. the problem is transformed into a quasiconvex problem, and then binary search is used to obtain the optimal solution. The simulation results verify the effectiveness of this scheme. The results show that the proposed scheme significantly improves the sum rate of the terminal compared to the reference scheme.

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 23-40, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122002

KEYWORDS

Wireless Powered Communication Network, Ttime Reversal, OFDM

1. INTRODUCTION

In recent years, the rapid development of connected devices has led to the rise of Internet of Things (IoT) applications, which are used in smart homes, smart transportation, medical monitoring, and disaster warning information [1][2]. In a traditional network, node devices are powered by a fixed energy source such as a battery, and the network life is limited due to power exhaustion, thereby affecting the performance of the network. In order to extend the runtime of the network, the battery needs to be replaced or replenished after the battery power is exhausted.However, in some applications, it is not technically and economically feasible for wireless nodes to be deployed in large-scale sensor networks or implanted into the human body to cause regular replacement or charging[3][4]. Energy harvesting (EH) is considered to be a promising technology to replace traditional energy sources such as batteries because it provides more cost-effective energy for wireless networks [5]. Especially, because radio frequency (RF) signals have dual purposes of wireless information transmission (WIT) and wireless energy transmission (WET), the EH of RF signals has attracted great attention [6]. RF power transmission is an EH technology. Wireless network nodes collect energy from RF signals and convert them into electrical energy [7]. Among various energy transmission systems, wireless power supply communication network (WPCN) has been widely studied [8]. For WPCN, WET and WIT are completely separate. Compared with the traditional wireless network whose terminal devices are powered by batteries, WPCN is more suitable for small wireless networks and can provide permanent energy for small wireless sensors and other terminal devices. In [9],a “harvest-then-transmit” protocol was proposed for a multi-user WPCN, where users first harvest energy from RF signals broadcast by a single antenna hybrid access point (AP) in the

DL and then transmit information to the AP in the UL.

In order to meet the rapidly increasing demand for wireless data traffic, and rate maximization is considered as an important index, and resource allocation has been widely studied due to the trade-off between WET and WIT in wireless packet networks. In [10], a WPCN protocol is proposed, in which users first obtain energy through a hybrid access point (HAP), and then transmit data to the HAP through time division multiple access (TDMA). And then through the analysis for the wireless power transmission and the optimum time distribution of information transmission, to maximize the total throughput of all users. In [11], the optimal power allocation for downlink power transmission and uplink information transmission is derived for multiple access fading networks using TDMA or Frequency Division Multiple Access (FDMA). In [12], the joint power allocation of downlink power transmission and the time allocation of uplink information transmission are optimized for WPCN based on TDMA. In [13], considering the limited energy storage capacity of users, an optimal allocation algorithm of energy and time is proposed. In [14], a new model for simultaneous transmission of wireless information and energy in WPCN is proposed. In [15], various resource allocation problems of cognitive WPCN are studied, aiming at maximizing the performance of

secondary users while satisfying the service quality of primary users. In [16], a full-duplex WPCN based on orthogonal frequency division multiplexing (OFDM) is considered, and the subcarrier allocation and power allocation under ideal and non-ideal self-interference cancellation are studied. The work in [17] all considers that the HAP is equipped with two antennas supporting full duplex and the users are half duplex. When the user supports full duplex, the performance of WPCN can be further improved by collecting more energy and selecting an appropriate antenna for uplink transmission. However, some practical scenarios located in toxic environments, underground, tunnels, remote areas, or disaster areas involve energy-constrained nodes that transmit information to IoT devices under limited power constraints, and environmental factors increase the difficulty of battery replacement. Hence, transmitters can harvest energy from dedicated RF signals to communicate with IoT devices [18]. Using power beacon (PB) as a dedicated wireless energy supply point, energy supply can be carried out for energy-constrained devices in the WET phase without the need for network communication [19]. The coverage probability of PB-based WPCN has been studied in [20], in which transmitters collect energy through PB during WET and then communicate with their corresponding receivers in WIT. In [21] proposes a new harvest-transmission scheduling method to maximize the total throughput of WPCN. In the designed method, multiple terminals can transmit information to AP nodes at the same time in one time slot. With single-user detection/decoding, only one of the composite signals can be successfully decoded at a time, and the other signals are treated as interference. Due to the existence of interference signal, the throughput performance is limited. In view of the above considerations, multiuser detection and continuous interference cancellation (SIC) are applied. Using the SIC algorithm, some interference signals can be decoded instead of being regarded as noise.

Based on the above discussion and literature review, the time reversal technique is used to suppress the interference of OFDM WPCN system in multi-terminal interference channel. By introducing PB, a WPCN model is constructed, which is composed of a single antenna PB, an energy-constrained AP and $k > 1$ single-antenna terminal devices.

AP and Internet of things terminals obtain energy from RF signals transmitted by PB to assist the downlink communication between AP and Internet of things devices. The goal is to maximize the total throughput of the system by jointly optimizing the time allocation and transmission power allocation of the system.

The main contributions of this paper are summarized as follows:

1. we propose a WPCN model, in which AP and IoT terminal nodes receive RF energy from PB, and then AP transmits information to IoT devices on the downlink. In the proposed WPCN, AP adopts the HTT protocol supported by PB in the WET phase, and uses OFDM to allocate transmission power in the WIT phase to transmit information to IoT devices.
2. For OFDM WPCN system, the time reversal technique is used to suppress the interference between transmission antennas, and the problem of sum rate maximization is solved by jointly optimizing EH time and transmission power. The resulting problems are all non-convex, and for WPCN of OFDM, because the subcarrier allocation involves binary variables, the mixed integer programming problem is transformed into a

quasi-convex problem, and then the optimal solution is obtained by binary search.

3. Finally, we give a large number of numerical results to prove that our proposed algorithm is effective in improving system sum rate.

The rest of this paper is organized as follows. A WPCN model is introduced in section 2. Section 3 investigate the system sum rate problem, respectively. Section 4 evaluates the performance of the presented algorithms by conducting numerical simulations and section 5 concludes the paper.

2. SYSTEM MODEL

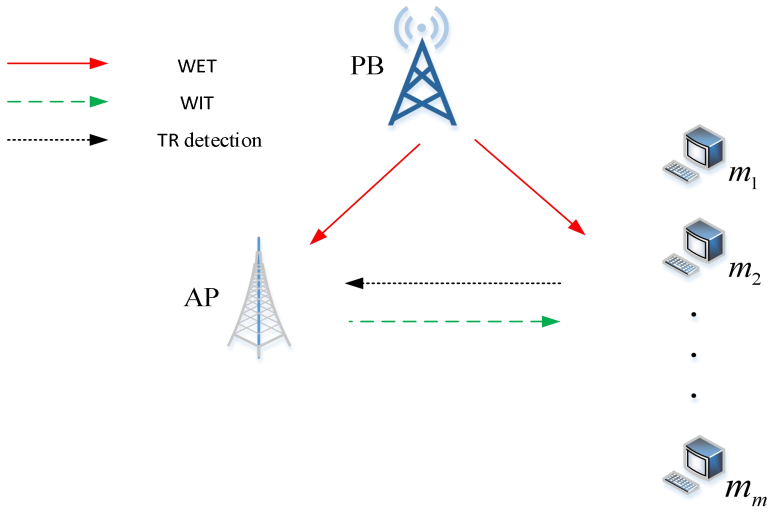


Figure 1. WPCN system model

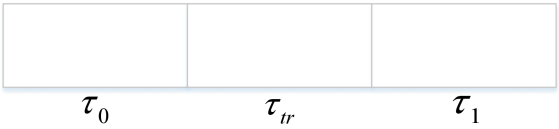


Figure 2. Frame structure of WPCN

In this paper, we consider a WPCN of OFDM based on time reversal, as shown in Figure 1. A single-antenna PB and an energy-constrained AP with N_T antennas and $k>1$ single-antenna Internet of things terminals are composed of WPCN. In the WPCN system, PB first carries out WET to the AP and the terminal node, and the AP and the terminal node receive the energy and store it in the rechargeable battery. Different from the traditional WPCN system, after the WET stage, the terminal node sends a time reversal probe signal to the AP. After the AP receives the probe signal, it performs WIT, Specifically, the data signal to be sent is sent to the terminal node after time reversal processing. All channels are assumed to have quasi-static flat fading and that the channel power gain remains constant during a transport block, but may vary between transport blocks. Without loss of generality, the time of a transport block is normalized to 1, which is represented by $T_{\max}$, that is, $T_{\max}=1$. Suppose in AP and all terminal channel state information (CSI) is perfectly known. In the IoT network, all transceivers, including AP, should be low-cost and low-power devices. Therefore, a

transmission block is divided into three stages by adopting the HTT protocol [16]. The first stage is that PB transmits RF energy, and AP and terminal EH, and the duration is τ_0. In the second stage, the terminal sends the time reversal detection signal with a duration of τ_{tr}, and in the third stage, the AP node transmits the information to the terminal node with a duration of τ_1.

In the WET phase, the AP and the terminal EH from the PB, and the total power received by the AP is $P_T = P_B \left\| h_p \right\|^2$, where P_B is the transmission power at the PB and $h_p \in \mathbb{C}^{N_T \times 1}$ is the channel vector from PB to AP.

The energy collected at AP can be expressed as follows:

$$E = \eta P_T \tau_0 \tag{1}$$

Where $\eta \in (0,1]$ is the energy conversion efficiency.

In WPCN network, the system performance is affected by the interference caused by multipath environment. Time reversal technology is considered to be an effective technology to combat multipath and time-selective fading in the field of wireless communication. The time-space focus of time reversal technique in multipath channel is used to suppress the interference caused by multipath effect in OFDM transmission. It is assumed that the terminal channel gains are sorted from low to high. As shown in Figure 2, AP transmits data to the terminal through N orthogonal subcarriers using OFDM during the WIT phase with duration τ_1. A binary SC allocation variable $x_{k,n}$ is introduced in the WIT phase, and SC n is allocated to the mth terminal, $x_{k,n} = 1$; otherwise, $x_{k,n} = 0$. The channel impulse response of the jth antenna of the AP and the mth terminal on SC n is $h_{jm}^{(n)}$. Considering the influence of multipath effect on the system in practice, it is assumed that all terminals send signals to AP at the same time, and the channel impulse response $h_{jm}^{(n)}$ is:

$$h_{jm}^{(n)}[t] = \sum_{l=0}^{L-1} \delta_{jm,l}^{(n)} \delta\left[t - \tau_{jm,l}^{(n)}\right] \tag{2}$$

Where L represents the number of multipaths, $l \in \{1,2,...,L-1\}$; $\delta_{jm,l}^{(n)}$ represents the amplitude of the l th multipath; $\tau_{jm,l}^{(n)}$ represents the time delay of the l th multipath. Discretize $h_{jm}^{(n)}[t]$ As：

$$h_{jm}^{(n)} = \left\{ h_{jm}^{(n)}[0], h_{jm}^{(n)}[1], ..., h_{jm}^{(n)}[L-1] \right\}^T \tag{3}$$

When the channel impulse response is discretized in time domain, the mean value a is satisfied $E\left[h_{jm}^{(n)}[l] \right] = 0$.

In the time reversal stage, the terminal node sends a time reversal probe signal to the AP node by using the EH to obtain the channel state information. At this time, the AP records the probe signal and performs time reversal preprocessing on the obtained channel state, that is, reverses it in the time domain. After time reversal processing, the equivalent channel between the AP and the terminal node can be obtained, and the equivalent channel is:

$$g_{jm}^{(n)} = \left\{ g_{jm}^{(n)}[0], g_{jm}^{(n)}[1], ..., g_{jm}^{(n)}[L-1] \right\}^T \tag{4}$$

Where each element in the above matrix is the tap value after time reversal processing and normalization can be written as:

$$g_{jm}^{(n)}[p] = \frac{\overline{h}_{jm}^{(n)}[L-1-P]}{\sqrt{E\left[\left\| h_{jm}^{(n)} \right\|^2 \right]}} = \frac{\overline{h}_{jm}^{(n)}[L-1-P]}{\sqrt{E\left[\left| \sum_{l=0}^{L-1} h_{jm}^{(n)}[l]^2 \right| \right]}} \tag{5}$$

where $\overline{\bullet}$ represents conjugation.

After time reversal processing, the channel gain $h_{jm}^{(n)}$ becomes $h_{jm}^{(n)} * g_{jm}^{(n)}$, which is specifically expressed as:

$$\left(h_{jm}^{(n)} * g_{jm}^{(n)} \right)[p] = \sum_{l=0}^{L-1} h_{jm}^{(n)}[l] g_{jm}^{(n)}[p-l] = \frac{\sum_{l=0}^{L-1} h_{jm}^{(n)}[l] \overline{h}_{jm}^{(n)}[L-1+l-p]}{\sqrt{E\left[\left| \sum_{l=0}^{L-1} h_{jm}^{(n)}[l]^2 \right| \right]}} \tag{6}$$

Where $h_{jm}^{(n)} * g_{jm}^{(n)}$ represents the convolution operation, and $p \in \{0,1,...,2L-2\}$. When $p = L-1$, the autocorrelation function is generated in the corresponding formula, and the above formula takes the maximum power center peak value at this time. According to the reference [21], most of the power of the signal will focus on the central tap, that is, the $L-1$ tap, so it is considered that the power on the $L-1$ tap is the transmission power of the ideal signal. The transmission power of the AP allocated to the SCn for transmitting information to the mth terminal is represented as $p_{m,n}$.

Therefore, the achievable rate of the mth terminal on the nth SC is:

$$r_{k,n} = \frac{B\tau_1}{N} x_{m,n} \log_2 \left(1 + SINR_{m,n} \right) \tag{7}$$

The $SINR$ at the mth terminal is expressed as:

$$SINR_{m,n}=\frac{p_{m,n}\left|h_{jm}^{(n)}*g_{jm}^{(n)}\right|^2}{(B/N)\left[\sum\limits_{\substack{i=1\\ i\neq j}}^{N_T}\sum\limits_{p=0}^{2L-1}p_{m,n,i}\left(h_{jm}^{(n)}*g_{jm}^{(n)}\right)+\sigma^2\right]} \tag{8}$$

Where $\sum\limits_{\substack{i=1\\ i\neq j}}^{N_T}\sum\limits_{p=0}^{2L-1}p_{k,n,i}\left(h_{jm}^{(n)}*g_{jm}^{(n)}\right)$ represents the interference of the $i\left(i\neq j\right)$ th antenna of the AP to the jth terminal on the subcarrier n; σ^2 represents noise power; For the convenience of the following calculation commands:

$$A_{m,n}=\frac{\left|h_{jm}^{(n)}*g_{jm}^{(n)}\right|^2}{(B/N)\left[\sum\limits_{\substack{i=1\\ i\neq j}}^{N_T}\sum\limits_{p=0}^{2L-1}p_{m,n,i}\left(h_{jm}^{(n)}*g_{jm}^{(n)}\right)+\sigma^2\right]}$$

Then, the sum rate of the WPCN scheme based on time reversal OFDM is calculated as:

$$R_{sum}=\sum_{m=1}^{M}\sum_{n=1}^{N}r_{mn}=\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{B\tau_1}{N}x_{m,n}\log_2\left(1+SINR_{m,n}\right) \tag{9}$$

3. PROBLEM PLANNING AND ALGORITHM DESIGN

3.1. Problem Planning

The optimization goal of this paper is to maximize the system sum rate of TR-OFDM-WPCN. By jointly optimizing τ_0, $x_{m,n}$, and $p_{m,n}$, the optimization problem in this paper is expressed as:

$$\begin{aligned}
&OP_1:\ \max_{\tau_0,x_{m,n},p_{m,n}}\ R_{sum}\\
&s.t.\ \ C_1:p_{m,n}\geq 0\quad \forall m\in M\ \ \forall n\in N\\
&\qquad C_2:\sum_{m=1}^{M}x_{m,n}=1\quad \forall m\in M\quad \forall n\in N\\
&\qquad C_3:x_{m,n}\in\{0,1\}\ \ \forall m\in M\quad \forall n\in N\\
&\qquad C_4:\left(1-\tau_0-\tau_{tr}\right)\sum_{m=1}^{M}\sum_{n=1}^{N}x_{m,n}p_{m,n}\leq E\quad \forall m\in M\quad \forall n\in N\\
&\qquad C_5:0\leq\tau_0\leq 1
\end{aligned} \tag{10}$$

Constraints C_2 and C_3 indicate that an SC is accurately assigned to each link. Constraint

C_4 means that the total energy consumed by AP in the WIT phase is less than the EH in the WET phase. In problem OP_1, the strong coupling between τ_0, $p_{m,n}$ and $x_{m,n}$ leads to the non-concavity of the objective function and the non-convexity of C_4. In addition, because there are binary assignment variables in this problem, OP_1 is a mixed integer nonconvex programming problem.

3.2. Algorithm Design

In this section, the joint energy acquisition time, SC and power allocation solution will be introduced to maximize the sum rate of WPCN of OFDM of TR. rewrite C_4 in OP_1 as:

$$\tau_0 \geq \frac{\left(1-\tau_{tr}\right)\sum_{m=1}^{M}\sum_{n=1}^{N} x_{m,n}p_{m,n}}{\sum_{m=1}^{M}\sum_{n=1}^{N} x_{m,n}p_{m,n} + \eta P_T} \tag{11}$$

Obviously the objective function in OP_1 is a non-increasing function of τ_0. Therefore, according to constraint C_4 in OP_1, for a given power allocation, the optimal τ_0 should hold the equation:

$$\tau_0 = \frac{\left(1-\tau_{tr}\right)\sum_{m=1}^{M}\sum_{n=1}^{N} x_{m,n}p_{m,n}}{\sum_{m=1}^{M}\sum_{n=1}^{N} x_{m,n}p_{m,n} + \eta P_T} \tag{12}$$

Substituting Equation (12) into (9), we get:

$$R_{sum} = \frac{\eta B P_T \left(1-\tau_{tr}\right)}{N\left(\sum_{m=1}^{M}\sum_{n=1}^{N} x_{m,n}p_{m,n} + \eta P_T\right)} \sum_{m=1}^{M}\sum_{n=1}^{N} x_{m,n}\log_2\left(1+SINR_{m,n}\right) \tag{13}$$

Rewrite OP_1 as:

$$\begin{aligned} OP_2: & \max_{\tau_0, x_{m,n}, p_{m,n}} \quad R_{sum} \\ s.t. \quad & C_1: p_{m,n} \geq 0 \quad \forall m \in M \quad \forall n \in N \\ & C_2: \sum_{m=1}^{M} x_{m,n} = 1 \quad \forall m \in M \quad \forall n \in N \\ & C_3: x_{m,n} \in \{0,1\} \quad \forall m \in M \quad \forall n \in N \end{aligned} \tag{14}$$

Lemma 1: Suppose g is a quadratic differentiable function. When ax+b>0, g(x)/(ax+b) is quasi-convex if y=g(x) is convex, and g(x)/(ax+b) is quasi-concave if y = g(x) is a concave function. (proof see Appendix A)

From Lemma 1, it can be observed that the objective function of OP_2 is a quasi-concave function with respect to $p_{m,n}$ given SC n .In order to solve OP_2 , the relaxation variable θ is introduced. Then OP_2 can be rewritten as:

$$\begin{aligned} OP_3: &\ \max \ R \\ s.t. &\ C_1, C_2, C_3 \\ &\ C_6: \max_{x_{m,n}, p_{m,n}} R_{sum} \le \theta \end{aligned} \tag{15}$$

OP_3 is a convex optimization problem, so its unique optimal solution can be found by the Lagrangian dual method.

The Lagrangian function of the optimization problem OP_3 can be written as:

$$\begin{aligned} &L\left(p_m, \mu\right) = R_{sum} - \mu\left(R_{sum} - \theta\right) \\ &= \frac{\left(1-\mu\right)\eta BP_T\left(1-\tau_{tr}\right)}{N} \sum_{m=1}^{M}\sum_{n=1}^{N} x_{m,n} \log_2\left(1+SINR\right) + \mu\theta\left(\sum_{m=1}^{M}\sum_{n=1}^{N} x_{m,n} p_{m,n} + \eta P_T\right) \end{aligned} \tag{16}$$

where $\mu \ge 0$ is the corresponding non-negative Lagrange multiplier. For a given SC allocation $x_{m,n} = 1$, the optimal power allocation on the SC in OP_3 can be solved by solving the above problem. According to the KKT condition, the analytical solution of optimal power allocation can be obtained(proof see Appendix B):

$$p_{m,n} = \left[\frac{\left(\mu-1\right)\eta BP_T\left(1-\tau_{tr}\right)}{\mu\theta N \ln 2} - \frac{1}{A_{m,n}}\right]^+ \tag{17}$$

Where $\left[x\right]^+ = \max(0, x)$.

Substituting formula (17) into (16), we get:

$$L\left(p_{m,n}, \mu\right) = \sum_{m=1}^{M}\sum_{n=1}^{N} x_{m,n}\left[\frac{C}{N\ln 2} - \frac{\mu\theta}{A_{m,n}}\right]^+ - \frac{C}{N}\log_2\left(1 + \left[\frac{CA_{m,n}}{\mu\theta\ln 2} - 1\right]^+\right) + \eta\mu\theta P_T \tag{18}$$

Where $C = \left(1-\mu\right)\eta BP_T\left(1-\tau_{tr}\right)$

$$\begin{aligned} OP_4: \quad & \max_{n} \; L\left(p_{m,n},\mu\right) \\ s.t. \quad & C_2: \sum_{m=1}^{M} x_{m,n}=1 \quad \forall m\in M \quad \forall n\in N \\ & C_3: x_{m,n}\in\{0,1\} \quad \forall m\in M \quad \forall n\in N \end{aligned} \tag{19}$$

Since each terminal only corresponds to one non-zero SC, the SC allocation function is defined as:

$$x_{m^{op},n}^{op}=\begin{cases}1, & \mathrm{m}=m^{op}\\ 0, & \forall m\neq m^{op}\end{cases} \tag{20}$$

Where $x_{m^{op},n}^{op}$ represents the optimal value of $x_{m,n}$ in the formula.

$$m^{op}=\arg\min_{m}\prod\left(m,n\right) \tag{21}$$

Where $\prod\left(m,n\right)=\left[\frac{C}{N\ln 2}-\frac{\mu\theta}{A_j}\right]^{+}-\frac{C}{N}\log\left(1+\left[\frac{CA_j}{\mu\theta\ln 2}-1\right]^{+}\right)$

The best θ is determined by dichotomy. Algorithm 1 is the process of solving OP_1. $\theta_l=0$ and $\theta_u=\omega$ are respectively expressed as the lower limit and upper limit of binary search, and ε is the accuracy of binary search. Define the upper bound ω of binary search as:

$$\omega=BM\log\left(1+\eta P_T\max_{m,n}\left\{A_{m,n}\right\}\right) \tag{22}$$

Table 1. Joint Time Power Allocation Scheme

Algorithm for solving OP_1
1. Calculate τ_{tr} using the given d and v_0.
2. initialization：$\theta_l=0$， $\theta_u=\omega$， $m=0$, ε is the error threshold.
3. Repeat steps (1)-(5) until step 4 is established
(1) $\theta(m)=(\theta_l+\theta_u)/2$;
(2) Obtain the optimal $p_{m,n}$ according to formula(4.16);
(3) Obtain the optimal $x_{m,n}$ according to formula(4.19);
(4) Substitute the obtained A and B into (4.11)，if $L\left(p_{m,n},\mu\right)>0$, $\theta_u=\theta$； else $\theta_l=\theta$；
(5) $m=m+1$；

4. Judge whether $\theta_u - \theta_l < \varepsilon$ is established, if established Output $p_{m,n}$ and $x_{m,n}$;

4. SIMULATION RESULTS

In this section, the sum rate of WPCN of time reversal OFDM is simulated and analyzed, and compared with the traditional optimization algorithm. Assuming that all channels conform to Rayleigh fading characteristics, the channel gain model is $10^{-3} d_m^{-\alpha}$, where d_m represents the distance from the mth terminal to the AP node, and $\alpha = 3$ is the channel fading index. Suppose PB is placed in (0,0),AP is placed in (0,5), and the terminal is placed in area of (-5,-20) and (-5,20). The bandwidth of the system is 180KHz, the number of subcarriers is 11, the energy efficiency is 0.8, and the noise power is -50dBm.

Figure 3 is a comparison of different algorithms for system sum rate convergence performance. It can be seen from Figure 4.3 that as the number of iterations increases, the sum rate of the three algorithms also increases gradually. Finally, the algorithm proposed in this paper has the maximum sum rate after convergence. The reason is that when the energy transmitting node and the information receiving node are placed together to form a hybrid node, a double-far and near phenomenon will occur, resulting in a larger path loss. In this paper, because the time reversal technology can make full use of multipath to focus the signal energy on the target point, and improve the receiving power of the target user when the transmit power of the terminal node is the same, the algorithm proposed in this paper has better system sum rate.

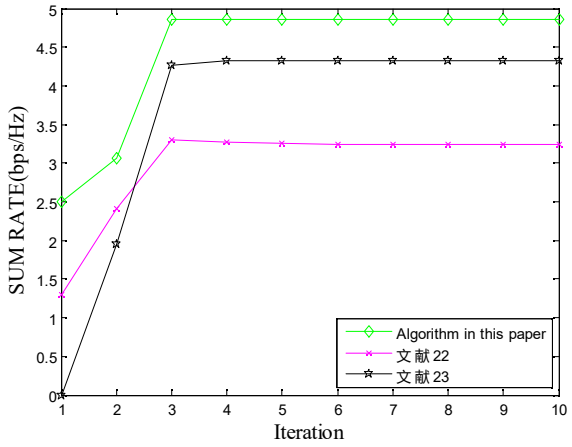


Figure 3 Graph of iterative algorithms for different schemes

Figure 4 shows the functional relationship between system sum rate performance of WPCN with respect to time allocation factor τ. It can be seen from the three-dimensional diagram that the sum rate changes with energy transmission time τ_0 and time reversal detection stage τ_{tr}. It can be clearly seen from Figure 4 and Figure 5 that in order to make the system

and rate large enough, the TR detection time τ_{tr} should be as short as possible while ensuring that the farthest terminal can complete TR detection. This is because when τ_{tr} is small, the time allocated to the energy transfer phase can be as much as possible, and the AP node can obtain enough energy from the RF signal transmitted by the PB node during the duration of the energy transfer. From Figure 4 and Figure 5, it can be seen that the τ_0 of the proposed OFDM WPCN algorithm is about 0.5s;

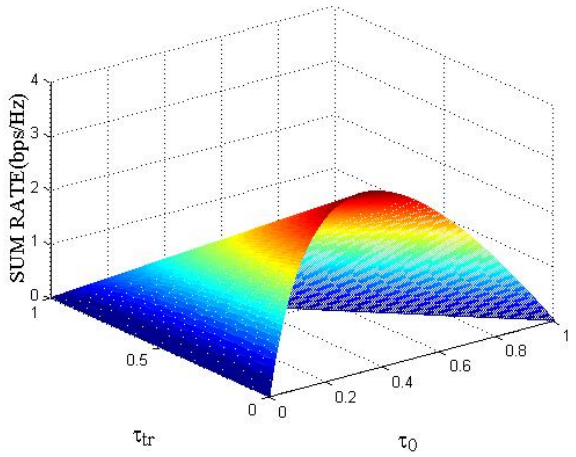


Figure 4 Sum rate and time relation diagram

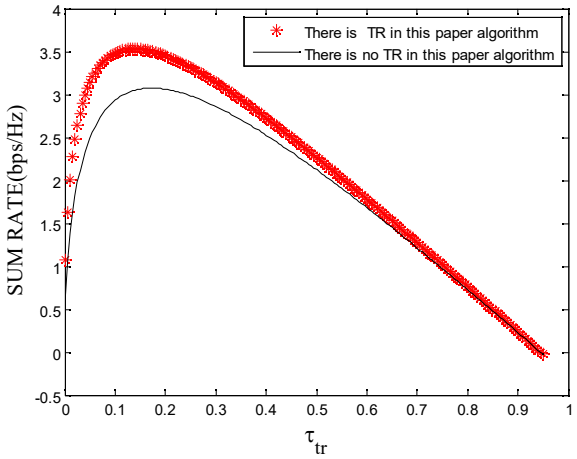


Figure 5 Sum rate and time reversal stage relation diagram

Figure 6 shows the curve of the system sum rate varying with the transmission power.Figure.6 is a simulation diagram of the variation of different scheme sum rates with power when the number of AP node antennas is 4. Figure 7 is a graph showing the variation of the system sum rate with the transmit power of the scheme in this paper when the number of multipaths in the system is different. It can be seen from Figure 6 and Figure 7 that the system sum rate of the proposed scheme and the comparative scheme both increase with the increase of the transmit power. This is because when the transmit power increases, the power

received by the receiver is greater than the noise power, so the signal-to-interference-to-noise ratio of the receiver also increases, so the sum rate of the system increases with the increase of the transmit power. And because the space-time focusing of time reversal can make full use of the multipath to make the signal energy coherent superposition at the receiver, the signal strength received by the terminal is greatly increased, so the scheme in this paper can effectively improve the system sum rate. It can be seen from figure 7 that with the increase of the number of multipaths, the system and rate also increase. The reason is that with the increase of the number of multipaths, the time reversal technique uses multipaths to provide additional spatial degrees of freedom for the system. The richer the multipath, the better the focusing effect, the stronger the energy of the received useful signal, and the better system and rate are obtained, so the effectiveness of the algorithm in this paper is verified.

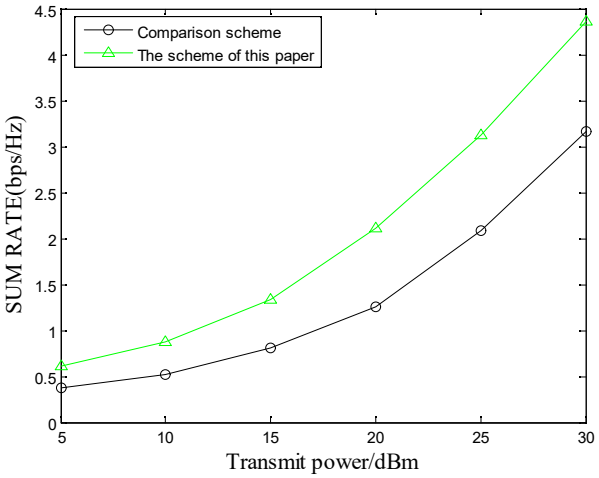


Figure 6 Diagram between sum rate and transmission power

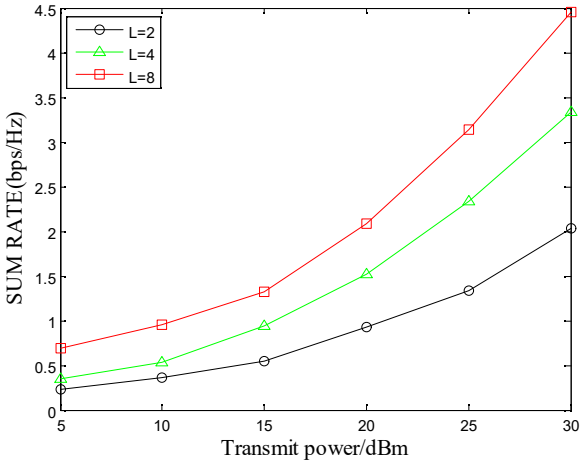


Figure 7 Diagram of sum rate with transmission power for different multipaths

Figure 8 shows the effect of the distance from the AP to the terminal on the sum rate of different algorithms when Pmax=30dBm, assuming that other parameters are the same. The sum rate of the two algorithms decreases with the increase of distance, because the path loss of WET and WIT increases with the increase of distance. The performance of the proposed algorithm is significantly better than that of the comparison algorithm. After the signal is

processed by TR technology, most of the energy of the signal is focused on the AP node, combined with the algorithm proposed in this paper can effectively increase the system sum rate. The system sum rate will decrease with the increase of distance, because the larger the distance, the more power consumption of the system, which leads to the lower system sum rate.

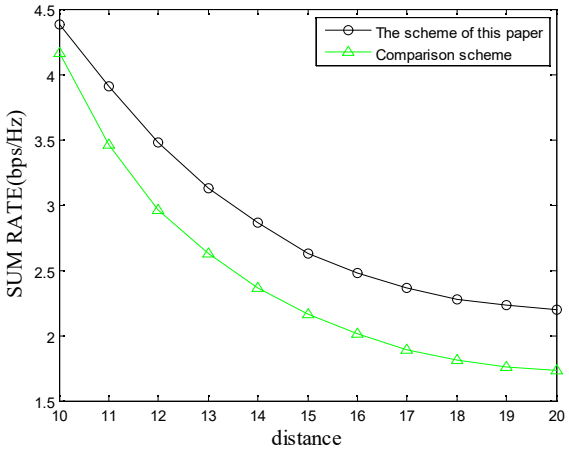


Figure 8 Diagram between sum rate and distance between transceiver and transceive

5. CONCLUSIONS

This paper studies the WPCN of OFDM, where APs and terminal nodes EH from the RF signal of the PB, and then use the harvested energy to transmit independent signals in the downlink to multiple terminal nodes. After introducing the time reversal technology to suppress the interference of the system, the transmit power, energy transmission time and subcarrier allocation of AP nodes are jointly optimized, and a maximizing system sum rate algorithm is proposed. Using the quasi-concave form of the proposed problem, the energy transfer time and optimal transmit power are obtained based on a dichotomy search. The numerical results show that the algorithm proposed in this paper is effective. And the influence of transmit power, AP-to-terminal distance and multipath on the system sum rate is analyzed. The simulation results prove that the scheme proposed in this section can suppress interference and help to improve the sum rate of the system. The WPCN system has broad application prospects and can be used in wireless sensor networks. Future research can allocate and optimize resources for multi-antenna HAP and multi-user complex multi-cell environments and vehicle networking environments. The focusing intensity of time reversal technology depends on channel estimation. When there is an error in channel estimation, it will affect the focusing of time reversal. The scheme of this paper assumes that the channel state information is perfect, but in practice, the channel estimation will be affected by many factors in the wireless environment. Therefore, when there is an error in channel estimation, the performance of the system will be affected.

ACKNOWLEDGEMENTS

This work is supported by the Natural Science Foundation of China (NSFC) grant funded by the China government (61771084).

APPENDIX A
PROOF OF LEMMA 1

Existence function $h(x)=\frac{g(x)}{ax+b}$

$$\frac{dh}{dx}=\frac{(ax+b)g'(x)-ag(x)}{(ax+b)^2} \tag{23}$$

Because $ax+b>0$,if $h'(x)=0$ is satisfied at this time, then $(ax+b)g'(x)-ag(x)=0$

$$\frac{d^2h}{dx^2}=\frac{d}{dx}\left(\frac{dh}{dx}\right)=\frac{(ax+b)g''(x)-\left(2a(ax+b)\left((ax+b)f'(x)-af(x)\right)\right)}{(ax+b)^4} \tag{24}$$

$$\frac{d^2h}{dx^2}\Big|_{h'(x)=0}=\frac{(ax+b)g''(x)}{(ax+b)^4} \tag{25}$$

It can be seen from the above that at any point where the slope is zero, when $g''(x)\le 0$ then the second derivative of $h(x)$ is nonnegative.当$g''(x)\le 0$时，$h''(x)\ge 0$， When $g''(x)\le 0$, then $h''(x)\ge 0$, which means that when $g(x)$ is convex, $h(x)$ is quasiconvex. When $g(x)$ is concave, $h(x)$ is quasi-concave.

APPENDIX B

According to the KKT condition, the analytical solution of optimal power allocation can be obtained. According to the KKT condition, the partial derivative of $p_{m,n}$ is zero.

$$\frac{\partial L}{\partial p_{m,n}}=\frac{(1-\mu)\eta BP_T(1-\tau_{tr})A_{m,n}}{N(1+p_{m,n}A_{m,n})\mathrm{In}2}+\mu\theta=0 \tag{26}$$

$$p_{m,n}=\frac{(\mu-1)\eta BP_T(1-\tau_{tr})}{\mu\theta N\mathrm{In}2}-\frac{1}{A_{m,n}} \tag{27}$$

Since the power cannot be negative, write the above formula as follows：

$$p_{m,n}=\left[\frac{(\mu-1)\eta BP_T(1-\tau_{tr})}{\mu\theta N\mathrm{In}2}-\frac{1}{A_{m,n}}\right]^+ \tag{28}$$

Where $\left[a\right]^{+} = \max\left(a, 0\right)$ and γ is constant.

REFERENCES

[1] K. W. Choi, P. A. Rosyady, L. Ginting, A. A. Aziz, D. Setiawan, andD. I. Kim,‘‘Theory and experiment for wireless-powered sensor networks:How to keep sensors alive,’’ IEEE Trans. Wireless Commun., vol. 17, no. 1,pp. 430–444, Jan. 2018.

[2] Z. Chu, F. Zhou, Z. Zhu, R. Q. Hu, and P. Xiao,‘‘Wireless powered sensor networks for Internet of Things: Maximum throughput and optimal power allocation,’’ IEEE Internet Things J., vol. 5, no. 1, pp. 310–321, Feb. 2018.

[3] R. Zhang and C. K. Ho,‘‘MIMO broadcasting for simultaneous wireless information and power transfer,’’ IEEE Trans. Wireless Commun., vol. 12, no. 5, pp. 1989–2001, May 2013.

[4] D. Sui, F. Hu, W. Zhou, M. Shao, and M. Chen,‘‘Relay selection for radio frequency energy-harvesting wireless body area network with buffer,’’IEEE Internet Things J., vol. 5, no. 2, pp. 1100–1107, Apr. 2018.

[5] C. K. Ho and R. Zhang,“Optimal energy allocation for wireless communications with energy harvesting constraints,” IEEE Trans. Signal Process., vol. 60, no. 9, pp. 4808–4818, Sep. 2012.

[6] H. Ju and R. Zhang,“Throughput maximization in wireless powered communication networks,” IEEE Trans. Wireless Commun., vol. 13, no. 1,pp. 418–428, Jan. 2014.

[7] X. Wang, Z. Nan, and T. Chen,‘‘Optimal MIMO broadcasting for energy harvesting transmitter with non-ideal circuit power consumption,’’ IEEE Trans. Wireless Commun., vol. 14, no. 5, pp. 2500–2512, May 2015.

[8] J. Xu, L. Liu, and R. Zhang,“Multiuser MISO beamforming for simultaneous wireless information and power transfer,” IEEE Trans. Wireless Commun., vol. 62, no. 18, pp. 4798–4810, Sep. 2014.

[9] X. Zhou, J. Guo, S. Durrani, and M. Di Renzo,‘‘Power beacon-assisted millimeter wave ad hoc networks,’’ IEEE Trans. Commun., vol. 66, no. 2,pp. 830–844, Feb. 2018.

[10] H. Ju and R. Zhang,“Throughput maximization in wireless powered communication networks,” IEEE Transactions on Wireless Communications, vol. 13, no. 1, pp. 418–428, 2014.

[11] Z. Hadzi-Velkov, N. Zlatanov, and R. Schober,“Multiple-access fading channel with wireless power transfer and energy harvesting,” IEEE Communications Letters, vol. 18, no. 10, pp. 1863–1866, 2014.

[12] Z. Hadzi-Velkov, I. Nikoloska, G. K. Karagiannidis, and T. Q. Duong,“Wireless networks with energy harvesting and power transfer: Joint power and time allocation,” IEEE Signal Processing Letters, vol. 23,no. 1, pp. 50–54, 2016.

[13] H. Lee, K.-J. Lee, H. Kim, B. Clerckx, and I. Lee,“Resource allocation techniques for wireless powered communication networks with energy storage constraint,” IEEE Transactions on Wireless Communications,vol. 15, no. 4, pp. 2619–2628, 2016.

[14] D. Xu and Q. Li,“Optimization of wireless information and power transfer in multiuser ofdm systems,” AEU-International Journal ofElectronics and Communications, vol. 90, pp. 171–174, 2018.

[15] “Joint power control and time allocation for wireless powered underlay cognitive radio networks,” IEEE Wireless Communications Letters, vol. 6, no. 3, pp. 294–297, 2017.

[16] H. Kim, H. Lee, M. Ahn, H.-B. Kong, and I. Lee,"Joint subcarrier and power allocation methods in full duplex wireless powered communication networks for ofdm systems," IEEE Transactions on Wireless Communications, vol. 15, no. 7, pp. 4745–4753, 2016.

[17] T. P. Do and Y. H. Kim,"Resource allocation for a full-duplex wireless powered communication network with imperfect self-interference cancelation," IEEE Communications Letters, vol. 20, no. 12, pp. 2482–2485,2016.

[18] C.-H. Lee,''Wireless information and power transfer for communication recovery in disaster areas,'' in Proc. IEEE Int. Symp. World Wireless,Mobile Multimedia Netw., Jun. 2014, pp. 1–4.

[19] Z. Chu, F. Zhou, Z. Zhu, R. Q. Hu, and P. Xiao,''Wireless powered sensor networks for Internet of Things: Maximum throughput and optimal power allocation,'' IEEE Internet Things J., vol. 5, no. 1, pp. 310–321, Feb. 2018.

[20] X. Zhou, J. Guo, S. Durrani, and M. Di Renzo,''Power beacon-assisted millimeter wave ad hoc networks,'' IEEE Trans. Commun., vol. 66, no. 2,pp. 830–844, Feb. 2018.

[21] M. Lei, X. Zhang, T. Zhang, L. Lei, Q. He and D. Yuan, "Successive Interference Cancellation for Throughput Maximization in Wireless Powered Communication Networks," 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), 2016, pp. 1-6, doi: 10.1109/VTCFall.2016.7881241.

[22] Li Qun, Wang Longqi, Xu Ding. Antenna Selection and Resource Allocation in Wireless Powered Communication Networks with Self-Energy Recycling[C]//2018 IEEE/CIC International Conference on Communications in China (ICCC). Beijing: IEEE Press, 2018: 824-828.

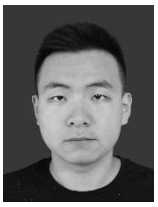
[23] Kim J., Lee H., Park S H., et al. Minimum rate maximization for wireless powered cloud radio access networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(1): 1045-1049.

Authors

Wei Liu Born in 1994, from Inner Mongolia,China. Now he is a master's student of Chongqing University of Posts and telecommunications. His main research interests are wireless power communication network and time reversal.

Fang Wei Li Born in 1960 in Chongqing, China. Professor and doctor of Chongqing University of Posts and telecommunications. His main research interests are electromagnetic field and electromagnetic wave, wireless network security, wireless transmission theory and technology.

Hai Bo Zhang Born in 1979 in Chongqing, China. Associate professor and master's supervisor of Chongqing University of Posts and telecommunications. His main research interests are vehicle networking, security authentication, key negotiation and so on.



Bo Li received the B.S. degree from Shanxi Institute ofEngineering and Technology, China, in 2018.Currently studying for a master's degree inChongqing University of Posts and Telecommunication. His major is informationand communication engineering. His research interest covers Internet of Thingsand wireless power communication network.

A SYSTEMATIC LITERATURE REVIEW ON INSECT DETECTION IN IMAGES

Lotfi Souifi[1], Afef Mdhaffar[1,2], Ismael Bouassida Rodriguez[1], Mohamed Jmaiel[1,2] and Bernd Freisleben[3]

[1]ReDCAD Laboratory, ENIS, University of Sfax, B.P. 1173 Sfax, Tunisia
[2]Digital Research Center of Sfax, 3021 Sfax, Tunisia
[3]Dept. of Math. & Comp. Sci., Philipps-Universität Marburg, Germany

ABSTRACT

Due to the advancements of deep learning (DL), particularly in the areas of visual object detection and convolutional neural networks (CNN), insect detecetion in images has received a lot of attention from the research community in the last few years. This paper presents a systematic review of the literature on the topic of insect detection as a case of object detection in images. It covers 50 research papers on the subject and responds to three research questions: i) type of dataset used; ii) detection technique used; iii) insect location. The paper also provides a summary of existing methods used for insect detection.

KEYWORDS

Systematic Literature Review (SLR), Deep Learning (DL), Object Detection, Insect Detection.

1. INTRODUCTION

Agriculture is the first human activity that enabled mankind to progress and develop. Agriculture and the food industry are the most important activities in the world today, owing to the world's growing population and its increasing need for food [1], [2]. Insects have long been regarded as a serious crop threat. Insects have the primary effect of reducing the amount of food available to people by lowering agricultural productivity. This might decrease the quantity and the quality of food [3]. For agricultural pest forecasting, insect detection is critical. Agricultural professionals detect insect infestations based on daily observations. This manual process costs farmers a lot of time and money. To reduce the use of risky and expensive chemical products, early detection and monitoring of insects are necessary for taking the proper action and determining whether the insects are dangerous or not. As a result of the advancement of deep learning (DL), particularly in image processing, several methods for insect detection have been proposed. In the field of" insect detection", there is only one review of the semantic literature. Amarathunga et al. [4] wrote this SLR in 2021, where 2021 publications are not considered. The authors [4] mention that the publication window is only between 2010 and 2020. The article clarifies classification methods but does not discuss insect detection methods. Therefore, in addition to using the most recent detection and classification techniques, our paper also gives more attention to the gathering and preparation of the used dataset.

2. SYSTEMATIC LITERATURE REVIEW

We perform a systematic literature review following five steps:

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 41-56, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122003

- Define research questions,
- Identify papers that are related to this topic,
- Add papers to the list containing titles and abstracts,
- Remove non-relevant papers,
- Identify the capabilities of each studied paper to answer the research questions.

2.1. Research Questions

The research questions that need to be answered by our study are as follows:

- RQ1: What type of datasets of insects are used?

This question focuses on the type of datasets used for insect detection (i.e., collected for the purpose of this paper or publicly available)

- RQ2: Which object detection method is used for insect detection?

A response to this question presents the commonly used technique for insect detection and the modification added to it. We notice that in some cases, the studied article presents a combination of detection and classification of insects.

- RQ3: What is the location of the studied insect?

RQ3's response typically includes a description of the study's location. It may not be mentioned in every instance. In this case, we consider the first author's research institute or university's location to be an insect's location.

2.2. Methodology

Our search methodology follows two phases. First, a group of keywords is defined based on the research questions. Second, the selected keywords are aggregated using AND and OR operators to formulate the results below:

("deep learning" AND "insect detection" AND "object detection" AND "insect detection" AND ("insect detection" OR "insect pest detection")).

2.3. Selection Criteria

This section presents the used selection criteria.

- Inclusion Criteria (IC):
  - publications that match one of the search items,
  - research studies from journals,
  - conferences studies that were published from January 2012 to March 2022.
- Exclusion Criteria (EC):
  - publications that were published before (or on) 31.12.2011,
  - publications that are not related to the research questions (i.e., insect detection), but appeared in the search,
  - language (only English is taken).

2.4. Data Collection

Figure 1 shows our search methodology. It follows three main steps. The first one (i.e., illustrated by the left side) describes the research findings for each of the search engines that were used. As a result, 407 articles were selected for our study. The 5 triangles illustrate the

second step. They describe the elimination criteria and the number of articles that have been removed. The final step (i.e., illustrated by the right side) displays the 50 articles that met the exclusion criteria in total, along with how they were distributed among the search engines.

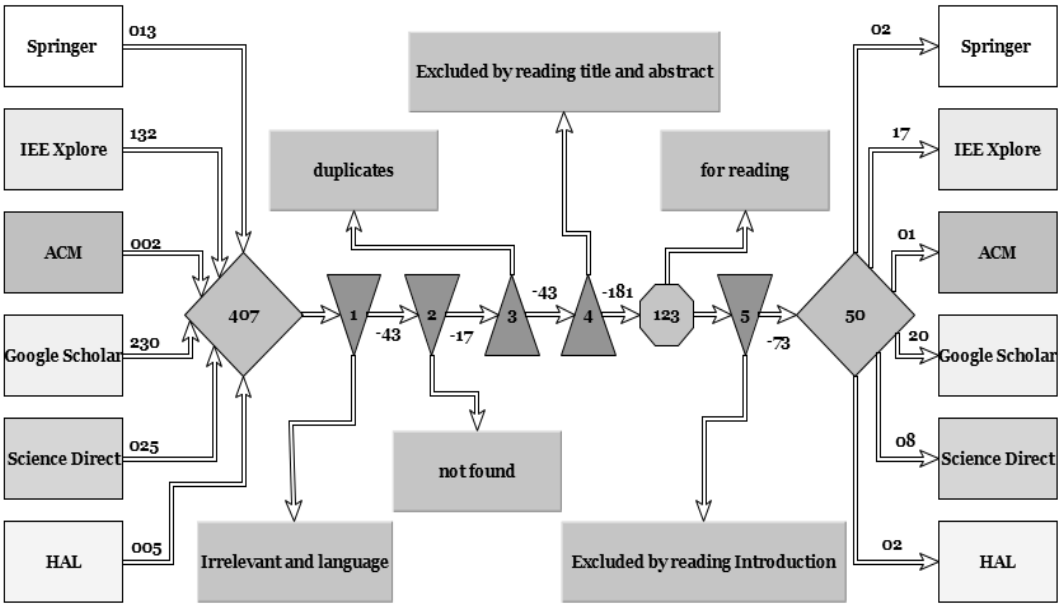


Figure 1. Data collection

3. RESULTS AND DISCUSSION

Our paper answers three different questions with respect to insect detection. Before we start discussing the questions, it is important to understand some basic concepts such as insect and insect pest.

3.1. Basic Concepts

Hill [5] defines pests as follows: any animal (or plant) that causes harm or damage to humans. Even if they are just causing annoyances, such animals or crops qualify as pests. Even if it does not belong to a pest species, an animal or plant taken out of the context is considered a pest (individually) in agricultural terms. The same author [6] later defines a pest as follows: in the broadest sense, a pest is an insect (or organism) that causes harm to humans, livestock, crops, or personal property. The key word is harm, which is usually translated as damage, which can be measured (often quantitatively) in many cases. Furthermore, damage is frequently equated to monetary losses. Nuisance and disturbance are examples of harm at the most basic level of interpretation. Thus, a buzzing mosquito at night can keep you awake, and face flies in tropical Africa can be very distracting and reduce your productivity. Insects are referred to as pests in our study.

3.2. RQ1: Type of Data Sets of Insects

Since the data set is very important for deep learning models, it is common to discuss the used data before the used model.

3.2.1. Web Sources

Articles follow this approach [7]– [10], and each of the previous articles uses different methods. However, the fact that they all focus on well-known insects is what unites them (e.g., spiders, mosquitoes, etc). The most well-known collection methods involve the use of search engines like Google [11] and Bing [12] or the use of websites like Flickr [13] as the standard gathering techniques. While Butera et al. [7] employ all three techniques, other authors [8], [9] use Google and Flickr, and some authors [10] do not mention the techniques employed.

Using web sources is relatively fast, but not suitable for all insects. Some insects may not be found (not the same name in all countries) or lack photos of the target insect. Combining web sources and on-site images could be the answer to this issue.

3.2.2. Combination of Web Sources and On-Site Images

Combining web sources and on-site images could be more effective. When the number of photographs is insufficient, this approach requires more time and resources than using web sources only [14]– [16]. According to Takimoto et al. [14], field collection took two years (2017 and 2018). While Hossain et al. [15] and Abeywardhana et al. [16] do not specify how much time has passed, Takimoto et al. [14] mentioned the use of a RICOH WG-4 digital camera alongside the Google search engine.

However, Hossain et al. [15] utilized a mobile phone to shoot images in the field along with the Google search engine. The web source is not mentioned by Abeywardhana et al. [16], although a digital camera and a mobile phone were used.

3.2.3. On-Site Images (Field or Laboratory)

Some researchers collect data in the field, even though it is the hardest method compared to the other approaches. Different techniques were used to collect data, such as different types of traps, including yellow traps [17], [18], pheromone traps [19]– [21], yellow boards [22], and smart traps [23], [24] with an integrated camera. Other researchers use cameras and mobile phones to take photos of insects directly in the laboratory or field. In Lyu et al. [25], the pictures of the desired insects were taken in the laboratory. While Yang et al. [26] used Zhongwei Kechuang industrial cameras installed in field, other authors [27]– [29] used digital mobile cameras. Currently, Du et al. [30] use high-resolution UAV, and Bjerge et al. [31] use previously constructed portable computer versions, while Ard et al. [32] use a Scoutbox along with a camera, and Rustia et al. [33] use a wireless image monitoring system. It took them two years and two months to collect all their data. Finally, Chen et al. [34] use a small smart car.

Collecting on-site images is still the most used approach for two reasons. First, a few open datasets are suitable for multi-class pest detection, and there is a lack of datasets that is suitable for a particular study goal. Menikdiwela et al. [9] explain why it is not a good idea to use an existing dataset. Because the spider's scale is quite large in comparison to the background in those images, they do not use any of the spider images from the ImageNet dataset. Second, the trained model will be more robust if the dataset is as close as possible to a real scene in the natural environment.

3.2.4. Existing Image Datasets

The simplest method is to just take existing data, which in several cases is pre-processed. There are different types of existing data sets:

- Data collected by the farmer but not processed, e.g., Nam et al. [35],
- The use of well-known datasets, e.g., IP 102 proposed by Wu et al. [36], which was used by several authors [37]– [39], and the 10c and 5c datasets, which were used by Liu et al. [40],
- Datasets gathered by research institutions, such as the Moth Classification and Counting (MCC) dataset and European Moths dataset (EU-Moths), which were used by Korsch et al. [41], a dataset collected by the Laboratory of Medical Zoology, which was used by Luo et al. [42], and a dataset collected by the Hefei Institute of Physical Science (under the name of pest24), which was used by Liu et al. [43],
- datasets gathered by other researchers, e.g., Mamdouh et al. [44] used the Dacus Image Recognition Toolkit (DIRT) data set collected by Kalamatianos et al. [45], Tanjim et al. [46] used datasets gathered by Rajan et al. [47], and Deserno et al. [48] used the yellow stick trap data set gathered by Ard et al. [49],
- the Kaggle contest data set, used by SutHo et al. [50].

Since all images have already been processed, this is the most straightforward approach. Nevertheless, it is still insufficient for all use cases and cannot contain all insect species.

3.2.5. Combination of Existing Image Data Sets and Other Methods

Some articles combine more than one approach, such as Cabrera et al. [51] who use an existing data set called Yellow Sticky Trap data set along with a web data set. Also, Mazare et al. [52] use existing data photo libraries taken by the beneficiary institute and collected from traps. Xia et al. [53] utilize data already collected by Xie et al. [54] and data downloaded from the Internet. Huang et al. [55] use a combination of three methods collected from an e-trap and data from the Internet from iNaturalist [56], and an existing data set IP102 [36] and ImageNet (proposed by Deng et al. [57]).

There are two articles [58], [59] that present recent datasets that professionals have gathered and annotated. Wang et al. [58] present AgriPest, with a total of 49.000 images and a total of 264.000 objects. This dataset was collected in the field and labelled by 20 agricultural experts. Li et al. [59] mention that their dataset was collected in a warehouse under the name of RGBInsect, and they used smartphones with traps. The labelling was done by experts. RGBInsect has a total of 7.514 images and more than 159.000 insect instances.

Table 1. The percentage of each used type of dataset in all articles

Dataset type	**(%)**	**Data set sub-classes**	**(%)**
Collected datasets	56	Web source	8
		Combination of web sources and on-site images	8
		On-site (field or laboratory)	40
Existing datasets	44	Existing datasets	30
		Combination of existing data sets and other methods	14

Before discussing the next research question, some points need to be clarified such as data labelling techniques and data augmentation.

After collecting a data set, all images must be labelled. This task should be conducted by experts or some automated tools. Our paper describes the used labelling tools and the human-based labelling methods.

In the studied articles, only 11 of them mentioned the tool used to label the images. Nam et al. [35] mentioned that they used BBOX [60] while other authors [10], [14], [17], [25], [28], [33], [44] used LabelImg [61] and further authors [27], [30], [50] used LabelMe [62]. Some articles [35], [42], [44] mention that the labelling is done by experts, and Rustia et al. [33] state that labelling was done by experts and entomologists, while Ding et al. [19] mention technicians.

Data augmentation techniques (DA) are used in 54% of the studied papers. Insufficient data for model training and testing is the cause of DA, which can result in over-fitting (which happens when a statistical model matches its training data close to perfection. Unfortunately, when this occurs, the algorithm's goal is defeated because it cannot accurately perform against unobserved data.) or under-fitting (i.e., a data model has a high error rate on both the training set and unobserved data because it cannot accurately represent the relationship between the input and output variables. It happens when a model is overly simplistic, i.e., when there is insufficient regularization, training time, or input features). Rotation and zooming are the most used techniques for DA in general. Only Liu et al. [40] did not specify which method they employed. Rotation is used by Ding et al. [19], whereas reflection is used by Khalifa et al. [37]. Some articles use more than the first two, such as adding noise [8], scaling [10], and flipping and colour adjustments [18, 34].

3.3. RQ2: Object Detection Method Used for Insect Detection

The discussion of the detection methods presented in the chosen articles will now proceed, but first, let's define the widely employed method.

3.3.1. Convolutional Neural Networks CNNs

Before discussing the employed methods, it is necessary to first review some fundamentals of deep learning.

O'Shea et al. [63] state that Convolutional Neural Networks (CNNs) are a type of deep artificial neural network commonly used in image analysis. A CNN can learn spatially related features by treating an image as a volume. The volume of images in a CNN is transformed using a variety of specialized layers. Most of the computation for classifying an image is done by a convolutional layer. A convolutional layer contains a series of kernels that move or convolve over an image volume. The ability of CNNs to recognize textures, shapes, colours, and other image features as their training progresses is one of their most significant advantages.

3.3.2. Object Detection Techniques

This section presents commonly used object detection techniques. In 2015, Girshick et al. [64] presented the Faster Region-based Convolutional Network method (Faster R-CNN) that takes as input an entire image and a set of object proposals. To create a feature map, the network first processes the entire image with several convolutional networks. Then, it extracts a fixed-length feature vector from the feature map. This model improves training and testing speed with increasing detection accuracy.

Liu et al [65] presented the Single Shot multi-box Detector (SSD) algorithm as a one-stage representative detection algorithm, which has an obvious speed advantage compared to the two-stage algorithm. Because of its good accuracy, SSD has become one of the main algorithms studied at present.

For a unified detector, Redmon et al. [66] proposed YOLO, which casts object detection as a regression problem from image pixels to spatially separated bounding boxes with associated class probabilities. Unlike the other approaches, YOLO does not include a stage for generating region proposals. As a result, YOLO uses a small set of candidate regions to detect objects directly. YOLO generates C class probabilities, B bounding box locations, and confidence scores by dividing an image into an S*S grid. In the articles under consideration, the following versions were used: YOLO V3 proposed by Redmon et al. [67], YOLO V4 proposed by Bochkovskiy et al. [68], and YOLO V5 [69].

3.3.3. Classification Techniques

Finally, we define some of the commonly used classification methods that have been mentioned in several articles.

Support Vector Machines (SVM) [70] are used for classification in supervised machine learning. A SVM can deal with issues like high dimensionality, small sample sizes, and more. It is often used for forecasting [71] and for resolving nonlinear data modelling issues [72].

In 2012, Krizhevsky [73] designed a new CNN model called AlexNet. It achieved a top-5 error of 15.3 in the Large-Scale Visual Recognition Competition (ILSVRC). AlexNet is composed of 8 layers. The first 5 layers are convolutional layers. Some of them are followed by max-pooling layers. The last 3 layers are fully connected layers.

In 2014, Simonyan et al. [74] proposed VGG. By stacking a few tiny convolution kernels and max-pooling layers, it increased the representation depth of the network. The structure's simplicity offered the benefit that the network performance could be enhanced by adding more depth. However, VGG makes use of more parameters, which consumes more memory. In the articles under consideration, the following versions were used: VGG16 and VGG19.

In 2015, the new state-of-the-art for classification and detection in ILSVRC14 was achieved by Szegedy et al.'s CNN model [75], called GoogleNet. It is a transfer learning CNN with 22 layers, also known as Inception v1. The inception layer is the core concept of GoogleNet. Concatenating a 1 x 1 convolutional (Conv) layer, a 3 x 3 convolutional layer, and a 5 x 5 convolutional layer into a single output vector is what makes up the inception layer. In the articles under consideration, the following versions were used: Inception and InceptionV2.

In 2016, He et al. [76] proposed ResNet. ResNet is one of the most used networks for classification and object detection tasks. The underlying idea are the residual blocks, which aim to simplify the training of neural networks characterized by many layers. Based on ResNet, Xie et al. [77] proposed ResNeXt, which uses the idea of increasing the cardinality. Compared with ResNet, it has the same parameters but higher accuracy. In the articles under consideration, the following versions were used: ResNet, ResNet18, and ResNet50.

In 2016, Iandola et al. [78] designed a new CNN model called SqueezeNet. The authors intention with SqueezeNet was to develop a smaller CNN with fewer parameters that could more readily fit into computer memory. The authors use SqueezeNet to reduce model size by 50* (5 MB) compared to AlexNet (240 MB) of parameters while maintaining or improving Alex-Net's top-1 accuracy.

In 2018, Sandler et al. [79] presented MobileNet as a deep convolutional architecture for mobile phones. It has a much smaller architecture and less calculation complexity than popular object

detector models like the R-CNN. In the articles under consideration, the following versions were used: MobileNet and MobileNetv2.

3.3.4. Existing Detection Techniques

YOLOv5 is used by several authors [44], [51], [80]. YOLOv4 is used by both Genaev et al. [27] and Chen et al. [34], where the former uses the regular version and the latter [30] uses the tiny version. Finally, several authors [10], [29], [31], [81], [82] use YOLOv3, also known by the use of Dark-Net backbone. We also notice that YOLO has been used in combination with other classification techniques, such as Takimoto et al. [14] used YOLOv4 and Efficient-Net, and Kuzuhara et al. [8] used YOLOv3 and Xception (proposed by Chollet et al. [83]), and Liu et al. [43] used YOLOv3 with Global Context (GC) Network (proposed by Cao et al. [84]). Finally, Rustia et al. [33] used a small version of YOLOv3 that is appropriate for small devices, as well as two different classifiers; the names of the two classifiers are not mentioned.

Faster R-CNN is the second most popular technique [39],[59]. Faster R-CNN is frequently combined with other classification techniques. It is used by Ramalingam et al. [85] with ResNet, [32], [86] with Inception ResNet [87], while Rong et al. [22] used Mask R-CNN (which is improved version of Faster R-CNN, proposed by He et al. [88]) with the ResNet backbone.

The last detection technique used is SSD. In some cases, it is used alone [25], [41], [89] while Nam et al. [35] use SSD with VGG16. Finally, Patel et al. [28] used a combination of three techniques: Faster R-CNN and SSD with two different feature extractors: Inception and MobileNet.

3.3.5. Existing Classification Techniques

In the case only for classification techniques, SutHo et al. [50] use MobileNetv2, Luo et al. [42] use Inception-Net v3, Abeywardhana et al. [16] use SqueezeNet, Porrello et al. [90] use ResNet, Roosjen et al. [24] use ResNet18, and Menikdiwela et al. [9] utilize VGG16 fin-tuned on spiders. In the case of multiple techniques combined, Xia et al. [53] use VGG19 with a Region Proposal Network (RPN), Rajan et al. [47] use SVM along with a prepossessing algorithm, Chen et al. [86] use an improved Retina-Net (proposed by Lin et al. [91]) and convolutional block attention module (CBAM). Wang et al. [38] use ResNet50 with Features Pyramid Network (FPN).

Several articles use transfer learning, such as Khalifa et al. [37] who use 3 classification techniques: AlexNet, GoogleNet, and SqueezeNet. Lie et al. [40] and Hong et al. [16] present 7 techniques that have been transferred, while Butera et al. [7] use 12 techniques. Wang et al. [58] use 6 techniques to validate the proposed data set.

3.3.6. Proposals of New Techniques

Six of the articles presented novel techniques. Ding et al. [19] demonstrated ConvNet, which is based on a standard CNN. Yang et al. [26] introduced the Multi-layer Convolutional Structure (MCSNet), a detection and classification model comprising three parts: a VGG16-based insect features subnet, a region proposal network (RPN), and a classification network. Region CNN (R-CNN) is used to build the model's overall architecture. MAMPNet, a Multi-Attention and Multi-Part convolutional neural network based on ResNet50, is presented by Huang et al. [55]. Du et al. [30] present Pest R-CNN, which is based on the classical object detection model, Faster R-CNN. Du et al. [30] also mention that their model is divided into three parts: the same as the previous model, but with improved feature extractors.

Liu et al. [21] propose Pest-Net, a region-based end-to-end approach that is also made up of three parts: the Channel Spatial Attention (CSA), a Region Proposal Network (RPN), and a Position-Sensitive Score Map (PSSM) used to replace the Fully Connected layer. The backbone used is based on CNN, but it does not mention which one exactly. The CSA consists of two parts: the 3D feature map and 1D feature vectors.

The use of existing techniques does not mean that no modifications have been made. Menikdiwela et al. [9] change the last layer from a 1000 output layer to two output layers (spider and non-spider). Lyu et al. [25] add a top-down module to the SSD used. SutHo et al. [50] add two fully connected layers to the existing model. Geneav et al. [27] change the backbone of the used YOLO. Yuan et al. [80] changes the case-based learning (CBL) of YOLOv5. Chen et al.[86] add an improved full convolutional network (FCN) with CBAM for detection and classification.

There are two articles that used existing techniques to propose an entire system for detection and classification based on traps along with monitoring systems (cameras). The articles are Junior et al. [17] and Bjerge et al. [23]. Junior et al. [17] propose InsectCv based on Mask R-CNN, while [23] propose an automated light trap to monitor moths and mentions the use of a CNN model.

Only two of the studied articles do not mention the name of the used models. Hossain et al. [15] mention CNN in general, and Mazare et al. [52] mention artificial neural networks.

Numerous types of insects, including spiders [9], wheat pests [25], and others, were found in the studied articles. There was an insect species called the moth that was present in numerous studies, and it has been a serious problem for farmers around the world. This insect has been detected in several publications [19], [23], [28], [30], [41], [50], [52]. whereas Patel et al. [28] detect three insects where two of them are moths, and Korsch et al. [41] detect this type of insect in 200 species.

Some articles add the count of the insect to their model [17], [19], [22], [44], [50], [55] to determine the effectiveness of their installed trap.

In the studied papers, the number of detected insects is variable:

- Some articles only detect one insect or one class: [9], [19], [23], [30], [41], [42], [44], [50], [52], [82], [90],
- Others detect two classes: [8], [14], [26], [46],
- Other papers detect 3 classes: [7], [20], [27], [28], [32], [48], [51],
- Other articles detect 4 classes: [17], and [89],
- Some of the studied papers detect 5 classes: [35], [10], and [39],
- There are some papers that detect 6 classes: [15], [16], [18], [25], [34],
- Other papers detect from 7 to 9 classes: [24] (7 classes), [31], [37], and [85] (8 classes), and [22] (9 classes),
- Some papers detect from 10 to 19 classes: [38] (10 classes), [86] (11 classes), [81] (12 classes), [55], and [40] (15 classes), [21] (16 classes), and [80] (17 classes),
- Only 3 articles detect more than 20 classes: [43], [53] (24 classes), and [29] (23 classes).

Since there are fewer studies as the number of insects increases, 66% of all articles focus on insects with a population of one to six.

To summarize, this section presented the used detection techniques, which were divided into sub-classes: existing detection techniques; existing classification techniques; and proposals of novel techniques.

3.4. RQ3: The Location of the Studied Insects

In this section, we will present the locations of the studied articles in Africa, Europe, Asia, America, and Australia.

- Asia: China (16), Vietnam, Japan, Thailand, Sri-Lanka, India, Taiwan, Singapore, Korea, Bangladesh,
- America: Brazil, Canada, USA, Peru,
- Europe: Denmark, Hungary, Romania, The Netherlands, Germany, Russia,
- Africa: Egypt (2),
- Australia: Australia (2).

Table 2. The distribution of the chosen articles in continents

Continent	**Number of Articles**	**(%)**
Asia	27	054
Africa	02	004
Europe	14	028
America	05	010
Australia	02	004
Total	50	100

As shown in Table 2, Asia has the majority of studies done in the field of insect detection, for 54% of all articles (27 of 50). We notice that China has 16 out of the total of 27 articles in Asia. In Africa, only Egypt has two studies, of which Mamdouh et al. [44] detect olive insects and Khalifa et al. [37] detect 8 types of insects.

Two of the studied papers conduct their research in other countries: [32], [35]. Nam et al. [35] mention that the research centre is in Japan, but the studies are in Vietnam. They mention that their studies are based on customer requirements. The customer is facing a problem with a specific type of insect and asks for help. Ard et al. [32] mention the capture and annotation being done by experts from greenhouse research centres based in Belgium and Spain, since they are part of Europe.

The publication date of the selected articles should also be mentioned. The selection criteria only produced 3 articles prior to 2018 [19] in 2016; and [9], [46] in 2017; and the remaining articles were published within the previous 4 years, as shown in figure 2. A total of 22 articles, or 44% of all articles, were published in 2021. Additionally, only nine articles were published in the first three months of 2022. If there is any significance to this, it might mean that there is a lot of interest in this area.

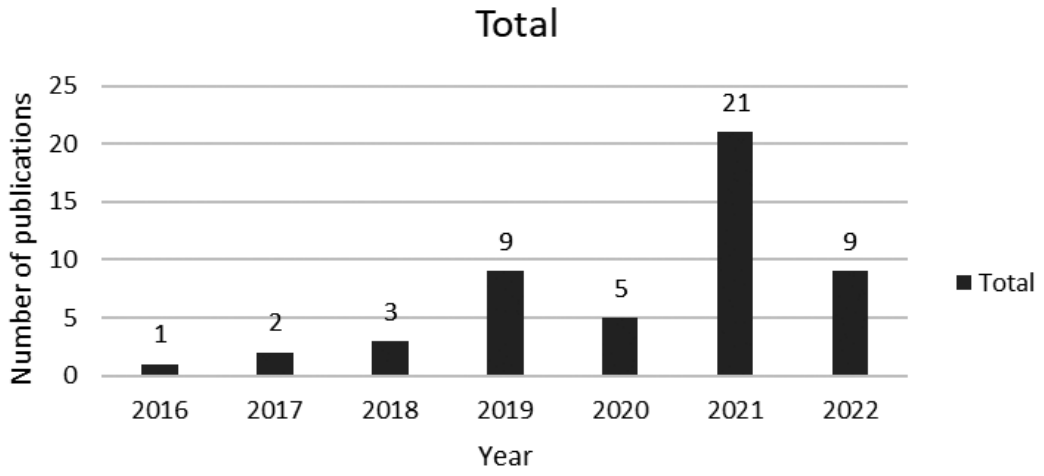


Figure 2. The distribution of the chosen articles

4. CONCLUSION

In our article, we provided a general overview of the field of insect detection by responding to three questions about it. The dataset that was used shows that on-site images that were taken in the field or a lab are more frequently used. We present the three types of detection techniques that are currently being used: existing detection or classification techniques; combinations of both; and new techniques. In addition, we discuss the use of transfer learning, which can shorten training periods. In response to the final query, which asks about the location of the detected insect, we note that China, which accounts for 56% of the articles chosen, is by far the leader in this area, while Africa still does not have a lot of concern in this field.

We now briefly outline our upcoming work. Initially, we will start with a small dataset that only shows one type of insect that is found in Tunisia before moving on to a larger data set that includes several types of them. Attacks on olive trees are commonplace for this insect. The next step is to develop a deep learning model capable of accurately identifying and counting this insect.

ACKNOWLEDGEMENTS

This work is supported by the German Academic Exchange Service (DAAD), Germany (Transformation Partnership: OLIVIA Project).

REFERENCES

[1] B. Ncube, W. Mupangwa, and A. French, (2018) “Precision agriculture and food security in africa,” in Systems analysis approach for complex global challenges. Springer, pp. 159–178.
[2] R. Gebbers and V. I. Adamchuk, (2010) “Precision agriculture and food security,” Science, vol. 327, no. 5967, pp. 828–831.
[3] R. N. Strange and P. R. Scott, (2005) “Plant disease: a threat to global food security,” Annual review of phytopathology, vol. 43, no. 1, pp. 83–116.
[4] D. C. K. Amarathunga, J. Grundy, H. Parry, and A. Dorin, (2021) “Methods of insect image capture and classification: A systematic literature review,” Smart Agricultural Technology, vol. 1, p. 100023.
[5] D. S. Hill, (1987) “Agricultural insect pests of the tropics and their control”. Cambridge University Press.

[6] D. S. Hill, (2012) "The economic importance of insects". Springer Science & Business Media.
[7] L. Butera, A. Ferrante, M. Jermini, M. Prevostini, and C. Alippi, (2021) "Precise agriculture: Effective deep learning strategies to detect pest insects," IEEE/CAA Journal of Automatica Sinica, vol. 9, no. 2, pp. 246–258.
[8] H. Kuzuhara, H. Takimoto, Y. Sato, and A. Kanagawa, (2020) "Insect pest detection and identification method based on deep learning for realizing a pest control system," in 2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE). IEEE, pp. 709–714.
[9] M. Menikdiwela, C. Nguyen, H. Li, and M. Shaw, (2017) "Cnn-based small object detection and visualization with feature activation mapping," in 2017 International Conference on Image and Vision Computing New Zealand (IVCNZ). IEEE, 2017, pp. 1–5.
[10] K. Li, J. Zhu, and N. Li, (2021) "Insect detection and counting based on yolov3 model," in 2021 IEEE 4th International Conference on Electronics Technology (ICET). IEEE, pp. 1229–1233.
[11] "Google images," https://images.google.com/, (Accessed on 08/05/2022).
[12] "Bing image inspiration feed," https://www.bing.com/images/feed?form=Z9L, (Accessed on 06/01/2022).
[13] "flicker photos on flickr — flickr," https://www.flickr.com/photos/tags/flicker/ (Accessed on 6/01/2022).
[14] H. Takimoto, Y. Sato, A. J. Nagano, K. K. Shimizu, and A. Kanagawa, (2021) "Using a two-stage convolutional neural network to rapidly identify tiny herbivorous beetles in the field," Ecological Informatics, vol. 66, p. 101466.
[15] M. I. Hossain, B. Paul, A. Sattar, and M. M. Islam, (2019) "A convolutional neural network approach to recognize the insect: A perspective in bangladesh," in 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART). IEEE, pp. 384–389.
[16] D. Abeywardhana, C. Dangalle, A. Nugaliyadde, and Y. Mallawarachchi, (2021) "Deep learning approach to classify tiger beetles of sri lanka," Ecological Informatics, vol. 62, p. 101286.
[17] T. D. C. Junior, R. Rieder, J. R. Di Domenico, and D. Lau, (2022) "Insectcv: A system for insect detection in the lab from trap images," Ecological Informatics, vol. 67, p. 101516.
[18] Y. Zhong, J. Gao, Q. Lei, and Y. Zhou, (2018) "A vision-based counting and recognition system for flying insects in intelligent agriculture," Sensors, vol. 18, no. 5, p. 1489.
[19] W. Ding and G. Taylor, (2016) "Automatic moth detection from trap images for pest management," Computers and Electronics in Agriculture, vol. 123, pp. 17–28, 2016.
[20] S.-J. Hong, S.-Y. Kim, E. Kim, C.-H. Lee, J.-S. Lee, D.-S. Lee, J. Bang, and G. Kim, (2020) "Moth detection from pheromone trap images using deep learning object detectors," Agriculture, vol. 10, no. 5, p. 170.
[21] L. Liu, R. Wang, C. Xie, P. Yang, F. Wang, S. Sudirman, and W. Liu, (2019) "Pestnet: An end-to-end deep learning approach for large-scale multiclass pest detection and classification," IEEE Access, vol. 7, pp. 45 301– 45 312.
[22] M. Rong, Z. Wang, B. Ban, and X. Guo, (2022) "Pest identification and counting of yellow plate in field based on improved mask r-cnn," Discrete Dynamics in Nature and Society, vol. 2022.
[23] K. Bjerge, J. B. Nielsen, M. V. Sepstrup, F. Helsing-Nielsen, and T. T. Hoye, (2021) "An automated light trap to monitor moths (lepidoptera) using computer vision-based tracking and deep learning," Sensors, vol. 21, no. 2, p. 343.
[24] P. P. Roosjen, B. Kellenberger, L. Kooistra, D. R. Green, and J. Fahrentrapp, (2020) "Deep learning for automated detection of drosophila suzukii: potential for uav-based monitoring," Pest Management science, vol. 76, no. 9, pp. 2994–3002.
[25] Z. Lyu, H. Jin, T. Zhen, F. Sun, and H. Xu, (2021) "Small object recognition algorithm of grain pests based on ssd feature fusion," IEEE Access, vol. 9, pp. 43 202–43 213.
[26] H. Yang, H. Zhao, D. Zhang, Y. Cao, S. W. Teng, S. Pang, X. Zhou, and Y. Li, (2022) "Auto-identification of two sitophilus sibling species on stored wheat using deep convolutional neural network," Pest Management Science, vol. 78, no. 5, pp. 1925–1937.
[27] M. A. Genaev, E. G. Komyshev, O. D. Shishkina, N. V. Adonyeva, E. K. Karpova, N. E. Gruntenko, L. P. Zakharenko, V. S. Koval, and D. A. Afonnikov, (2022) "Classification of fruit flies by gender in images using smartphones and the yolov4-tiny neural network," Mathematics, vol. 10, no. 3, p. 295.
[28] D. J. Patel and N. Bhatt, (2019) "Insect identification among deep learning's meta-architectures using tensorflow," Int. J. Eng. Adv. Technol, vol. 9, no. 1, pp. 1910–1914.

[29] P. Tresson, P. Tixier, W. Puech, and D. Carval, (2019) "Insect interaction analysis based on object detection and cnn," in 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP). IEEE, pp. 1–6.
[30] L. Du, Y. Sun, S. Chen, J. Feng, Y. Zhao, Z. Yan, X. Zhang, and Y. Bian, (2022) "A novel object detection model based on faster r-cnn for spodoptera frugiperda according to feeding trace of corn leaves," Agriculture, vol. 12, no. 2, p. 248.
[31] K. Bjerge, H. M. Mann, and T. T. Hoye, (2021) "Real-time insect tracking and monitoring with computer vision and deep learning," Remote Sensing in Ecology and Conservation.
[32] N. Ard and H. S. J. Hemming, (2018) "Detection and classification of insects on stick-traps in a tomato crop using faster r-cnn," in Proceedings of the Netherlands Conference on Computer Vision, pp. 1–4.
[33] D. J. A. Rustia, J.-J. Chao, L.-Y. Chiu, Y.-F. Wu, J.-Y. Chung, J.-C. Hsu, and T.-T. Lin, (2021) "Automatic greenhouse insect pest detection and recognition based on a cascaded deep learning classification method," Journal of Applied Entomology, vol. 145, no. 3, pp. 206–222.
[34] C. Chen, Y. Liang, X. Tang, M. Dai, and K. Zhou, (2022) "The research of pest detection in granary based on yolov4," Social Science Research Network.
[35] N. T. Nam and P. D. Hung, (2018) "Pest detection on traps using deep convolutional neural networks," in Proceedings of the 2018 International Conference on Control and Computer Vision, pp. 33–38.
[36] X. Wu, C. Zhan, Y.-K. Lai, M.-M. Cheng, and J. Yang, (2019) "Ip102: A large scale benchmark dataset for insect pest recognition," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 8787–8796.
[37] N. E. M. Khalifa, M. Loey, and M. H. N. Taha, (2020) "Insect pests recognition based on deep transfer learning models," J. Theor. Appl. Inf. Technol, vol. 98, no. 1, pp. 60–68.
[38] Z. Wang, L. Qiao, and M. Wang, (2022) "Agricultural pest detection algorithm based on improved faster rcnn," in International Conference on Computer Vision and Pattern Analysis (ICCPA 2021), vol. 12158. SPIE, pp. 104–109.
[39] M. E. Karar, F. Alsunaydi, S. Albusaymi, and S. Alotaibi, (2021) "A new mobile application of agricultural pests recognition using deep learning in cloud computing system," Alexandria Engineering Journal, vol. 60, no. 5, pp. 4423–4432.
[40] H. Liu and J. S. Chahl, (2021) "Proximal detecting invertebrate pests on crops using a deep residual convolutional neural network trained by virtual images," Artificial Intelligence in Agriculture, vol. 5, pp. 13–23.
[41] D. Korsch, P. Bodesheim, and J. Denzler, (2021) "Deep learning pipeline for automated visual moth monitoring: Insect localization and species classification," INFORMATIK 2021.
[42] C. Luo, P. Pearson, G. Xu, and S. M. Rich, (2022) "A computer vision-based approach for tick identification using deep learning models," Insects, vol. 13, no. 2, p. 116.
[43] Q. Liu, Z. Yan, F. Wang, and C. Ding, (2021) "Research on object detection algorithm for small object of pests based on yolov3," in 2021 International Conference on Computer Information Science and Artificial Intelligence (CISAI). IEEE, pp. 14–18.
[44] N. Mamdouh and A. Khattab, (2021) "Yolo-based deep learning framework for olive fruit fly detection and counting," IEEE Access, vol. 9, pp. 84 252–84 262.
[45] R. Kalamatianos, I. Karydis, D. Doukakis, and M. Avlonitis, (2018) "Dirt: The dacus image recognition toolkit," Journal of Imaging, vol. 4, no. 11, p. 129.
[46] M. Tanjim-Al-Akib and N. Islam, (2017) "Feature optimization of whitefly detection algorithm using image segmentation and feature analysis,".
[47] P. Rajan, B. Radhakrishnan, and L. P. Suresh, (2016) "Detection and classification of pests from crop images using support vector machine," in 2016international conference on emerging technological trends (ICETT).IEEE, pp. 1–6.
[48] M. Deserno and A. Briassouli, (2021) "Faster r-cnn and efficient net for accurate insect identification in a relabeled yellow sticky traps dataset," in 2021 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor). IEEE, pp. 209–214.
[49] "Raw data from yellow sticky traps with insects for training of deep learning convolutional neural network for object detection — research@wur," https://research.wur.nl/en/datasets/raw-data-from-yellowsticky-traps-with-insects-for-training-of-de, (Accessed on 08/05/2022).
[50] J. Suto, (2021) "Embedded system-based sticky paper trap with deep learning based insect-counting algorithm," Electronics, vol. 10, no. 15, p. 1754.

[51] J. Cabrera and E. Villanueva, (2022) “Investigating generative neural-network models for building pest insect detectors in sticky trap images for the Peruvian horticulture,” in Annual International Conference on Information Management and Big Data. Springer, pp. 356–369.
[52] A. Mazare, L. Ionescu, D. Visan, N. Belu, and A. Lita, (2019) “Pests detection system for agricultural crops using intelligent image analysis,” in 2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME). IEEE, pp. 228–231.
[53] D. Xia, P. Chen, B. Wang, J. Zhang, and C. Xie, (2018) “Insect detection and classification based on an improved convolutional neural network,” Sensors, vol. 18, no. 12, p. 4169.
[54] C. Xie, J. Zhang, R. Li, J. Li, P. Hong, J. Xia, and P. Chen, (2015) “Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning,” Computers and Electronics in Agriculture, vol. 119, pp. 123–132.
[55] R. Huang, T. Yao, C. Zhan, G. Zhang, and Y. Zheng, (2021) “A motor-driven and computer vision-based intelligent e-trap for monitoring citrus flies,” Agriculture, vol. 11, no. 5, p. 460.
[56] “A community for naturalists inaturalist,” https://www.inaturalist.org/, (Accessed on 06/01/2022).
[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, (2009) “Imagenet: A large-scale hierarchical image database,” in 2009 IEEE conference on computer vision and pattern recognition. Ieee, pp. 248–255.
[58] R. Wang, L. Liu, C. Xie, P. Yang, R. Li, and M. Zhou, (2021) “Agripest: A large-scale domain-specific benchmark dataset for practical agricultural pest detection in the wild,” Sensors, vol. 21, no. 5, p. 1601.
[59] J. Li, H. Zhou, D. S. Jayas, and Q. Jia, (2019) “Construction of a dataset of stored-grain insects images for intelligent monitoring,” Applied Engineering in Agriculture, vol. 35, no. 4, pp. 647–655.
[60] “puzzledqs/bbox-label-tool: A simple tool for labeling object bounding boxes in images,” https://github.com/puzzledqs/BBox-Label-Tool, (Accessed on 06/01/2022).
[61] “heartexlabs/labelimg: labelimg is a graphical image annotation tool and label object bounding boxes in images,” https://github.com/heartexlabs/labelImg, (Accessed on 06/01/2022).
[62] “wkentaro/labelme: Image polygonal annotation with python (polygon, rectangle, circle, line, point and image-level flag annotation).” https://github.com/wkentaro/labelme, (Accessed on 06/01/2022).
[63] K. O’Shea and R. Nash, (2015) “An introduction to convolutional neural networks,” arXiv preprint arXiv:1511.08458.
[64] R. Girshick, (2015) “Fast r-cnn,” in Proceedings of the IEEE International Conference on Computer Vision (ICCV).
[65] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, (2016) “Ssd: Single shot multibox detector,” in European conference on computer vision. Springer, pp. 21–37.
[66] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, (2016) “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[67] J. Redmon and A. Farhadi, (2018) “Yolov3: An incremental improvement,” arXiv preprint arXiv:1804.02767,.
[68] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, (2020) “Yolov4: Optimal speed and accuracy of object detection,” ArXiv, vol. abs/2004.10934.
[69] “ultralytics/yolov5: Yolov5 in pytorch onnx coreml . tflite,” https://github.com/ultralytics/yolov5, (Accessed on 06/01/2022).
[70] C. Cortes and V. Vapnik, (1995) “Support-vector networks,” Machine learning, vol. 20, no. 3, pp. 273–297.
[71] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu et al., (2008) “Top 10 algorithms in data mining,” Knowledge and information systems, vol. 14, no. 1, pp. 1–37.
[72] H.-x. Zhao and F. Magoules, (2012) “A review on the prediction of building energy consumption,” Renewable and Sustainable Energy Reviews, vol. 16, no. 6, pp. 3586–3592.
[73] A. Krizhevsky, I. Sutskever, and G. E. Hinton, (2012) “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, vol. 25.
[74] K. Simonyan and A. Zisserman, (2014) “Very deep convolutional networks forvlarge-scale image recognition,” arXiv preprint arXiv:1409.1556.
[75] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, (2015) “Going deeper with convolutions,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9.

[76] K. He, X. Zhang, S. Ren, and J. Sun, (2016) “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.
[77] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, (2017) “Aggregated residual transformations for deep neural networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[78] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, (2016) “Squeezenet: Alexnet-level accuracy with 50x fewer parameters and! 0.5 mb model size,” arXiv preprint arXiv:1602.07360.
[79] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, (2018) “Mobilenetv2: Inverted residuals and linear bottlenecks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520.
[80] Z. Yuan, W. Fang, Y. Zhao, and V. S. Sheng, (2021) “Research of insect recognition based on improved yolov5,” Journal of Artificial Intelligence, vol. 3,no. 4, p. 145.
[81] F. A. G. d. Silveira, E. C. Tetila, G. Astolfi, A. B. d. Costa, and W. P. Amorim, (2021) “Performance analysis of yolov3 for real-time detection of pests in soybeans,” in Brazilian Conference on Intelligent Systems. Springer, pp. 265–279.
[82] A. M. Hubalde, D. A. Padilla, and D. A. C. Santos, (2021) “A yolo-based approach for aedes aegypti larvae classification and detection,” in 2021 6th International Conference on Image, Vision and Computing (ICIVC). IEEE, pp. 161–167.
[83] F. Chollet, (2017) “Xception: Deep learning with depthwise separable convolutions,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251–1258.
[84] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, (2019) “Gcnet: Non-local networks meet squeeze-excitation networks and beyond,” in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops.
[85] B. Ramalingam, R. E. Mohan, S. Pookkuttath, B. F. G´omez, C. S. C. Sairam Borusu, T. Wee Teng, and Y. K. Tamilselvam, (2020) “Remote insects trap monitoring system using deep learning framework and iot,” Sensors, vol. 20, no. 18, p. 5280.
[86] Y. Chen, X. Zhang, W. Chen, Y. Li, and J. Wang, (2020) “Research onrecognition of fly species based on improved retinanet and cbam,” IEEE Access, vol. 8, pp. 102 907–102 919.
[87] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, (2017) “Inception-v4, inception-resnet and the impact of residual connections on learning,” in AAAI.
[88] K. He, G. Gkioxari, P. Dollar, and R. Girshick, (2017) “Mask r-cnn,” in 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988.
[89] R. Trufelea, M. Dimoiu, L. Ichim, and D. Popescu, (2018) “Detection of harmful insects for orchard using convolutional neural networks,” U.P.B. Sci. Bull., Series C.
[90] A. Porrello, S. Vincenzi, P. Buzzega, S. Calderara, A. Conte, C. Ippoliti, L. Candeloro, A. Di Lorenzo, and A. C. Dondona, (2019) “Spotting insects from satellites: modeling the presence of culicoides imicola through deep cnns,” in 2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS). IEEE, pp. 159– 166.
[91] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, (2017) “Focal Loss for Dense Object Detection,” arXiv e-prints, p. arXiv:1708.02002.

AUTHORS

Lotfi Souifi received the licence in computer science from the faculty of science of Monastir, university of Monastir, Tunisia, in 2016. He is currently a student in the second year of master in cyber physical systems in the high institute of computer science and multimedia of Sfax, university of Sfax, Tunisia. His current research areas include computer vision, deep learning, and object detection.

Afef Mdhaffar received the Engineering and M.Sc. degrees from the National Engineering School of Sfax, Tunisia, in 2007 and 2009, respectively, and the Ph.D. degree in computer science from the University of Marburg, Germany, in 2014. She is currently an Assistant Professor at the University of Sfax, Tunisia, and an Associate Researcher at the Digital Research Center of Sfax, Tunisia. She is involved in self-* architectures applied to cloud computing environments, data analytics methods, e-health, and precision agriculture.

Ismael Bouassida Rodriguez received the engineering and M.Sc. degrees in computer science from the National School of Computer Sciences of Manouba (Tunisia) and his Ph.D. from the University of Toulouse (France) and the University of Sfax (Tunisia) in 2011. He joined the National School of Engineers in Sfax as Assistant Professor of computer science in 2010 and he became an Associate Professor in 2012 at the Higher Institute of Computer Science and Multimedia of Sfax. His current research areas include software engineering of distributed systems, graphs grammars, self-adaptive and pervasive systems, and autonomic middleware.

Mohamed Jmaiel obtained his diploma of engineer in computer science from Kiel (Germany) University in 1992 and his Ph.D. from the Technical University of Berlin in 1996. He is, since 2009, full professor in computer science at the National School of Engineers of Sfax (Tunisia). His current research areas include software engineering of distributed systems, formal methods in model-driven architecture, self-adaptive and pervasive systems, autonomic middleware. Currently, he is the general director of the Digital Research Center at the Technopark of Sfax.

Bernd Freisleben received his M.Sc. in computer science from Pennsylvania State University, USA, in 1981, and his PhD and Habilitation in computer science from Technical University of Darmstadt, Germany, in 1985 and 1993, respectively. He is a full professor of computer science in the Department of Mathematics and Computer Science, University of Marburg, Marburg, Germany. His current research interests include data analytics, machine learning, and biomedical applications.

TOWARDS A SMART MULTI-MODAL IMAGE REGISTRATION PROCESS

Marwa Chaabane[1,2], Bruno Koller[2] and Ismael Bouassida Rodriguez[3]

[1]Department of Computer Science, University of Kiel, Germany
[2]Scanco Medical AG, 8306 Brüttisellen Switzerland
[3]ReDCAD Laboratory, ENIS, University of Sfax, Tunisia

ABSTRACT

The multi-modal image registration is a complex task in the medical domain. This task requires usually several manual interventions by the user/expert of the domain to adjust the image registration parameters properly to the characteristics of the processed image data. For this aim, the user needs to extract the relevant information from the image data and their meta-information. In this paper, we propose a novel architecture for a smart fully automatic multi-modal registration process. This architecture is based on a MAPE-K loop inspired by the architecture of autonomous systems.

KEYWORDS

Image registration, multi-modal image data, MAPE-K loop, medical domain.

1. INTRODUCTION

This work is performed within Marie Sklodowska-Curie ETN MgSafe[1]. The main objective of this project is to develop and enhance imaging technologies for recently established Mg implants by quantifying their physical impact and suitability for this class of materials in future human applications.

Highly sophisticated imaging modalities such as micro Computed Tomography (μCT), Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), Ultra Sound and Photo Acoustic (USPA) and histological sections, are developed in order to deliver data on different time and length scales of the body reaction and material behaviour during Mg degradation.

The obtained multi-modal imaging data was partially combined with molecular biological/biochemical analysis. All relevant biological and chemical in vivo and ex vivo data need to be merged. The combination of these results will allow for an upscaling of the processes to-wards humans and deliver valuable data in terms of patient safety.

The project is expected to generate a vast amount of 2D and 3D data from different modalities. The aim is to combine these data and to describe their relative relation in space and time with a multi-modal image registration process. The collected data to be combined, are characterized by their heterogeneity i.e., with different modalities, color systems, dimensions and resolutions and are provided by different machines in several institutes.

[1] https://www.mgsafe.eu/

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 57-67, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122004

Due to this heterogeneity, the multi-modal registration process usually needs many user's interventions to:

- Retrieve the path of the meta-files to extract relevant information.
- Collect the relevant information required to adjust the image registration parameters.
- Provide and adjust the parameters of the registration
- Pre-process the image data manually to minimize their heterogeneity e.g., applying a noise filter on an CT scan containing artifacts.

These manual interventions require an extra time and effort from the user and make the process of registration not easy to perform. In this work, we aim to develop a smart-automatic and user-friendly architecture for the management of heterogeneous image data. This architecture aims to reduce the user's intervention during the image registration process.

This paper will be organized as follows: In section 2, we present the background of our work. In section 3 is devoted we present related work dealing with automatic techniques for registration of multi-model medical images. Section 4 is devoted to detail our proposed architecture for image registration inspired by autonomous systems based on MAPE-K loop and its different steps. In section 5, at a first place, we illustrate our case study. At a second place, we apply the MAPE-K loop on this case study. In section 6, we focus on the implementation phase of the monitoring phase of the MAPE-K loop.

2. BACKGROUND

In this section, we describe the image data structure. Then, we explain the basic concepts of the image registration process in the medical domain.

2.1. Imaging Data Description

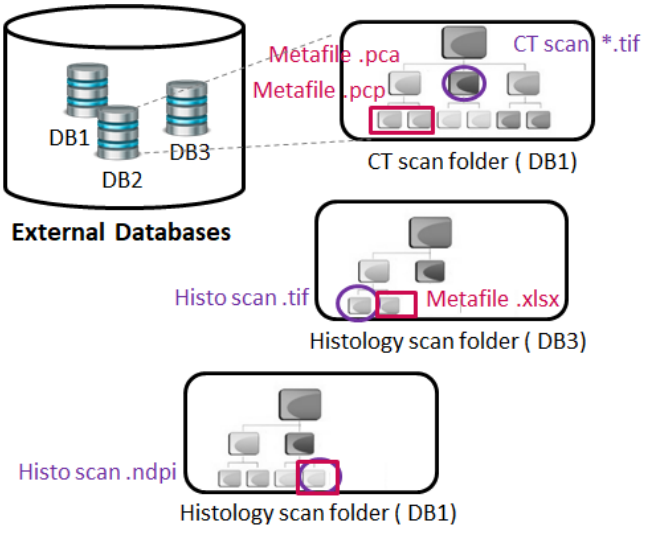


Figure 1. Data structure heterogeneity

A specimen can be scanned in several institutes by different scanners, and an institute scans multiple specimens. Which results in a set of scans of the same specimen connected to each other according to a timeline. A scan has a specific modality (CT, MRI, US, etc.) and may have one or more meta-files describing it. Since the imaging data are stored in different external databases

belonging to several institutes, the meta-data and data structure are different from one provider institute to another. Figure 1 illustrates the data structure heterogeneity through an example of three external image databases DB1, DB2 and DB3 provided by different institutes. First, the location of meta-files and image files are different in the three databases. Moreover, the relevant information of the CT scan in DB1 should be extracted from the meta-files with ".pcp" format and the relevant information of the histology section should be extracted from the header of the ".ndpi" file which contains both the meta-data and the image. While the relevant information of the CT scan in DB3 have to be extracted from the meta-file with .xlsx format. In addition, meta-files in DB1 and DB2 provide the relevant information following different text patterns.

2.2. Image Registration Process

In a generic way, a registration process is composed mainly of three steps. The first step is the segmentation of the colored images to binary images. It's an optional step that is highly required for the histological sections modality which refers to a slice of tissue which is viewed under a microscope to give cells information.

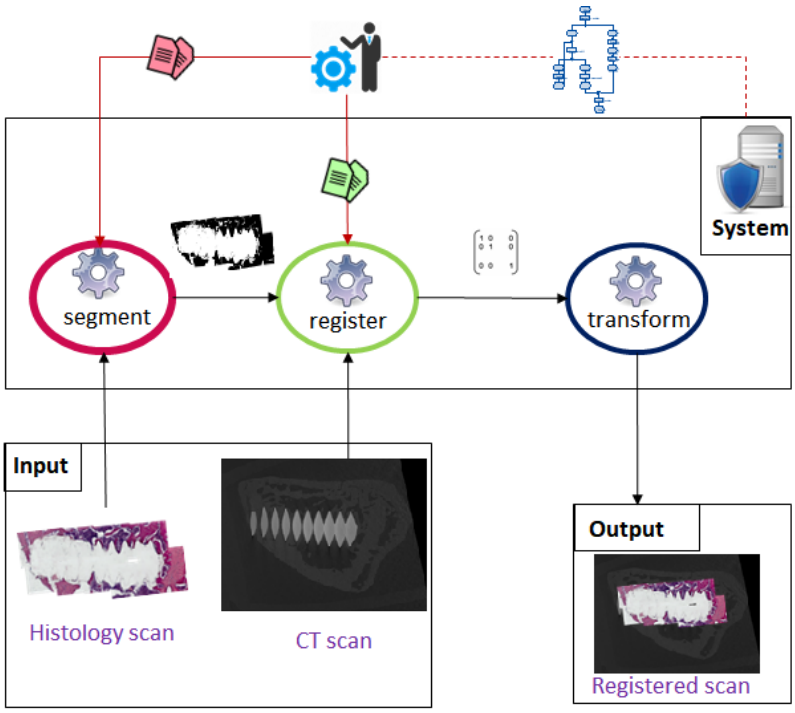


Figure 2. Image registration process

A histological section can be stained with different staining methods to highlight relevant features of the tissue e.g., "Hematoxylin and Eosin" where nuclei are stained blue, the cytoplasm are stained pink. The second step is the registration of one of the pair images inputs. And the final step is the transformation of the registered image. Figure 2 depicts the steps of an image registration process. In Figure 2, the system takes as input a 2D stained histological section and 2D section of a CT volume. First the system segments the colored histological image to a binary image. Then, it computes the transformation matrix of the binary image to match to the CT image, during the registration step. Finally, the system transforms the colored histological image according to the generated transformation matrix and gives as output a registered image.

3. RELATED WORK

In literature, there are several work dealing with 2D to 3D image registration -with the aim of combining histological sections to CT volumes-, and 3D to 3D image registration to combine CT to CT volumes over time, MRI to CT volumes and PET to CT volumes.

The work of Huanjie et al. [8] proposes an algorithm to perform the image registration of 3D bone images coming from different modalities (MR, SPECT and CT) to combine anatomical and functional information. This algorithm aims to improve the image registration accuracy based on feature-based image registration method.

Also, the research of Sabokrohiyeh et al. [6] is interested in image registration of 4D flow MRI (poor quality) to 3D cine MRI; multi 2D slices (high quality) following a sketch-based image registration method to improve registration accuracy.

With the same objective of achieving a high accuracy, the work of Liu at al. [4] suggests a weighed image registration method based on curvature feature.

In addition, the research of Albers et al. [2] is a recent work dealing with registration of 2D histological data to 3D CT volume based on an elastic image registration. This type of registration requires to be performed following 2 steps: First, retrieve the orientation of the 2D histology section to extract the appropriate section within the CT volume (manually). Then perform an automatic 2D to 2D registration. Even though the previously mentioned work are interested in performing 2D/3D to 3D image registration semi-automatically or automatically, they still limited to perform the registration image by image.

On the other hand, medical analysis relies on a big amount of imaging data. In this context, Xamflow[2] is a workflow-based framework proposed by Lucid. It enables to execute a workflow e.g., registration, on a vast imaging data set. This framework allows to execute static workflows which fits to process imaging data which have the same characteristics, e.g., image registration of a set CT volumes scanned overtime with the same machine.

Thus, the existing solutions are limited to process data that have the same characteristics. The current solutions are time consuming and require the user's intervention along the steps of the registration process to adjust the registration parameters properly to the given data set characteristics.

4. SMART ARCHITECTURE FOR IMAGE REGISTRATION INSPIRED FROM AUTONOMOUS SYSTEMS' ARCHITECTURE

The autonomic computing was proposed as a solution to tackle many complex tasks in several fields as industry 4.0, e-health, smart grid, and agriculture. It brings a system with self-management capabilities to dynamically adapt itself to its context.

Inspired by the autonomous systems' applications we propose to apply a MAPE-K loop [7] [1] [5] on the multi-modal image registration process. A MAPE-K loop is a conceptual architecture for autonomic applications has been suggested by IBM [3]. The aim of this architecture is to adapt autonomously the behaviour of a system according to the changes of their environments.

[2] https://xamflow.lucid.ch/

The principal element of this architecture is the autonomic manager which interacts with managed entities within sensors and actuators. The autonomic manager is responsible for the phases of Monitoring, Analysis, Planning and Execution.

During the execution of the monitoring phase, the autonomic manager collects information about the context via the sensors. In the Analysis phase, it analyzes whether an adaptation of the behaviour is needed. The Planning phase consists in the selection of the most suitable reconfiguration actions which is called a plan of decision. Finally, in the Execution phase the appropriate plan of decision is selected and executed.

On the other hand, a registration process takes place as depicted in the workflow in Figure 3: The user selects two images as input e.g., a CT scan and a histology section and extracts manually the relevant information (e.g., dimensions of the image, pixel size, color system) from the meta-data of the image. Then, the user decides the different steps to be execute by the system and provides the collected relevant information in parameter: First, the system converts the data to a readable format. Second, it segments the histology-colored section and the CT images to binary images. Third, it computes the transformation matrix. And finally, it transforms the histology section and gives as output a registered image.

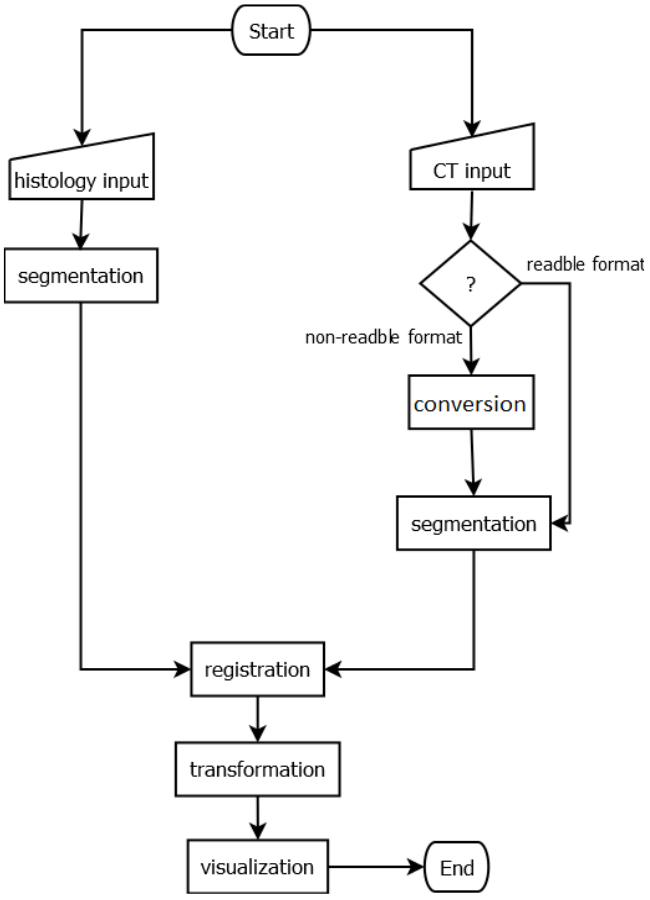


Figure 3. An example of registration workflow: histology to CT

This workflow can be automatically re-used only on another set of data containing the same characteristics, in a static way. However, due the heterogeneous nature and huge number of combinations of image data (e.g., registration CT to CT, registration CT to PET, registration MRI to USPA, etc.), more adaptability is needed to process the data automatically.

To apply the MAPE-K loop on the registration process, we define:

- The registration workflow as the behavior of our system.
- The image data in input as a changing context of the system, i.e., the system is required to adapt its behavior running an appropriate image registration workflow, according to the characteristics and modalities of the images in input.
- The metadata of the images as the sensors' module providing information about the context changes.

5. CASE STUDY: MAPE-K LOOP APPLIED ON THE MULTI-MODAL REGISTRATION

In this section, we detail our case study. Figure 4 explains how the different phases of the MAPE-K loop are applied on our case study. The system is connected to the data layer composed of three databases: "Ref Database", "Knowledge base" and "Reg information database"

5.1. Ref Database

The "Ref database" is connected to the external database of the provider institute. It includes the references to the specimen folders located in the external database. It also includes information about the provider institute, modality, reference of the specimen (in order to link scans belonging to the same specimen).

5.2. Knowledge Base

The Knowledge base includes rules that describe the appropriate options that the system can run according to a specific context.

5.3. Reg Information Database

The "Reg information database" is connected to the Knowledge base. It feeds the Knowledge base with the behaviour history of the system which is the workflows executed previously. This database includes the executed workflows, the id of the rules, the selected plans of decision and the efficiency of the behaviour represented with an error function.

5.4. Monitoring, Analysis, Planning and Execution

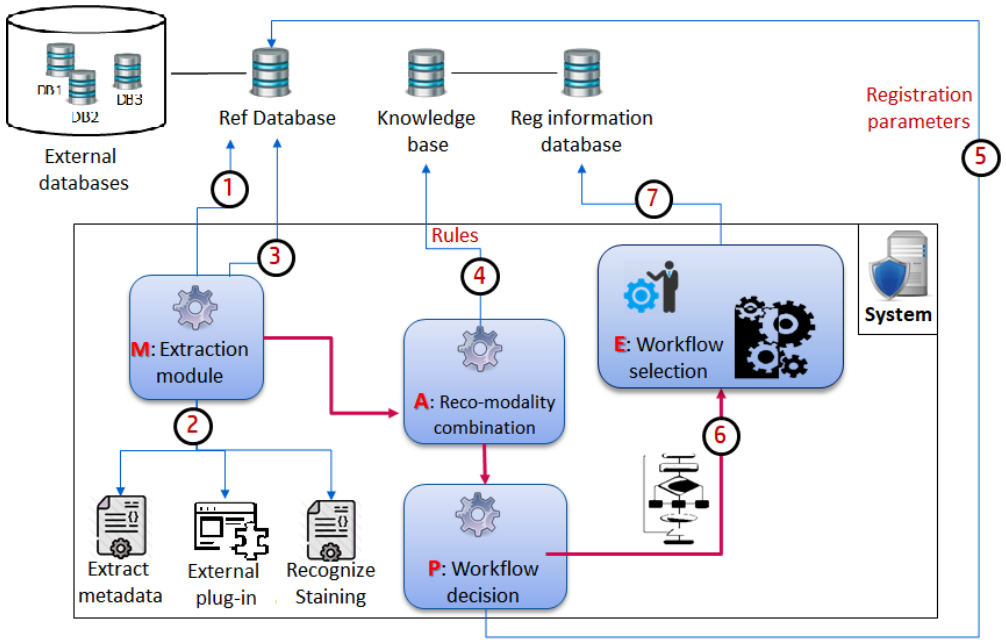


Figure 4. MAPE-K applied on image registration process

First, during the monitoring phase, the “extraction module” collects the relevant information belonging to the image data given as input. The relevant information is collected from the meta-files of the images, the header of the image files or the image files (2). The extraction module access to the external data via “Ref database” (1). After operating, it feeds “Ref database” with collected information (3). Then, in the analysis phase the “Reco modality combination” analyzes the observations provided by the previous phase, either based on the set of predefined rules or based on the historical of the behaviour of the system stored in the knowledge base (4). According to the provided analysis, in the planning phase the module “Workflow decision” decides a plan of decision suggesting a set of appropriate registration workflows to execute. The module “Workflow decision” collects the needed parameters values adequate to each suggested registration workflow from the relevant information stored in “Ref database” (5). Also, provides the description of the suggested registration workflows to “Workflow decision” module (6). Finally, in the execution phase, the “Workflow selection” module selects and executes most suitable registration workflow based on the provided plan of decision and the previous system behaviours stored in “Reg information database” (7). Then, it feeds “Reg information database” with the new system behaviour. As the Monitoring phase is critical to deal with a huge amount of data characterized by their heterogeneity, in this Paper, we highlight the Monitoring phase.

6. Monitoring Phase: The Extraction Module

In the following section, we detail the implementation of the extraction module representing the monitoring phase. The required relevant information is collected from the meta-files (e.g. the pixel size and the dimensions of a CT 3D volume), from the header of the image (e.g. the dimensions of an histology image “.dicom” format), or from the image file (e.g. the staining method of an histology section).

6.1. Extraction from Meta-File

The meta-files have different location structures in the different external databases. Further, the meta-files have different text patterns depending on the type of the imaging machine. A metadata extraction module is responsible on defining the placement the appropriate meta-file containing the required relevant information.

This module, also, allows to recognize the text pattern of the meta-file. According to these two features it runs a text extraction code to extract the relevant information and store it.

6.2. Extraction from Header

Images with some format such as ".dicom" or ".ndpi" includes information in the header of the image file. Our extraction module supports ".ndpi" format. It connects to an external plugin which enables decompressing and ".ndpi" file and reading its header.

6.3. Extraction form Image File: Staining Recognition

The staining method of a histology section is an essential information required to register a histology section to any 3D grayscale volume. This information is required for the segmentation to a binary image to decide the adequate segmentation method to process. This information should be implicitly deducted from the image file.

For this aim, based on a machine learning code for colors identification, we made improvements to develop a new code for staining recognition of a histology section. Our improved code supports the four staining methods shown in Figure 5: HE, Levai Laczko, Masson's trichrome and Van Kossa.

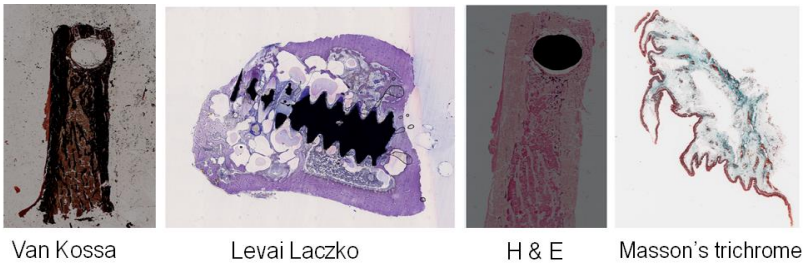


Figure 5. Examples[3] of different stained bone tissue section including an implant

Our code takes as input a histology image and gives as output the name of the used staining method. In the initial code of colors identification[4], first, the image is read. Then, the code identifies the n top existing colors based on the k-means algorithm to get colors cluster and prepares a class colors table including a set of color ranges. Then, Match the image to the appropriate class. Listing 1 shows the first and second steps of the code (respectively in line 7 and line 8).

[3] histological sections provided by: Institute of Metallic Biomaterials, Helmholtz-Zentrum hereon GmbH, Geesthacht-Germany; Institute of Clinical Dentistry, University of Oslo, Norway; the Virtual Microscopy Database (VDM)

[4] https://github.com/kb22/Color-Identification-using-Machine-Learning

```
 1 from sklearn.cluster import KMeans
 2 import matplotlib.pyplot as plt
 3 import numpy as np
 4 import cv2
 5 from skimage.color import rgb2lab, deltaE_cie76
 6
 7 def read_image (image_path):
 8 def identify_colors (image, number_of_colors, show_chart):
 9
10 IMAGE_DIRECTORY ="..\\..\\..\\ histology_images"
11 print (IMAGE_DIRECTORY)
12 COLORS = {
13 'trichome_green_1': [50, 121, 115],
14 'trichome_green_2': [50, 200, 190],
15 'trichrome_red': [200, 30, 50],
16 'HE_pink': [240, 190 ,235],
17 'HE_purple': [200, 30, 50],
18 }
19 for file in os.listdir (IMAGE_DIRECTORY):
20 if not file.startswith ('.'):
21 images.append(get_image(os.path.join(IMAGE_DIRECTORY, file)))
22 names.append(file)
```

Listing 1. Colors threshold initialization

While, Listing 2 shows the third step our code after improvement. The histology images are stained with two or more colors. Thus, we need to match the image to a composed staining cluster instead if color cluster. For instance, we developed the method "match image by color or and" to match an image to trichrome masson goldner's staining based on a predefined rule (from line 2 to line 16). In this example the rule is: if the identified top n colors of the image contain green1 or green2 and red then the image is matched to trichrome masson goldner's cluster (line 18).

```
 1
 2 def match_image_by_color_or_and (image, color1, color2, color3,
threshold1, threshold2, threshold3, number_of_colors ):
 3 for i in range (number_of_colors ):
 4 curr_color = rgb2lab (np.uint8 (np.asarray([[image_colors[i]]])))
 5 diff1 = deltaE_cie76 (selected_color1, curr_color)
 6 diff2 = deltaE_cie76 (selected_color2, curr_color)
 7 diff3 = deltaE_cie76 (selected_color3, curr_color)
 8 if (diff1 < threshold1):
 9 select_image1 = True
10 if (diff2 < threshold2):
11 select_image2 = True
12 if (diff3 < threshold3):
13 select_image3 = True
14 if ((select_image1 and select_image3) or (select_image2 and
select_image3)):
15 select_image = True
16 return select_image
17 plt.figure(figsize = (20 , 40))
18 show_selected_images_or_and (names,images, COLORS['trichome_green1'],
COLORS['trichome_green2'], COLORS['trichome_red'], 30 ,35 ,50 ,10)
```

Listing 2. Matching image to staining method

7. CONCLUSION

In face of the increasing amount of imaging data in the medical domain and the need of combining these data for better medical analysis, several work aim to improve image registration process in term of the registration accuracy. Other work aim to perform this process as automatically as possible on a large set of imaging data. Nevertheless, the highly heterogeneous characteristics of these data limits the proposed solutions.

Inspired by the autonomous systems, we proposed a MAPE-K loop applied on the image registration process to adapt automatically the functioning of the solution according to the changing characteristics of our data. In this paper, we detailed the monitoring phase of the MAPE-K loop as it has the most important impact on reducing the manual user intervention in the imaging registration process.

ACKNOWLEDGEMENT

We would like to thank the Virtual Microscopy Database (VDM)[5], Helmholtz- Zentrum hereon GmbH, Geesthacht-Germany (HZG)[6] and Institute of Clinical Dentistry, University of Oslo, Norway[7] for providing data sets of different stained histology sections.

REFERENCES

[1] I. Abdennadher, (2022), "DAACS: a decision approach for autonomic computing systems", *J. Supercomput.*, Vol. 78, No. 3, pp3883-3904.
[2] J. Albers, A. Svetlove, J. Alves, A. Kraupner, F. di Lillo, M. A. Markus, G. Tromba, F. Alves, and C. Dullin, (2021), "Elastic transformation of histological slices allows precise coregistration with microct data sets for a refined virtual histology approach", *Scientific Reports*, Vol. 11.
[3] IBM, (2006), *An architectural blueprint for autonomic computing*.
[4] B. Liu, X. Gao, H. Liu, X. Wang, and B. Liang, (2018), *A fast weighted registration method of 3d point cloud based on curvature feature*. ICMIP, pp83-87, Association for Computing Machinery.
[5] Rutten, N. Marchand, and D. Simon, (2017). *Feedback Control as MAPE-K Loop in Autonomic Computing*, pp349-373.
[6] S. Sabokrohiyeh, K. Ang, M. Elbaz, and F. Samavati, (2019), *Sketch-based registration of 3d cine mri to 4d flow mri*. ICBBT, pp14-21, Association for Computing Machinery.
[7] G. Sancho, (2010). *Adaptation d'architectures logicielles collaboratives dans les environnements ubiquitaires. Contribution à l'interopérabilité par la sémantique*. PhD thesis, Toulouse 1 University Capitole, France.

[5] https://www.virtualmicroscopydatabase.org/
[6] https://www.hereon.de/
[7] https://www.odont.uio.no/

AUTHORS

Marwa Chaabane received the master's degree in computer science: Industry 4.0 from the University of Pau and the Adour Region (France) in 2019. She also received the Bachelor and master's degree in computer science and Multimedia from the university of Sfax (Tunisia), respectively in 2015 and 2017. She is currently a computer scientist at Scanco Medical AG and PhD student at the University of Kiel- Germany, within Marie Skłodowska-Curie ETN MgSafe project.

Bruno Koller received his Ph.D. in Electrical and Electronics Engineering from ETH Zürich (Switzerland) in 1993. He is currently CEO of Scanco Medical AG, a SME in Brüttisellen near Zurich has been a pioneer in the field of high-resolution micro-computed tomography (µCT) for more than three decades

Ismael Bouassida Rodriguez received the engineering and M.Sc. degrees in computer science from the National School of Computer Sciences of Manouba (Tunisia) and his Ph.D. from the University of Toulouse (France) and the University of Sfax (Tunisia) in 2011. He joined the National School of Engineers in Sfax as Assistant Professor of computer science in 2010 and he became an Associate Professor in 2012 at the Higher Institute of Computer Science and Multimedia of Sfax. His current research areas include software engineering of distributed systems, graphs grammars, self-adaptive and pervasive systems, and autonomic middleware.

IMPROVING ROBUSTNESS OF AGE AND GENDER PREDICTION BASED ON CUSTOM SPEECH DATA

Veera Vignesh Kandasamy and Anup Bera

Accenture Solutions India Pvt Ltd, India

ABSTRACT

With the increased use of human-machine interaction via voice enabled smart devices over the years, there are growing demands for better accuracy of the speech analytics systems. Several studies show that speech analytics system exhibits bias towards speaker demographics, such age, gender, race, accent etc. To avoid such a bias, speaker demographic information can be used to prepare training dataset for the speech analytics model. Also, speaker demographic information can be used for targeted advertisement, recommendation, and forensic science. In this research we will demonstrate some algorithms for age and gender prediction from speech data with our custom dataset that covers speakers from around the world with varying accents. In order to extract speaker age and gender from speech data, we've also included a method for determining the appropriate length of audio file to be ingested into the system, which will reduce computational time. This study also identifies the most effective padding and cropping mechanism for obtaining the best results from the input audio file. We investigated the impact of various parameters on the performance and end-to-end implementation of a real-time speaker age and gender information extraction system. Our best model has a RMSE value of 4.1 for age prediction and 99.5% for gender prediction on custom test dataset.

KEYWORDS

Age and Gender prediction, Data Bias, Speech Analytics, CNN, LSTM, Wav2Vec.

1. INTRODUCTION

Speech analytics is the most predominant subject that has been used in various industries as a tool to understand consumers, predict human behaviour and create content. The speech analytics has various use cases like contact centre operations (Litvinov et.al)[1] customer service improvement (Scheidt et.al) [2], new language learning and mental health screening (Yiling Li) [3]. In all of these use cases, speech to text, speech sentiment, text to speech technologies are considered primarily. Speech analytics is the practice of collecting and analysing voice data with the use of speech technologies to improve the user experience. Speech analytics comprises of various functions, but it is subjected to bias as all other machine learning algorithms. These biases might occur at any stage in the speech analysis process. In case of call centre applications, the speech to text models used, might be trained on some data irrespective of speaker demographics taken into consideration. When speech to text is performed on audio calls for a specific age group or gender for which training data is not readily available, this can lead to bias issues. In this study, we are attempting to understand speaker characteristics in terms of age and gender for a certain set of audio samples that represent the overall population. If the initial speech-to-text model was trained using data with skewed sampling for age group and gender, it will be difficult for the system to

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 69-83, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122005

maintain accuracy across all age groups and gender. To avoid bias of a model, it is important to prepare a training dataset which has equal representation of all age groups and gender. To achieve this, there should be some frameworks for extraction of the age and gender information from a speech data. In this study we will explore some of the deep learning approaches to predict the speakers age, age group and gender.

1.1. Unique Contribution of this Work are:

•In this study, we have combined multiple datasets like TIMIT, VCTK, NISP, and GMU that provide more generalization in terms of age distribution and gender of the speaker. This helps in making the solution more robust and generalised during the usage in real world problems

•As part of the research, the appropriate duration of the audio recording required to accurately assess the age and gender is determined. On the combined custom dataset, the 5s appears to provide better generalisation values based on performance and computational cost.

•In this study, the impact of padding and cropping input audio files on accuracy metrics is also investigated.

•Accurately predict age [discrete and continuous] and gender from the audio samples

- Weighing mechanisms used to better understand classes with lower representation (Classification Problem)

Our paper is organized as follows: Section 2 reviews the available literature on the selected topic and gives some theoretical background, section 3 contains the Research methodology as well as research questions section 4 contains the results based on our research, and section 5 concludes our research paper with future work.

2. THEORETICAL BACKGROUND AND LITERATURE REVIEW

This section briefly discusses some issues, and showcases literature related to devising the age and gender detection problem.

2.1. Use of Machine Learning Models to Detect Age and Gender

In Safavi et.al. [4] propose a hybrid model that makes predictions and classifies age, gender, and accent using different models. XGBoost, a decision tree based on ensemble technique, is used to classify gender, KNN for Age, and Random Forest for detecting the accent of a speaker. In Jasuja et.al. [5], implemented a gender specific user classification using a deep learning model based on the Multilayer Perceptron (MLP). The proposed model was trained with various parameters and produced an MLP model with 96% accuracy in the test dataset. These predictions were showcased on a very clean dataset which would not be in case of real-world situations like a call centre.

2.2. Use of Deep Learning Models to Detect Age and Gender

Buyukyilmaz [6] proposed a multi-layer perceptron deep learning model helped detect gender based on the auditory characteristics of sound and voice. The classification model was 96.74 percent accurate. Maka and Dziurzanski [7] used a dataset containing 438 men and 192 women (indoor and outdoor listening scenes) and used 630 speakers in experiments on gender

identification issues in different language contexts. They found that nonlinear smoothing improved classification accuracy by 2% to become 99.4% in total. Tursunov [8] propose a novel CNN model with Multi-Attention Module (MAM) for age and gender classification using speech spectrograms from the input speech signals. Sánchez-Hevia [9] identifies the available types of DNN techniques to simultaneously detect age and gender in the context of Interactive Voice Response (IVR) systems. The results reveal that the larger the network, the better the results, but the improvement is insufficient to justify the additional processing cost.

3. RESEARCH METHODOLOGY

3.1. Dataset Description

In order to make the Age and Gender prediction model robust we have collected various available datasets. The following is a brief description about each of them

NISP Dataset [10]**:** The NISP dataset consists of audio samples from 345 speakers (60% male and 40% female) each contributing about 4-5 minutes of data. For our use case we are considering only English sentences spoken by the speakers. The dataset contains the exact age of the speaker, and the distribution ranges from teens to forties.

TIMIT Dataset [11]**:** The dataset includes recordings of over 600 American speakers from eight different dialects with 70% male and 30% female. The dataset includes the speaker's exact age, which ranges between the twenties and the seventies.

VCTK Corpus [12]**:** The VCTK dataset is made up of audio recordings of 110 different speakers reading 400 different phrases from newspapers. It contains the exact age of the speaker, ranging from teens to thirties, with male speakers with 57% of the data and female speakers of 43%. The audio was recorded in a professional noise free setting with two distinct microphones at 24 bits and then down sampled to 48KHz stored as FLAC. Only the audio from microphone 2 is being considered.

GMU Corpus [13]**:** The dataset contains speakers from various part of the world with different accent read the same English sentence. The latest version of the dataset contains around 2982 samples of such recordings from which we have removed synthesized audio.

Combined Dataset: In the individual datasets listed above, there is a significant imbalance in the data's male-female distribution, and each data set contains speakers only from a single region. To mitigate this, we must first create a bias-free dataset that covers the entire spectrum of speakers.

Table 1 lists the number of records from each of the dataset across different age groups. Since the datasets are in different formats, we built a pipeline to standardise the audio input that must be passed into our model. The choice of the parameters chosen to be standardized are discussed below.

Table 1. Dataset details across age groups

Dataset	**Infants**	**Teens**	**Twenties**	**Thirties**	**Forties**	**Fifties**	**Sixties**	**Seventies**	**Eighties**	**Nineties**	***Total***
GMU	2	320	1344	542	325	260	103	43	20	3	***2962***

NISP	0	688	11226	2222	556	0	0	0	0	0	***14692***
TIMIT	0	0	3630	1860	570	200	30	10	0	0	***6300***
VCTK	0	5456	37183	1234	0	0	0	0	0	0	***43873***
TOTAL	***2***	***6464***	***53383***	***5858***	***1451***	***460***	***133***	***53***	***20***	***3***	***67827***

3.2. Research Questions

Majority of the work in this study focuses on creating a balanced dataset towards sensitive variables, such as age and gender and to optimize the computational aspect with better prediction accuracy.

Following are the key questions answered in this research:

•How can we create a dataset that represents the entire range of speakers without bias?
•What is the effectiveness of the proposed data ingestion module over state of art age and gender detection models?
•What is the required audio length for understanding characteristics of age and gender accurately?
•How can we accurately predict age and gender using a single model?
•What is the impact of padding and cropping on the training and validation metrics?

3.3. Data Processing Framework

3.3.1. Data Standardization

We know from the last section that the data comes from different sources with different formats and sample rates different naming for the speaker's profile information. A Pipeline was created to convert all the datasets to wav format with 16KHz sampling rate in order to provide a balance between the training time and the accuracy of the model.

Table 2 show cases the list of datasets with their existing file type and other metadata.

All the datasets are converted into the standard format mentioned below.

•File Type: WAV
•Sampling Rate: 16000 KHZ
•Number of Channels: 1
•Bits Per sample: 16
•Encoding: PCM S

Table 2. Dataset Metadata description

Dataset Name	File Type	Sample Rate	Channels	Bits Per Sample	Encoding
TIMIT	WAV	16000	1	16	PCM_S
NISP	WAV	48000	1	16	PCM_S
GMU	WAV	44100	1	16	PCM_S
VCTK	FLAC	48000	1	16	PCM_S

3.3.2. Description of Dataset

From Table 1 we understand that the number of records in the infants and nineties are extremely low. For our use case, we are concentrating on speakers whose age range is in teens to fifties. We

also down sampled the number of records in the teens and twenties age range to avoid bias in the model towards these majority classes.

Table 3 showcases the final Combined Dataset prepared that includes the train and validation set from the various standard datasets. Both age group (Teens – Fifties) distribution and gender (Male & Female) distribution are shown in the same. While preparing the dataset, we made sure to include the best possible distribution that represents both men and women which is shown in Figure 1

Table 3. Age and Gender Dataset

Dataset	Teens	Twenties	Thirties	Forties	Fifties	Male	Female
GMU	274	169	469	262	224	685	713
NISP	582	1314	1889	487	0	2545	1727
TIMIT	0	301	1165	402	92	1494	466
VCTK	4683	4094	1050	0	0	3845	5982
Total	**5539**	**5878**	**4573**	**1151**	**316**	**8569**	**8888**

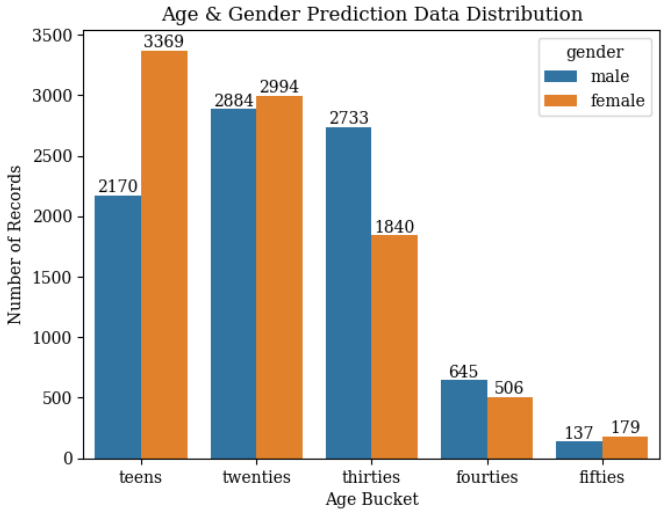


Figure 1. Age & Gender Dataset Distribution

3.3.3. Audio Duration processing

Audio can be thought of as a single-dimensional array of numbers. In order to process the input sequences, our model requires them to be the same length. This can be accomplished through the use of pre-processing techniques. A cut off duration can be defined based on which either padding or cropping can be done to the input audio. When our audio is shorter than the cut-off, we must pad the sequence data to the left or right to make it in desired length. Similarly, if the length of the audio exceeds the cut off, we crop it to the desired length on the right, left, or at random. The current literature places little emphasis on the ideal length of audio required for accurate age and gender prediction, owing to the dataset's limited distribution of audio length. In our study, we experimented with various audio lengths that would be required to optimise prediction accuracy as we have included data from multiple datasets. We also investigate the

impact of various padding and cropping strategies on the input audio. Further experiments were conducted following the selection of the ideal audio duration required.

3.4. Experimental Design

The following experiments were developed using the combined dataset created in order to find answers to the research's key questions.

1. Finding the ideal duration that would predict the age and gender of the speaker. Using the combined dataset we chose a model and applied by modifying the duration of the audio.
2. Finding the effective padding and cropping strategy for the selected audio duration. Padding and Cropping can be done either on the left or right or at random position of the audio.
3. Find the optimal the number of MFCC feature.
4. CNN model with MFCC & Wav2vec as the input features
5. LSTM model with MFCC & Wav2vec as the input features
6. CNN+LSTM model with MFCC & Wav2vec as the input features
7. Multi CNN+LSTM model with MFCC & Wav2vec as the input features.

Every experiment builds on the one before it, answering important questions at each stage and using the optimal parameters afterward. For instance: The ideal audio length for predicting age and gender is determined in experiment 1 and used to determine the padding and cropping method in experiment 2.

3.5. Feature Extraction

Feature extraction techniques are used to extract pertinent information from an audio source. Existing literature uses MFCC [9], Mel Spectrogram [14], Mel-filter banks [15] as input features, along with Pitch, Chroma, and Tonnetz as additional features for performing age and gender classification. It is established that MFCC provides resiliency to Noise in the dataset.

In the case of pretrained models like wav2vec [16] which is trained in an unsupervised manner on a different set of audio dataset can be fine-tuned based on the problem using transfer learning techniques. In these pretrained models feature extractions are accomplished internally; nonetheless, a certain format and sample rate are required to use these models. Internally wav2vec uses CNN encoder over each segment of the raw audio and learns by predicting the masked speech unit.

As part of our research, we have experimented with MFCC and Wav2Vec as the feature extraction technique.

3.6. Model Description

Our model takes in audio from the custom dataset as batches. The general flow as shown in Figure 2. consists of 3 steps. The feature extraction layer takes in two values either wav2vec or MFCC. In case of MFCC the best number is chosen from the result of previous experiment. The model layer consists of CNN, LSTM, CNN+LSTM, Multi CNN + LSTM. The experiments were carried out with various combination of feature and model type. The model is then able to predict three different values age, age group and gender of the speaker.

We have developed these models in python 3.9 [17] using Pytorch Lightning [18] which is a wrapper around Pytorch [19].

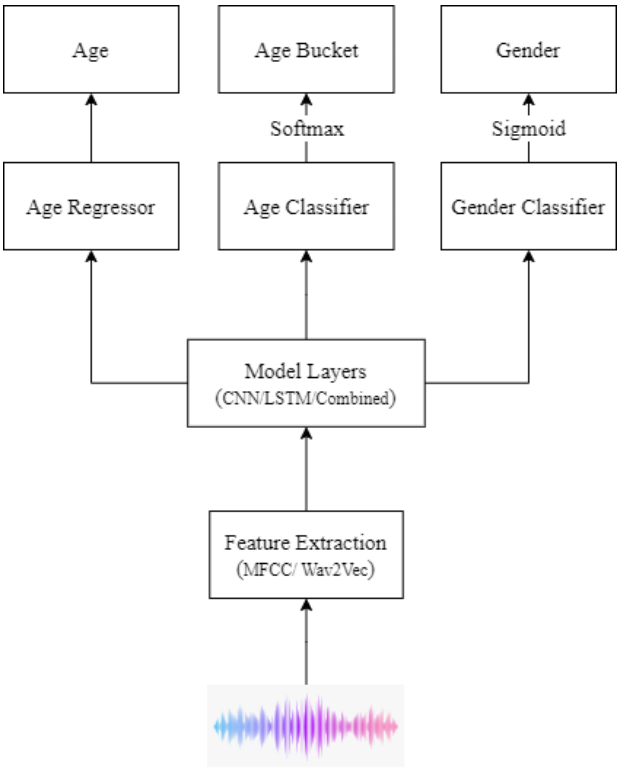


Figure 2. General Model Description

3.6.1. CNN Model

We have two kinds of CNN model to accommodate for variation in feature extractors. For Wav2Vec feature we pass in the raw audio and for MFCC based model we extract the MFCC and then pass it to the model. The Model layer consists of Conv1d Layer with ReLU and Batch Normalization. Soft Attention is applied at the end to reduce the dimension.

Output of the attention layer is then passed to the multi head setup where it predicts age, age group and gender of the speaker from the input. As age is regression problem, we use a single neuron as the last layer. The neurons in the final layer for age group correspond to the five age groups we have in our situation. We have taken a softmax of these values so as to get the probability of each class. While predicting gender as it's a binary classification problem we have one neuron as the last layer with sigmoid applied.

3.6.2. LSTM Model

LSTM models are known to work good for sequence prediction problems. Similar to CNN Model we have two variants of the model based on the feature extractor. Model layer consists of LSTM with soft attention. Output of the attention layer is then passed to the multihead classifier and regressor.

3.6.3. CNN+LSTM Model

We combined the aforementioned two models to increase complexity and, as a result, performance while enabling our model to pick up additional features. Model Layer consists of

Conv1d layer followed by LSTM and Soft Attention. Output of the soft attention is then used to predict the age, age group and gender.

3.6.4. Multi CNN+LSTM Model

In this Model we have experimented with varying Convolution stack(n*Conv1d+ReLU+BatchNorm) and LSTM with self-attention layers. The model is made more complex so that it can pick up new features and improve its performance in classification and regression tasks.

3.7. Evaluation of Performance

Our models predict the following outcomes

1. The speaker's exact Age - Regression Problem
2. The speaker's age group - Classification Problem
3. The speaker's Gender prediction - Classification Problem (Binary)

In the case of Age as a regression problem, our model should predict the continuous value, which should be compared to the ground truth label, with the goal of minimising the error.

In the case of a classification problem, gender has two values, male and female, whereas the age group has five values that the model must predict (teens, twenties, thirties, forties, and fifties). The classification problem's goal is to compare the predicted class to the ground truth label and make it predict the exact class as accurately as possible.

Regression Problem: In case of a regression problem, the RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) statistics are used, with lower values indicating better performance. Mathematically

$$RMSE = \sqrt{\frac{\sum(y - \hat{y})^2}{n}}$$

$$MAE = \frac{\sum|(y - \hat{y})|}{n}$$

Where:

y- Actual Value of Age
$\hat{y}$- Predicted Value of Age

Classification Problem: The preferred metric for a classification task is Accuracy, which generates probability values against the given ground truth label.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

TP- True Positive
TN- True Negative

FP- False Positive
FN- False Negative

Confusion Matrix was also used to understand the distribution of accuracy under each age group and each gender.

4. RESULTS AND DISCUSSION

4.1. Ideal duration of Audio

The audio sources in the dataset range in length from 1 to 80 seconds because they are compiled from multiple sources. To predict demographic data like age and gender from speaker audio that generalises across multiple different data sources, the ideal audio length must be determined.

We have chosen CNN+LSTM Based model with MFCC as a feature in which we have used the default number of MFCC as 40 to predict on the same test dataset with default padding and cropping strategy set to left. We have used RMSE and MAE values from the test sets as performance metric.

From the results in the Table 4. 5s chunks perform best in terms of metric. In the case of audio calls from the call centre use case, where the file length exceeds 600 seconds, extracting 5 seconds per speaker would be more than sufficient. This would save computational time an effort and real time response.

Table 4. Audio Length based Performance Assessment

Length of Audio vs Performance			**Custom Test Set**	
Model	**Feature**	**Duration (s)**	**RMSE**	**MAE**
CNN+LSTM	MFCC	3	5.21	3.37
CNN+LSTM	MFCC	5	4.55	2.69
CNN+LSTM	MFCC	10	5.16	3.34
CNN+LSTM	MFCC	15	5.14	3.35

4.2. Optimal Padding and Cropping Strategy

In the case of audio, the length of the 1D sequence must be the same. Our dataset contains variable-length audio sequences, and in real-world use cases, audio will be variable-length as a result of voice activity detection linked speaker diarisation. From the previous experiment we have investigated and understood the effect of audio duration on model performance using RMSE and MAE Metrics. 5s audio length is best suited for our use case.

It is also critical to comprehend the impact of underlying factors that cause the 5s duration to perform better. Padding and cropping are used to make the audio sequence the same length.

Table 5. displays the outcomes of various padding and cropping strategies. It is observed that the Padding of audio on the right and cropping at random yields better results when compared to other strategies for our use case.

Table 5. Padding and Cropping Vs Model Performance

Padding and Cropping Vs Model Performance				Custom Test Set	
Model	**Feature**	**Pad Crop Strategy**	**Duration(s)**	**RMSE**	**MAE**
CNN+LSTM	MFCC	pad - left, crop - random	5	5.21	3.37
CNN+LSTM	MFCC	pad - right, crop - random	5	4.55	2.69
CNN+LSTM	MFCC	pad - left, crop - left	5	5.16	3.34
CNN+LSTM	MFCC	pad - left, crop - right	5	5.14	3.35

4.3. Choosing the Number of MFCC

While using MFCC as the feature extractor for the audio parameter can be set to extract the number of features for every chunk of audio known as Number of MFCC features. By varying this number of MFCC we can also extract differential and acceleration components, also known as Delta and Delta-Delta Components The existing literature makes it clear that including these elements during feature extraction results in better ASR and other speech analytics Model performance.

We worked on varying the values on number of MFCC while keeping all other factors standard to try to find what generalises best for extracting demographic information from the speaker audio. Results from Table 6 verifies the fact that using the additional features of MFCC helps in improving the metric and hence achieving better results on overall custom dataset.

Table 6. Choosing number of MFCC

Selection of number of MFCC		Custom Test Set	
Model	**Hyperparameter (number of MFCC)**	**RMSE**	**MAE**
CNN+LSTM	13	5.49	3.7
CNN+LSTM	40	4.55	2.69

4.4. Experimental Results

4.4.1. CNN Model

The duration of the audio, padding, and cropping method during data ingestion, as well as number of MFCC during feature extraction, have now been identified. We have built models that leverage these values to predict the age, age group and gender information from the speaker audio.

The output from the feature extraction is then passed to the CNN based model followed by self attention. Output of the self attention then predicts all three values age, age group and gender of the input audio. From the Table 7 we can understand that the Wav2Vec feature is giving the best result in terms age RMSE (4.29) and MAE (2.83) for age prediction and for gender prediction the overall accuracy stands at 99.5% with male group having an accuracy of 99.8% and female with 99.1%.

Table 7. CNN Model Result

Feature	Age		Age Group	Gender		
	RMSE	MAE	Accuracy	Male	Female	All
MFCC	4.91	3.15	63.7%	98.8%	99.1%	98.9%
Wav2Vec	4.29	2.83	60.6%	99.8%	99.1%	99.5%

4.4.2. LSTM Model

In case of LSTM model, the input is received from the feature extraction technique selected returns the output. The output is then passed to the self attention layer. The output of self attention is then used to predict age, age group and gender.

From the results in Table 8 We have Wav2Vec Feature giving the best result. In case of LSTM model age group accuracy has increased by over 10% when compared to the CNN based model, with slight decrease in overall performance. We concluded that LSTM based model is able to capture the age group information better than the CNN Model.

Table 8. LSTM Model Result

Feature	Age		Age Group	Gender		
	RMSE	MAE	Accuracy	Male	Female	All
MFCC	5.89	4.16	50.8%	98.9%	97.1%	98.0%
Wav2Vec	4.42	2.66	69.6%	99.6%	99.1%	99.3%

4.4.3. CNN+LSTM Model

CNN Model provides the better result for age and gender prediction while LSTM model captures age group with higher accuracy. In order to improve the complexity and hence to improve the performance we have merged the above said two models CNN and LSTM. This gives our model the ability to learn the features of all the prediction variables better. Model Layer consists of Conv1d layer followed by LSTM and Soft Attention. Self attention then predicts age, age group and gender information.

As per the results in Table 9. MFCC feature is giving better accuracy in terms of age group prediction. In case of the Age and gender prediction the Wav2Vec feature extractor performs better with Age RMSE 4.25 and MAE 2.66, Gender overall accuracy is over 99.3% with female predicted 99.6% and male 99.0% prediction accuracy.

Table 9. CNN+LSTM Model Result

Feature	Age		Age Group	Gender		
	RMSE	MAE	Accuracy	Male	Female	All
MFCC	4.55	2.69	68.3%	99.3%	99.2%	99.2%
Wav2Vec	4.25	2.66	62.0%	99.0%	99.6%	99.3%

4.4.4. Multi CNN+LSTM Model

In case of Multi CNN + LSTM based model we have experimented with 2CNN+LSTM, 3CNN+LSTM results of which are discussed below.

Table 10 provides a relatively better performance in predicting the age and gender with Wav2Vec as feature. The RMSE and MAE values of MFCC feature hasn't improved from the previous model CNN+LSTM.

Table 10. 2CNN+LSTM Model Result

Feature	Age		Age Group	Gender		
	RMSE	MAE	Accuracy	Male	Female	All
MFCC	4.7	2.86	65.9%	99.2%	99.3%	99.2%
Wav2Vec	4.1	2.36	66.6%	99.2%	99.7%	99.5%

Table 11 shows that the results haven't improved much, and the learning has plateaued as the prediction on the test set doesn't provide a significant jump in performance.

We have also experimented with multiple other combinations like CNN + 2 LSTM and 2CNN+2LSTM etc. which doesn't seem to improve on the metrics of the predictive variables and has led us to confirm that the model performance couldn't be improved with variation of CNN and LSTM model.

Table 11. 3CNN+LSTM Model Result

Feature	Age		Age Group	Gender		
	RMSE	MAE	Accuracy	Male	Female	All
MFCC	4.66	2.80	63.8%	99.5%	99.1%	99.3%
Wav2Vec	4.11	2.45	65.9%	99.3%	99.4%	99.4%

4.5. Summary of Results

In Figure 3. results of various models with varying features are shown for Age Prediction. We can see that wav2vec has performed better in terms of RMSE value consistently better across all the model categories.

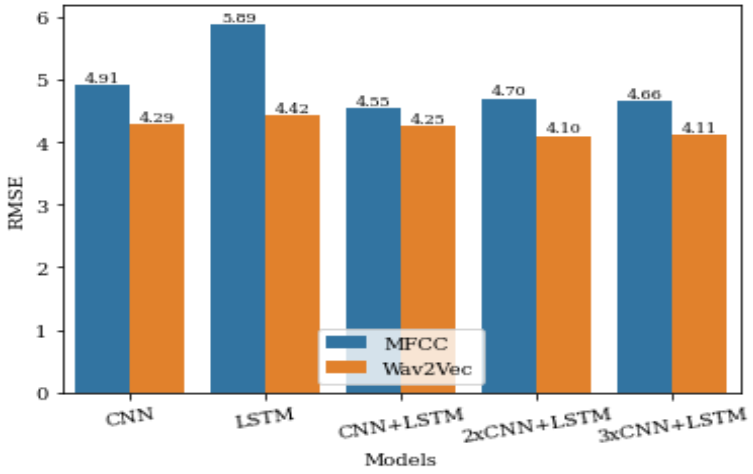


Figure 3. Age Prediction RMSE across models

In Figure 4. We have our models consistently performing better in terms of accuracy of gender prediction in case of wav2vec feature.

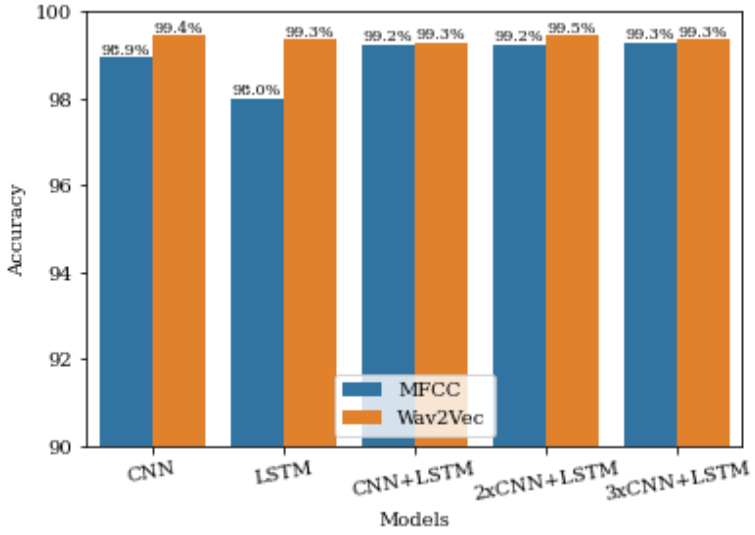


Figure 4. Gender Prediction Accuracy across models

5. CONCLUSION AND FUTURE WORK

In this paper, we have approached the speaker demographic information extraction from speech data by keeping a keen focus on the dataset. A custom dataset with WAV format and 16 KHz sampling rate was curated by selectively combining various standard datasets like TIMIT, NISP, VCTK as well as GMU, thereby improving the robustness of the model being developed to be used in enterprise grade applications. In this paper, we discussed various techniques to improve the model performance by tuning the various stages of model development such as Data Ingestion, Pre-Processing, Feature Extraction, and Model Selection and their effects on model performance. We understand from our research that audio input of a speaker as low as 5 seconds is enough to extract the age, gender and age group of the speaker.

We have explored various approaches with different input features as well as model parameters for age and gender prediction with our custom dataset. Our model capable to work with any speech data across globe from different geolocation. Also the same model can predict both age and gender from speech data. The results showcased that age, age group and gender prediction were best on Wav2vec + 2 CNN + LSTM.

Our future research will be focus on other speech features and wav2vec2 feature as model input features and also will expand the dataset to include more speaker coverages as well as multi language speech data across the globe.

REFERENCES

[1] D. M. Litvinov, "Speech analytics architecture for banking contact centers," in 10th Annual International Scientific and Practical Conference named after AI Kitov Information Technologies and Mathematical Methods in Economics and Management, IT and MM-CEUR Workshop Proceedings, 2021.

[2] S. a. Q. C. Scheidt, "Making a Case for Speech Analytics to Improve Customer Service Quality: Vision, Implementation, and Evaluation.," International Journal of Information Management 45., vol. 45, no. Elsevier: 223–32, 2019.

[3] L. Y, L. Y, D. H and e. al, "Speech databases for mental disorders: A systematic review," General Psychiatry, 2019.

[4] S. Safavi, M. Russell and P. Jančovič, "Automatic Speaker, Age-Group and Gender Identification from Children's Speech," Computer Speech & Language 50., no. Elsevier: 141–56, 2018.

[5] L. Jasuja, A. Rasoo and G. Hajela., "Voice Gender Recognizer Recognition of Gender from Voice Using Deep Neural Networks.," in International Conference on Smart Electronics and Communication (Icosec), 319–24. IEEE., 2020.

[6] M. Buyukyilmaz and A. O. Cibikdiken, "Voice gender recognition using deep learning," in Proceedings of 2016 International Conference on Modeling, Simulation and Optimization Technologies and Applications (MSOTA2016). Paris, France: Atlantis Press, 2016.

[7] T. Maka and P. Dziurzanski, "An Analysis of the Influence of Acoustical Adverse Conditions on Speaker Gender Identification.," in In XXII Annual Pacific Voice Conference (Pvc), 2014.

[8] A. J. Y. C. S. K. Tursunov, "Age and Gender Recognition Using a Convolutional Neural Network with a Specially Designed Multi-Attention Module Through Speech Spectrograms.," Sensors 21, 2021.

[9] H. A. Sánchez-Hevia, R. Gil-Pit, M. Utrilla-Manso and M. Rosa-Zurera, "Age Group Classification and Gender Recognition from Speech with Temporal Convolutional Neural Networks," Multimedia Tools and Applications., Vols. Springer, 1–18, 2022.

[10] S. B. Kalluri, D. Vijayasenan, S. Ganapathy and P. Krishnan, "NISP: A Multi-Lingual Multi-Accent Dataset for Speaker Profiling.," ICASSP 2021-2021 Ieee International Conference on Acoustics, Speech and Signal Processing (Icassp), 2021.

[11] L. F. L. W. M. F. J. G. F. a. D. S. P. J. S. Garofolo, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," NASA STI/Recon technical report, 1993.

[12] J. Yamagishi, C. Veaux and K. MacDonald, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," 2019.

[13] S. H. Weinberger and S. A. Kunath, "The Speech Accent Archive: towards a typology of English accents.," in In Corpus-based studies in language use, language learning, and language documentation.

[14] Y. A. Wubet and K.-Y. Lian, "A Hybrid Model of Cnn-Svm for Speakers' Gender and Accent Recognition Using English Keywords.," in 2021 Ieee International Conference on Consumer Electronics-Taiwan (Icce-Tw), 2021.

[15] S. B. Kalluri, D. Vijayasenan and S. Ganapathy., "utomatic Speaker Profiling from Short Duration Speech Data.," Speech Communication 121. Elsevier: 16–28, 2020.

[16] S. Schneider, A. Baevski, R. Collobert and M. Auli, "Wav2vec: Unsupervised Pre-Training for Speech Recognition.," arXiv, vol. 1904.05862, 2019.

[17] G. Van Rossum and F. L. Drake., "Python 3 Reference Manual," Scotts Valley, CA: CreateSpace, 2009.

[18] F. William, "The PyTorch Lightning team," Pytorch lightning, 2019.

[19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan and T. Killeen, "PyTorch: An Imperative Style, High-Performance Deep Learning Library.," In Advances in Neural Information Processing Systems 32, 2019.

AUTHORS

Veera Vignesh graduated from Birla Institute of Technology in 2018. Currently, he is a Data Scientist at Accenture Solutions India Pvt Ltd.

Anup Bera has done M.Tech from IIT Kharagpur in 2006 and he has total 16 years of industry experience and currently working at Accenture Solution India Pvt Ltd

DEEP LEARNING TECHNIQUE TO DENOISE ELECTROMYOGRAM ARTIFACTS FROM SINGLE-CHANNEL ELECTROENCEPHALOGRAM SIGNALS

Muhammad E. H. Chowdhury[1], Md Shafayet Hossain[2], Sakib Mahmud[1] and Amith Khandakar[1]

[1]Department of Electrical Engineering, Qatar University, Doha, 2713, Qatar
[2]Department of Electrical, Electronic and Systems Engineering, Universiti Kebangsaan Malaysia, Bangi, 43600, Selangor, Malaysia

ABSTRACT

The adoption of dependable and robust techniques to remove electromyogram (EMG) artifacts from electroencephalogram (EEG) is essential to enable the exact identification of several neurological diseases. Even though many classical signal processing-based techniques have been used in the past and only a few deep-learning-based models have been proposed very recently, it is still a challenge to design an effective technique to eliminate EMG artifacts from EEG. In this work, deep learning (DL) techniques have been used to remove EMG artifacts from single-channel EEG data by employing four popular 1D convolutional neural network (CNN) models for signal synthesis. To train, validate, and test four CNN models, a semi-synthetic publicly accessible EEG dataset known as EEGdenoiseNet has been used the performance of 1D CNN models has been assessed by calculating the relative root mean squared error (RRMSE) in both the time and frequency domain, the temporal and spectral percentage reduction in EMG artifacts and the average power ratios between five EEG bands to whole spectra. The U-Net model outperformed the other three 1D CNN models in most cases in removing EMG artifacts from EEG achieving the highest temporal and spectral percentage reduction in EMG artifacts (90.01% and 95.49%); the closest average power ratio for theta, alpha, beta, and gamma band (0.55701, 0.12904, 0.07516, and 0.01822, respectively) compared to ground truth EEG (0.5429; 0.13225; 0.08214; 0.002146; and 0.02146, respectively). It is expected from the reported results that the proposed framework can be used for real-time EMG artifact reduction from multi-channel EEG data as well.

KEYWORDS

EEG, EMG artifacts, Deep Learning, Single Channel, Denoising, Convolutional neural network.

1. INTRODUCTION

An electroencephalogram (EEG) signal is recorded non-invasively at the scalp which represents electrical pulses originating from the brain's electrophysiological activity [1]. EEG is important for several therapeutic uses and in neurological research. EEG is mostly used to diagnose epileptic seizures and Alzheimer's disease in humans [2, 3]. Other applications of EEG include drowsiness level measurement [4], detection of human emotions [5], estimation of cognitive workload [6], brain-computer interfaces (BCIs) [7], and biometric systems [8].

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 85-98, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122006

EEG recordings are highly susceptible to physiological noises such as myogenic artifacts [9], ocular artifacts [10], cardiac abnormalities [11], and non-physiological disturbances like power line noise and motion artifacts [12]. These noises will have a big effect on how the EEG data is analyzed, and in the worst cases, they could even lead to a completely wrong clinical diagnosis. To reduce artifacts and keep as much of the neural information as possible, an efficient and robust framework is vital which can remove EMG artifacts from EEG data.

The Fourier transform or the wavelet transform was commonly used to translate the signal from the temporal domain to the spectral domain and then filter out the spectral components that correspond to the EMG artifacts. The EMG artifacts-free EEG signal can be obtained using either the inverse Fourier transform or the inverse wavelet transform. The use of Wiener filter [13], adaptive filter [14], Hilbert-Huang Transformation (HHT) [15], empirical mode decomposition (EMD) [16], variational mode decomposition (VMD) [17], independent component analysis (ICA) [18], and canonical correlation analysis (CCA) [19] etc. are some other methods for EEG denoising. These methods aimed to transform the signal from its original space to a new space where the signal and noise can be separated. Some of the major disadvantages of utilizing a signal-processing-based algorithm to remove EMG artifacts from EEG recordings are the low percentage reduction in EMG artifacts removal, high possibility of removing important neural information from EEG by considering artifacts components, poor performance in dynamic situations, etc. Extended EMD (EEMD)-ICA and EEMD-CCA [20, 21] are two hybrid approaches that were reported to remove EMG artifacts from EEG. Although the integration of ICA/CCA along with EMD/EEMD improved the performance to some extent, the underlying assumptions remained unclear and requires optimization through trial and error basis. As an example, in the EEMD-CCA, the choice of two autocorrelation criteria is experimentally determined in diverse circumstances [22].

The development of novel network topologies and learning methods, as well as the increase in computer resources and enormous data processing capacity, have significantly enhanced the performance of deep learning (DL) neural networks in recent years. Several technological problems, including image processing [23] and natural language processing [24], have been effectively solved using DL. The categorization of motor imagery based on EEG [25] and the reconstruction of EEG [26] are two notable applications of EEG-related analysis that make use of DL methods.

EEG artifacts have also been eliminated using certain well-known deep learning (DL) models, including auto-encoder [27], convolutional neural networks (CNNs) [28, 29], and recurrent neural networks [29, 30]. In comparison to conventional signal processing-based models, DL models have the following two advantages: (i) universality, where a standardized architecture can accommodate a variety of artifact removal tasks without the need for manual designs of prior assumptions on a particular type of artifacts; and (ii) higher capacity, where deep learning enforces a significant performance improvement. As expected, a significant performance boost compared to the traditional signal-processing-based methods was reported in these studies yet some of the shortcomings of these deep learning-based approaches are : testing the proposed models with a partial dataset rather than using the k-fold cross-validation approach, lack of generability of the model, room for a further performance boost, smaller number of performance metrics for evaluating the DL models, etc which are addressed properly in this study.

The goal of this study is to create an efficient and robust DL model that can clean out EMG artifacts from EMG-corrupted EEG data. To denoise EMG artifacts from contaminated EEG signals, four different 1D CNN models, namely (i) Feature Pyramid Network, or FPN [31], (ii) U-Net [32], (iii) Multi-level Context Gating U-Net (MCGU-Net) [33], and (iv) LinkNet [34] have been implemented in this work. The efficacy of the models is then compared by computing

several performance metrics, using the publicly available semi-synthetic dataset, EEGdenoiseNet. Our main contributions are listed here, briefly:

- The outcomes of our thorough analysis provide a convincing demonstration of the effectiveness of 1D CNN models in eliminating EMG artifacts from noisy EEG.
- The four 1D-CNN models are trained and tested using a five-fold cross-validation method and consistent results obtained from each fold ensure the reliability and robustness of the proposed framework.
- After being trained on single-channel, semi-synthetic EEG data, our created framework may be utilized to denoise multichannel, real-world EEG data.

The rest of this article is organized as follows: The single-channel EEG dataset, semi-synthetic corrupted EEG data creation, and normalization approaches used in this research are described in Section 2. The experimental setup and the performance assessment metrics are covered in Section 3. In Section 4, the performance of the proposed four different cutting-edge 1D CNN segmentation networks is quantified, and the findings are thoroughly discussed. The limitations of our research are described in Section 5 and finally, a succinct conclusion is presented in section 6.

2. MATERIALS AND METHODS

The framework for efficient EEG denoising (removing EMG artifacts from contaminated EEG) utilizing 1D-CNN-based segmentation networks is shown in **Figure 1** and is self-explanatory. In two distinct subsections, the dataset utilized, the semi-synthetic data creation, and the normalization processes employed in this work are elaborately explained below:

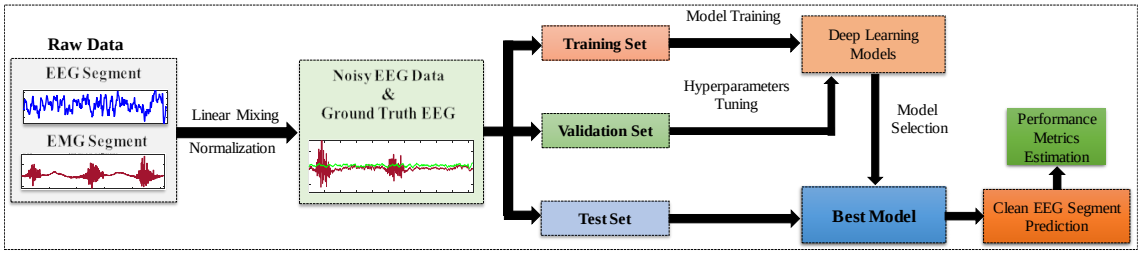


Figure 1. A proposed framework to remove EMG artifacts from corrupted EEG signals.

2.1. Dataset Description

The EEGdenoiseNet, a semi-synthetic EEG dataset is used in this study [29]. The noisy EEG and equivalent clean EEG provided by EEGdenoiseNet are used for model training and quantitative performance assessment. It may be used to assess how well different models generalize in real-world circumstances. The EEGdenoiseNet dataset includes 4,514 clean EEG signal segments and 5,598 clean EMG signal segments. Each clean EEG and EMG signal segment is 2 seconds long.

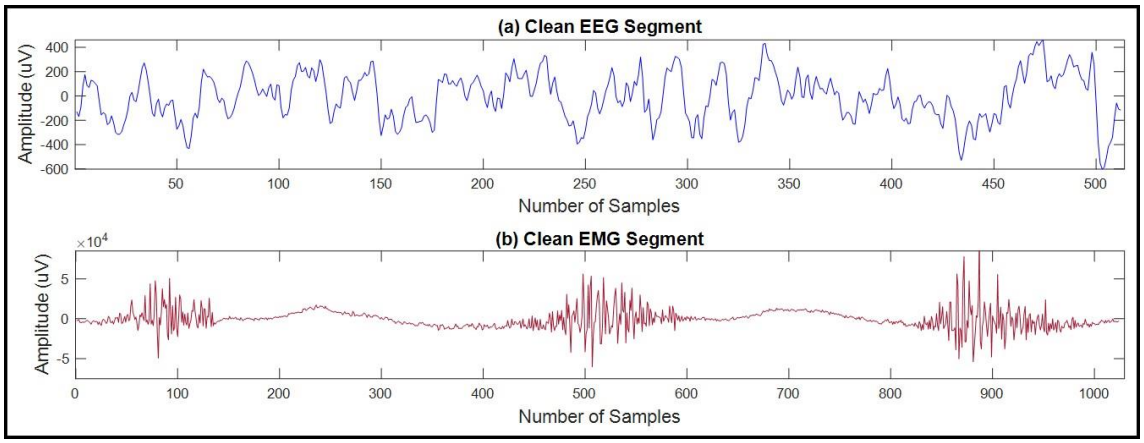


Figure 2. Example segments in EEGdenoiseNet dataset. (a) Clean/Ground truth EEG segment, (b) Clean EMG segment.

The raw EEG signals were pre-processed with a band-pass filter (1-80 Hz) to remove unnecessary high-frequency components of EEG, a notch filter (50 Hz) to remove the powerline noises, and re-sampled at 256 Hz by the authors of the dataset [29]. The EEG signals were segmented at this stage, with each segment containing 512 data points (2 seconds). It is important to note that single-channel EEG signals were used to construct the clean EEG segments. Raw face EMG signals were resampled to 512 Hz after being band-pass filtered (bandwidth: 1-120 Hz) and notch filtered at the power-line frequency (50 Hz). Because the EMG signal is concentrated in the high-frequency band, the EMG signals were sampled at 512 Hz instead of 256 Hz to preserve the high-frequency component of EMG signals. Finally, one-dimensional, 2-second EMG segments with 1024 data points per segment were generated. **Figure 2** depicts one clean EEG, and one clean EMG segment for visual illustration.

2.2. Semi-Synthetic Contaminated EEG Data Generation and Normalization

In this study, the clean EEG segments are linearly mixed with EMG segments (**Equation 1)** to produce the semi-synthetic EMG contaminated EEG:

$$y = x + \lambda.n \tag{1}$$

where x denotes the clean EEG segment/ground truth EEG; n is the clean EMG segment (as noise), and y is the semi-synthetic EMG contaminated EEG segment. The signal-to-noise ratio (SNR) of the EMG corrupted EEG signal can be varied by changing the value of the scaling factor λ by solving **Equation 2**:

$$SNR = 10log\frac{RMS(x)}{RMS(\lambda.n)} \tag{2}$$

The root mean square (RMS) can be calculated as follows:

$$RMS\,(z) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} z_i^2} \tag{3}$$

where N is the number of time-domain samples of segment, z and z_i is the i_{th} sample point of segment, z. It is apparent from **Equation 2** that a smaller λ corresponds to lesser EMG artifacts and hence a lower λ indicates a higher SNR. Similarly, higher noise level results in poorer SNR.

The SNR of an EEG that has been contaminated by EMG artifacts typically ranges from -7dB to 2dB [35].

In this work, the linear mixing process enables us to generate a pair of EEG segments (x, y) where the clean/ground truth EEG (x) can be utilized as the outputs and the semi-synthetic EMG contaminated EEG (y) as the inputs to train the end-to-end deep learning models. However, in this study, (x, y) pairs are not directly fed to 1D CNN models. Rather, both the EMG contaminated EEG and ground truth EEG segments are divided by the standard deviation of the EMG contaminated EEG segments (σ_y) and the resultant **(Equation 4**) normalized segments $(\hat{x}, \hat{y})$ are utilized as inputs and outputs to facilitate the learning procedure of the deep learning models.

$$\hat{x} = \frac{x}{\sigma_y} \; ; \; \hat{y} = \frac{y}{\sigma_y} \tag{4}$$

4,514 clean EEG segments and 5,598 EMG segments are mixed while maintaining a particular SNR to produce the EMG-contaminated EEG segments. 4,514 clean EMG segments are chosen at random from a pool of 5,598 EMG segments to match the number of the EEG signal segments. Before combining EEG and EMG segments linearly, all the clean EEG segments are up-sampled at 512 Hz to match the sampling frequency of EMG segments. We repeat this procedure ten times for ten different SNR levels (-7 dB, -6 dB, -5 dB, -4 dB, -3 dB, -2 dB, -1 dB, 0 dB, 1 dB, and 2 dB), giving us 45,140 EMG-contaminated EEG segments. **Figure 3** shows a randomly selected (out of 45,140 segments) EMG contaminated EEG segment. In **Figure 3**, the corresponding ground truth EEG segment is also overlaid to better visualize the impact of EMG contamination in EEG.

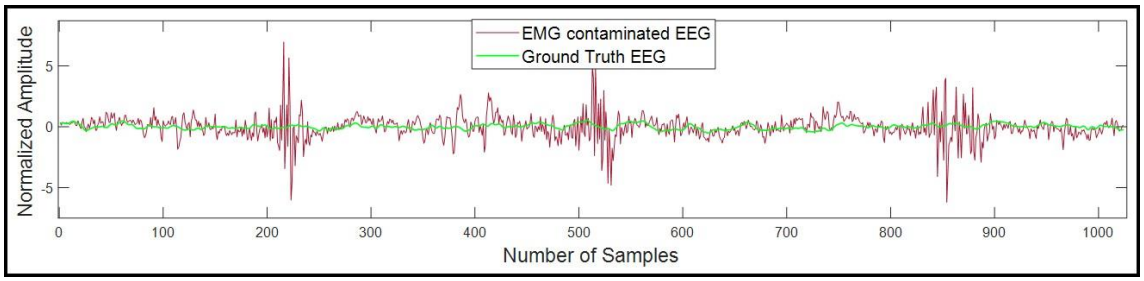


Figure 3. Example segments of EMG contaminated EEG and corresponding overlaid ground truth EEG.

3. EXPERIMENT DETAILS

The four 1D CNN models (FPN, U-Net, MCGU-Net, and LinkNet) are trained in an end-to-end framework, i.e., the normalized contaminated EEG segments $(\hat{y})$ are fed into the neural networks as input and corresponding ground truth EEG segments $(\hat{x})$ are fed as output so that the DL models can develop a nonlinear function that maps the contaminated EEG segments $(\hat{y})$ to the ground truth EEG segments $(\hat{x})$. During training, the mean squared error (MSE) is selected as the loss function, ADAM optimizer is applied to optimize the loss function. The learning rate α is set to 0.0005. The training set consists of 80% of the data, while the test set consists of the remaining 20%. The validation set is selected from the 10% of the training set. It is important to note that each EEG and EMG segment in the dataset is acquired from a distinct individual, hence the 80-20-10 splitting results in no data leakage across the train, test, and validation sets. In the Google ColabPro environment, the five-fold cross-validation approach is adopted to independently train, validate, and test all four networks separately. Two distinct experiments are conducted in this research and are explained below:

3.1. Experiment A

As previously mentioned, 4,514 EEG segments contaminated with EMG artifacts are created for each of the 10 SNR values (-7 dB, -6 dB, -5 dB, -4 dB, -3 dB, -2 dB, -1 dB, 0 dB, 1 dB, and 2 dB), yielding 45,140 EEG segments overall. The EMG-corrupted EEG and associated ground truth EEG segments are used to train each of the four 1D CNN models 10 times for a total of 10 distinct SNR values, separately. This means that at SNR level -7 dB, 80% of the 4,514 segments (3,611) are used as the training set, while the remaining 20% of the 4,514 segments (903) are used as the test set. Similarly, segments generated from nine more SNR levels (-6 dB, -5 dB, -4 dB, -3 dB, -2 dB, -1 dB, 0 dB, 1 dB, and 2 dB) are individually subjected to the same procedure, separately. Table 1 consists of the information for train and test sets formation.

Table 1. Formation of train and test sets for experiment B

SNR level (in dB)	Total Number of ground truth EEG and EMG contaminated EEG	Train set (80% of ground truth EEG and EMG contaminated EEG pairs)	Test set (20% of ground truth EEG and EMG contaminated EEG pairs)
-7	4,514	3,611	903
-6	4,514	3,611	903
-5	4,514	3,611	903
-4	4,514	3,611	903
-3	4,514	3,611	903
-2	4,514	3,611	903
-1	4,514	3,611	903
0	4,514	3,611	903
+1	4,514	3,611	903
+2	4,514	3,611	903

To assess the effectiveness of all four DL models in removing EMG noises from contaminated EEG, Deep Supervision [36] is applied, and three well-known performance metrics (correlation coefficient, relative root mean squared error in the time and frequency domain) are computed.

3.2. Experiment B

In the second experiment, the training and test sets are created in a more controlled setting. For each of the ten distinct SNR levels, eighty percent EMG corrupted EEG segments and their corresponding ground truth EEG segments are chosen for each of the ten distinct SNR levels (3,611 pairs of ground truth EEG and EMG contaminated EEG for each SNR level), and then all the 10 × 3,611 pairings are combined to form the training set of 36,110 pairs of ground truth EEG and EMG contaminated EEG. The remaining 20% EMG-contaminated EEG segments and their corresponding ground truth EEG segments are also formed (903 pairings of ground truth EEG and EMG contaminated EEG for each SNR level) and combined (10 × 903) to create the test set of 9,030 pairs. Table 2 represents the formation of train and test sets.

Table 2. Formation of train and test sets for experiment B

SNR level(in dB)	Total Number ofGround Truth EEG and EMG Contaminated EEG	80% of Ground Truth EEG and EMG Contaminated EEG	Total Number of Pairs forTrain Set	20% of Ground Truth EEG and EMG Contaminated EEG	Total Number of Pairs for Test Set
-7	4,514	3,611	36,110	903	9,030
-6	4,514	3,611		903	
-5	4,514	3,611		903	
-4	4,514	3,611		903	
-3	4,514	3,611		903	
-2	4,514	3,611		903	
-1	4,514	3,611		903	
0	4,514	3,611		903	
+1	4,514	3,611		903	
+2	4,514	3,611		903	

In experiment A, DL models are trained and evaluated for certain SNR level segments; as a result, they are not capable of performing well if tested with other SNR-level EEG segments. This is the main difference between experiments A and B. Since there is no way to determine the SNR level of the applied input signal in real-world circumstances, models trained as experiment B should be more reliable when evaluated with EMG-contaminated EEG segments having different SNR levels.

In experiment B, all four 1D CNN segmentation networks are trained with and without Deep supervision [36]. It is found that the deep supervision did not improve the model performance, hence we have not reported the results utilizing deep supervision for this experiment. Unlike experiment A, the performance of the trained models is quantified using five performance metrics (average power ratio between five different EEG bands to the whole spectra, relative root mean squared error in the time domain, relative root mean squared error in the frequency domain, the temporal percentage reduction in EMG artifacts, and lastly, the spectral percentage reduction in EMG artifacts.

3.3. Performance Metrics

It is key to quantify the model's performance to ascertain how well it can predict the signal. In addition, identifying and selecting the appropriate performance metrics is crucial. In this regard, to measure the adeptness of the four 1D CNN models quantitatively, the correlation coefficient (CC) in the time domain, the percentage reduction in artifacts in the time and frequency domain, the temporal and spectral relative root mean squared error (RRMSE) is measured using the following formulas which can be found in Equation 5-9 [12, 29]:

$$CC_{temporal} = \frac{Cov(\hat{z},\hat{x})}{\sqrt{Var(\hat{z})Var(\hat{x})}} \quad (5)$$

$$\eta = 100\ (1 - \frac{1-CC_{temporal(after)}}{1-CC_{temporal(before)}}) \quad (6)$$

$$\gamma = 100\ (1 - \frac{1-CC_{spectral(after)}}{1-CC_{spectral(before)}}) \quad (7)$$

$$RRMSE_{temporal} = \frac{RMS(\hat{z}-\hat{x})}{RMS(\hat{x})} \quad (8)$$

$$RRMSE_{spectral} = \frac{RMS(PSD(\hat{z})-PSD(\hat{x}))}{RMS(PSD(\hat{x}))} \quad (9)$$

Where, Cov stands for covariance, Var means the variance, $\hat{x}$ is the normalized ground truth EEG, $\hat{z}$ is the predicted segment, η stands for the temporal percentage reduction in EMG artifacts, γ is the spectral percentage reduction in EMG artifacts, $CC_{temporal(after)}$ is the temporal CC between predicted and ground truth EEG segment, $CC_{temporal(before)}$ is the temporal CC between contaminated and ground truth EEG segment, $CC_{spectral(after)}$ is the frequency domain CC between predicted and ground truth EEG segment, $CC_{spectral(before)}$ represents the frequency domain CC between contaminated and ground truth EEG segment, and PSD is the power spectral density.

In addition to the aforementioned performance metrics, the average power ratio of each EEG frequency band (delta [1-4 Hz], theta [4-8 Hz], alpha [8-13 Hz], beta [13-30 Hz], and gamma [30-80 Hz] bands) to the whole band (1-80 Hz) is also computed and reported for the EMG contaminated EEG, ground truth EEG, and predicted EEG segments.

4. RESULTS AND DISCUSSION

The results of experiments A and B are described and discussed in detail in this section.

4.1. Results of Experiment A

Figure 4 depicts the average temporal correlation coefficient values between predicted denoised EEG segments (after the removal of EMG artifacts) and the ground truth EEG segments for ten distinct SNR values ranging from -7 dB to +2 dB obtained from four different 1D CNN models separately. The RRMSE values in the temporal and spectral domain are also plotted against ten distinct SNR levels in the same figure.

For EMG artifacts removal from EEG, all four 1D CNN models performed very close as is evident from Figure 4 where almost overlapping curves of performance metrics can be seen. Overall, for EMG artifacts removal from noisy EEG, the performance of the DL models improved (higher CC, lower RRMSE value in temporal and spectral domain) with the increment of SNR level and vice versa, which is expected. As the SNR level increases, EMG noise also reduces proportionally and therefore DL models face lesser challenges in developing the non-linear mapping function for predicting denoised EEG segments. In this process, the performance of the models improves for higher SNR levels and vice versa.

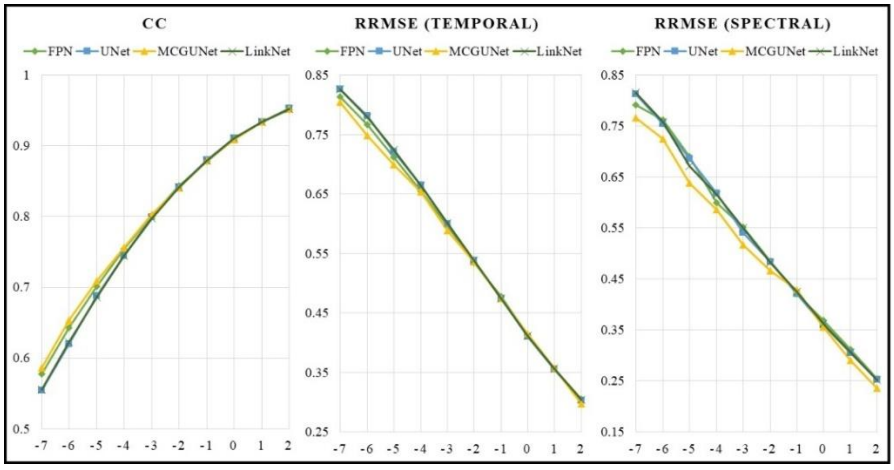


Figure 4. Performance parameters of four 1D CNN models at ten different SNR levels after denoising EMG-contaminated EEG.

4.2. Results of Experiment B

The goal of the four 1D segmentation networks is to remove/reduce EMG artifacts from the EEG segments. To provide a qualitative assessment of the denoising models used, Figure 5(a) shows one EMG-contaminated EEG segment whereas Figure 5(b)–(e) shows EMG artifacts-free EEG (predicted EEG) segments for all four models.

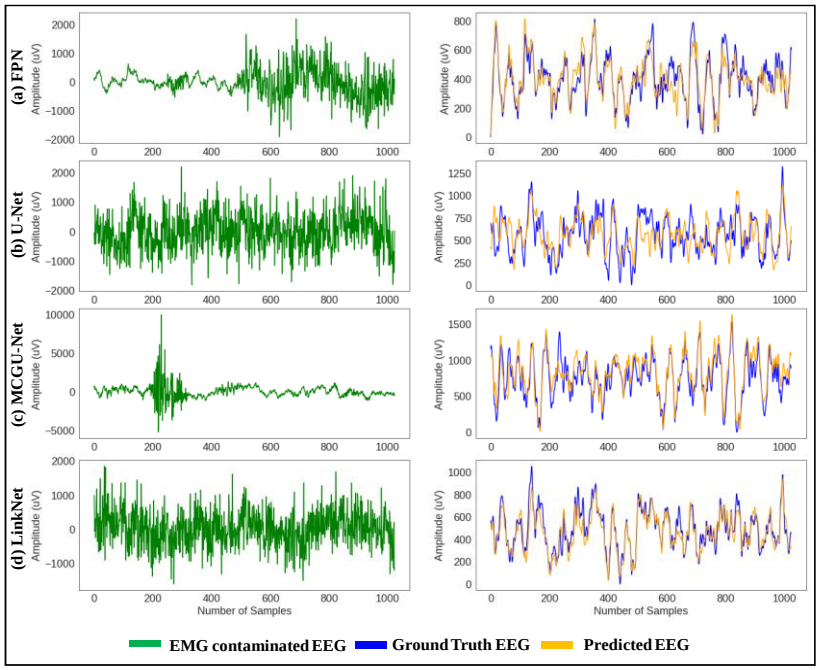


Figure 5. (a) Example EMG contaminated EEG segment; EMG artifacts-free EEG segments predicted by (b) FPN, (c) U-Net, (d) MCGU-Net, and (e) LinkNet networks overlaid with the ground truth EEG.

In Table 3, Temporal and spectral percentage reduction in EMG artifacts, as well as Temporal and spectral RRMSE values are summarized for the prediction of EMG artifacts free EEG segments by the four different 1D CNN models separately. For EMG corrupted EEG segments, the best denoising (elimination of EMG artifacts) performance is generated by the U-Net model. In the time domain, 90.01% reduction in EMG artifacts is found whereas, in the frequency domain, 95.49% EMG artifacts removal is observed while U-Net is utilized. Again, the very same model produced the lowest RRMSE values both in the time and frequency domain (0.10042 and 0.20276, respectively) in comparison with the other three 1D CNN networks. While the U-Net model dominates in producing better denoising performance, LinkNet produced the lowest temporal percentage reduction in EMG artifacts (85.09%) along with the highest temporal RRMSE (0.15681) wheras the FPN model produced lowest spectral percentage reduction in EMG artifacts (91.31%) along with the highest spectral RRMSE (0.31715).

Table 3. Performance parameters of 1D CNN models after denoising EMG-contaminated EEG. The asterisk (*) sign represents the best performing results

Model Name	**Temporal Percentage Reduction in EMG Artifacts (in %)**	**Spectral Percentage Reduction in EMG Artifacts (in %)**	**Temporal RRMSE +/- Std. Deviation**	**Spectral RRMSE +/- Std. Deviation**
FPN [31]	86.31	91.31	0.14131 +/- 0.08200	0.31715 +/- 0.26364
U-Net [32]	90.01*	95.49*	0.10042 +/- 0.07001*	0.20276 +/- 0.14862*
MCGU-Net [33]	89.10	94.13	0.13811 +/- 0.08458	0.28561 +/- 0.23205
LinkNet[34]	85.09	91.88	0.15681 +/- 0.08843	0.30818 +/- 0.18860

Table 4 represents the average power ratios calculated between five different EEG bands (Delta, Theta, Alpha, Beta, Gamma) before and after the elimination of EMG artifacts. The closer value of the average power ratios of different bands compared with ground truth EEG signifies the close resemblance and data fidelity while predicted by the deep learning models. We found that the U-Net model dominates in producing the closest average power ratios for theta, alpha, beta, and gamma-band while compared with the ground truth EEG and for delta band power ratio, MCGU-Net performed best whereas the EMG contaminated EEG produced the worst average power band ratios for all the five bands due to the incorporation of EMG noises, as expected.

Table 4. Average power ratios of five EEG frequency bands before and after the denoising of EMG-contaminated EEG. The asterisk (*) sign represents the best-performing results.

Model/ Method	**Delta**	**Theta**	**Alpha**	**Beta**	**Gamma**
FPN [31]	0.41474	0.5641	0.12416	0.06783	0.0155
U-Net [32]	0.40727	0.55701*	0.12904*	0.07516*	0.01822*
MCGU-Net [33]	0.40725*	0.57707	0.12354	0.06745	0.01351
LinkNet[34]	0.41106	0.56383	0.12619	0.0697	0.01604
EMG contaminated EEG	0.1331	0.10418	0.06205	0.20869	0.55126
Ground Truth EEG	0.40504	0.5429	0.13225	0.08214	0.02146

Due to the temporal complexity and presence of high-frequency components in EMG signals, removing EMG artifacts is always more challenging compared to the removal of other physiological artifacts from EEG recordings [37]. Similar phenomena are also observed in this well-crafted study. However, the findings from this work show that EEG artifact reduction DL models are very dependable owing to the amazing flexibility and strong generalizability of deep learning models. Although the DL-based denoising algorithms need a significant quantity of ground truth EEG data during the training phase, they may be used consistently to remove EMG artifacts from noisy EEG signals after the model has been trained.

The use of a semi-synthetic dataset is also justified by the fact that the presence of ground truth EEG is required for assessing 1D CNN models' performance. In this study, the EMG segments that are linearly mixed with EEG segments were collected from the real-life test subjects rather than producing it artificially which further ensures the efficacy of the proposed models.

RRMSE in the time and frequency domain, as well as CC, have been calculated to assess the efficacy of the four distinct 1D segmentation networks in predicting artifact-free EEG data, as also reported in [29]. Additionally, to measure the effectiveness of DL models, we have included two new performance criteria in this study, namely the percentage decrease in EMG artifacts in

the time and frequency domains separately. We firmly believe that including the three-performance metrics mentioned before with these new two would help standardize the quantitative performance evaluation of DL models to a great extent.

5. CONCLUSIONS

In this paper, four 1D CNN models i.e. FPN, U-Net, MCGU-Net, and LinkNet are utilized to remove EMG artifacts from EEG signal. While using U-Net, the EMG artifacts from EMG-corrupted EEG are decreased by 90.01% and 95.49%, in the time and frequency domains, respectively. The lowest temporal and spectral RRMSE (0.1 and 0.2, respectively) for the denoising of EMG artifacts from contaminated EEG segments are produced by the U-Net model. Also, the same model produced the closest average power ratio for the theta, alpha, beta, and gamma band compared with the ground truth EEG while removing EMG artifacts compared to the other three models. The findings of this work provide a convincing demonstration of the reliability of 1D CNN models in removing EMG artifacts caused by tainted EEG data. Our results demonstrated that DL approaches have a significant ability to remove EMG artifacts from EEG data, even at high noise levels.

ACKNOWLEDGEMENTS

The dataset used in this research work is kindly shared by Zhang et al. [29]. This research is financially supported by Qatar National Research Foundation (QNRF), Grant number NPRP12s-0227-190164 and International Research Collaboration Co-Fund (IRCC) grant: IRCC-2021-001. The statements made herein are solely the responsibility of the authors.

REFERENCES

[1] J. R. Wolpaw, D. J. McFarland, G. W. Neat, and C. A. Forneris, "An EEG-based brain-computer interface for cursor control," Electroencephalography and clinical neurophysiology, vol. 78, no. 3, pp. 252-259, 1991.

[2] Y. Zhang, Y. Guo, P. Yang, W. Chen, and B. Lo, "Epilepsy seizure prediction on EEG using common spatial pattern and convolutional neural network," IEEE journal of biomedical and health informatics, vol. 24, no. 2, pp. 465-474, 2019.

[3] S. Yang, J. M. S. Bornot, K. Wong-Lin, and G. Prasad, "M/EEG-based bio-markers to predict the MCI and Alzheimer's disease: a review from the ML perspective," IEEE Transactions on Biomedical Engineering, vol. 66, no. 10, pp. 2924-2935, 2019.

[4] C. Berka et al., "Real-time analysis of EEG indexes of alertness, cognition, and memory acquired with a wireless EEG headset," International Journal of Human-Computer Interaction, vol. 17, no. 2, pp. 151-170, 2004.

[5] V. Gupta, M. D. Chopda, and R. B. Pachori, "Cross-subject emotion recognition using flexible analytic wavelet transform from EEG signals," IEEE Sensors Journal, vol. 19, no. 6, pp. 2266-2274, 2018.

[6] P. Antonenko, F. Paas, R. Grabner, and T. Van Gog, "Using electroencephalography to measure cognitive load," Educational psychology review, vol. 22, no. 4, pp. 425-438, 2010.

[7] P. Gaur, R. B. Pachori, H. Wang, and G. Prasad, "An automatic subject specific intrinsic mode function selection for enhancing two-class EEG-based motor imagery-brain computer interface," IEEE Sensors Journal, vol. 19, no. 16, pp. 6938-6947, 2019.

[8] A. Rahman et al., "Robust biometric system using session invariant multimodal EEG and keystroke dynamics by the ensemble of self-ONNs," Computers in Biology and Medicine, vol. 142, p. 105238, 2022.

[9] B. W. McMenamin et al., "Validation of ICA-based myogenic artifact correction for scalp and source-localized EEG," Neuroimage, vol. 49, no. 3, pp. 2416-2432, 2010.

[10] A. Flexer, H. Bauer, J. Pripfl, and G. Dorffner, "Using ICA for removal of ocular artifacts in EEG recorded from blind subjects," Neural Networks, vol. 18, no. 7, pp. 998-1005, 2005.

[11] J. Jorge, C. Bouloc, L. Bréchet, C. M. Michel, and R. Gruetter, "Investigating the variability of cardiac pulse artifacts across heartbeats in simultaneous EEG-fMRI recordings: A 7T study," NeuroImage, vol. 191, pp. 21-35, 2019.
[12] M. S. Hossain et al., "Motion Artifacts Correction from Single-Channel EEG and fNIRS Signals Using Novel Wavelet Packet Decomposition in Combination with Canonical Correlation Analysis," Sensors, vol. 22, no. 9, p. 3169, 2022.
[13] B. Somers, T. Francart, and A. Bertrand, "A generic EEG artifact removal algorithm based on the multi-channel Wiener filter," Journal of neural engineering, vol. 15, no. 3, p. 036007, 2018.
[14] A. G. Correa, E. Laciar, H. Patiño, and M. Valentinuzzi, "Artifact removal from EEG signals using adaptive filters in cascade," in Journal of Physics: Conference Series, 2007, vol. 90, no. 1, p. 012081: IOP Publishing.
[15] A. Mert and A. Akan, "Hilbert-Huang transform based hierarchical clustering for EEG denoising," in 21st European Signal Processing Conference (EUSIPCO 2013), 2013, pp. 1-5: IEEE.
[16] M. Marino et al., "Adaptive optimal basis set for BCG artifact removal in simultaneous EEG-fMRI," Scientific reports, vol. 8, no. 1, pp. 1-11, 2018.
[17] M. S. Hossain et al., "Motion Artifacts Correction from EEG and fNIRS Signals using Novel Multiresolution Analysis," IEEE Access, vol. 10, pp. 29760-29777, 2022.
[18] E. Urrestarazu, J. Iriarte, M. Alegre, M. Valencia, C. Viteri, and J. Artieda, "Independent component analysis removing artifacts in ictal recordings," Epilepsia, vol. 45, no. 9, pp. 1071-1078, 2004.
[19] W. De Clercq, A. Vergult, B. Vanrumste, W. Van Paesschen, and S. Van Huffel, "Canonical correlation analysis applied to remove muscle artifacts from the electroencephalogram," IEEE Transactions on Biomedical Engineering, vol. 53, no. 12, pp. 2583-2587, 2006.
[20] K. Zeng, D. Chen, G. Ouyang, L. Wang, X. Liu, and X. Li, "An EEMD-ICA approach to enhancing artifact rejection for noisy multivariate neural data," IEEE transactions on neural systems and rehabilitation engineering, vol. 24, no. 6, pp. 630-638, 2015.
[21] X. Chen, Q. Chen, Y. Zhang, and Z. J. Wang, "A novel EEMD-CCA approach to removing muscle artifacts for pervasive EEG," IEEE Sensors Journal, vol. 19, no. 19, pp. 8420-8431, 2018.
[22] K. T. Sweeney, S. F. McLoone, and T. E. Ward, "The use of ensemble empirical mode decomposition with canonical correlation analysis as a novel artifact removal technique," IEEE transactions on biomedical engineering, vol. 60, no. 1, pp. 97-105, 2012.
[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, vol. 25, 2012.
[24] A. Vaswani et al., "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.
[25] Z. Tang, C. Li, and S. J. O. Sun, "Single-trial EEG classification of motor imagery using deep convolutional neural networks," Optik, vol. 130, pp. 11-18, 2017.
[26] T.-j. Luo, Y. Fan, L. Chen, G. Guo, and C. Zhou, "EEG signal reconstruction using a generative adversarial network with wasserstein distance and temporal-spatial-frequency loss," Frontiers in neuroinformatics, vol. 14, p. 15, 2020.
[27] N. M. N. Leite, E. T. Pereira, E. C. Gurjao, and L. R. Veloso, "Deep convolutional autoencoder for EEG noise filtering," in 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2018, pp. 2605-2612: IEEE.
[28] W. Sun, Y. Su, X. Wu, and X. Wu, "A novel end-to-end 1D-ResCNN model to remove artifact from EEG signals," Neurocomputing, vol. 404, pp. 108-121, 2020.
[29] H. Zhang, M. Zhao, C. Wei, D. Mantini, Z. Li, and Q. Liu, "Eegdenoisenet: A benchmark dataset for deep learning solutions of eeg denoising," Journal of Neural Engineering, vol. 18, no. 5, p. 056057, 2021.
[30] H. Zhang, C. Wei, M. Zhao, Q. Liu, and H. Wu, "A novel convolutional neural network model to remove muscle artifacts from EEG," in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 1265-1269: IEEE.
[31] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117-2125.
[32] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in International Conference on Medical image computing and computer-assisted intervention, 2015, pp. 234-241: Springer.

[33] M. Asadi-Aghbolaghi, R. Azad, M. Fathy, and S. J. a. p. a. Escalera, "Multi-level context gating of embedded collective knowledge for medical image segmentation," 2020.
[34] A. Chaurasia and E. Culurciello, "Linknet: Exploiting encoder representations for efficient semantic segmentation," in 2017 IEEE Visual Communications and Image Processing (VCIP), 2017, pp. 1-4: IEEE.
[35] X. Chen, H. Peng, F. Yu, and K. Wang, "Independent vector analysis applied to remove muscle artifacts in EEG data," IEEE transactions on instrumentation and measurement, vol. 66, no. 7, pp. 1770-1779, 2017.
[36] Q. Zhu, B. Du, B. Turkbey, P. L. Choyke, and P. Yan, "Deeply-supervised CNN for prostate segmentation," in 2017 international joint conference on neural networks (IJCNN), 2017, pp. 178-184: IEEE.
[37] Z.-Q. J. Xu, Y. Zhang, T. Luo, Y. Xiao, and Z. Ma, "Frequency principle: Fourier analysis sheds light on deep neural networks," arXiv preprint arXiv:.06523, 2019.

AUTHORS

Muhammad E. H. Chowdhury received his Ph.D. degree from the University of Nottingham, U.K., in 2014. He worked as a Postdoctoral Research Fellow at the Sir Peter Mansfield Imaging Centre, University of Nottingham. He is currently working as an Assistant Professor with the Department of Electrical Engineering, Qatar University. He has filed several patents and published more than 100 peer-reviewed journal articles, conference papers, and several book chapters. His current research interests include biomedical instrumentation, signal processing, wearable sensors, medical image analysis, machine learning and computer vision, embedded system design, and simultaneous EEG/fMRI. He is currently running several NPRP, UREP, and HSREP grants from Qatar National Research Fund (QNRF) and internal grants (IRCC and HIG) from Qatar University along with academic projects from HBKU and HMC. He has been involved in EPSRC, ISIF, and EPSRC-ACC grants along with different national and international projects during his tenure at the University of Nottingham. He is a Senior Member of IEEE, and a member of British Radiology, ISMRM, and HBM. He is serving as an Associate Editor for IEEE Access and a Topic Editor and Review Editor for Frontiers in Neuroscience. He has recently won the COVID-19 Dataset Award, AHS Award from HMC, and National AI Competition awards for his contribution to the fight against COVID-19.

Md Shafayet Hossain received his B.Sc. in Electrical and Electronic Engineering from the Islamic University of Technology (IUT), Dhaka, Bangladesh. Currently, he is pursuing an M.Sc. in Universiti Kebangsaan Malaysia (UKM) focusing on the removal/reduction of motion artifacts from physiological signals (especially from EEG signals) using traditional signal processing and machine learning-based models. Mr. Shafayet is an active member of Qatar University Machine Learning Group (initiated and supervised by Dr. Muhammad E. H. Chowdhury) from October 2020 to the present. He also worked as a Research Assistant at Qatar University from May 2021 to September 2021. He joined the Department of EEE of Bangladesh University of Business and Technology, Dhaka, Bangladesh as a Lecturer in 2015 and is currently on study leave. His research interests include biomedical signal processing, machine learning & deep learning, bio-sensor design, green energy harvesting, and power electronics.

Sakib Mahmud received his Bachelor of Science (BSc.) degree in Electrical Engineering with an Honours from Qatar University (QU) in June 2020 and currently pursuing his Master of Science (MSc.) in Electrical Engineering from the same department. As an undergraduate student, he received the Dean's Award for six semesters during 2016-2019. As an undergraduate student at QU, He has been a part of two UREP projects funded by the QNRF. Upon graduation, he joined QU's master's program to pursue a postgraduate degree in electrical engineering and work as a Graduate Research Assistant (GRA) at the same time, under the same department. As a GRA, and a member of QU Machine Learning Group, he was hired in two QNRF funder NPRP grants and three High Impact grant from QU. Currently he is hired in a High Impact grant funded by QU. Apart from his research work, he has been an Electrical Engineering Intern in the Electrical Consultant Group (E.C.G.) Qatar Branch, Data Science Intern in Data Glacier and has been selected as an Artificial Intelligence (AI) Intern in AI Ready Academy, a joint training program for

Machine Learning and Data Science by Microsoft and ZAKA, among 2000 competitive applicants around the Middle East and North Africa (MENA) region, He has expertise in Electronics Circuitry design, 3D Modeling, Biomedical Signal and Image Processing, Computer Vision, Machine and Deep Learning, and Data Analytics and Visualization in various platforms. Currently, he has published 10 articles in peer reviewed Journals, one conference paper and one patent.

Amith Khandakar received the B.Sc. degree in electronics and telecommunication engineering from North South University, Bangladesh, and the master's degree in computing (networking concentration) from Qatar University, in 2014. He graduated as the Valedictorian (President Gold Medal Recipient) of North South University. He is an IEEE Senior Member. He is also a certified Project Management Professional and the Cisco Certified Network Administrator. He has 2 patents and published around 60 peer-reviewed journal articles, conference papers, and four book chapters. His current research interests include biomedical instrumentation, wearable sensors, medical image analysis, machine learning and Engineering Education. He is also running UREP grants from QNRF and internal grants from Qatar University.

STOCK PRICE PREDICTION MODEL BASED ON DUAL ATTENTION AND TCN

Yifeng Fu and He Xiao

Department of Software Engineering,
Jiangxi University of Science and Technology, Nanchang, China

ABSTRACT

The stock market is affected by many variables and factors, and the current forecasting models for time series are often difficult to capture the complex laws among multiple factors. Aiming at this problem, a stock price prediction model based on dual attention mechanism and temporal convolutional network is proposed. First, a convolution network more suitable for time series is used as the feature extraction layer. Feature attention is introduced to dynamically mine the potential correlation between the input factor features and closing prices. Second, based on Gated Recurrent Unit, on the other hand, a temporal attention mechanism is introduced to improve the model's ability to learn important time points and obtain importance measures from a temporal perspective. The experimental results show that the proposed model performs better than the traditional prediction model in the error index of stock price prediction and realizes the interpretability of the model in terms of index characteristics and time.

KEYWORDS

Time convolutional network, GRU, Temporary attention, Feature attention, Interpretability .

1. INTRODUCTION

Stock price forecasting refers to analyzing stock-related data and making predictions about the subsequent trend or fluctuation of the stock. The forecast results of stock prices can play a certain reference and guidance role for investors. However, because the time series data of the stock market have the characteristics of non-linearity, non-stationarity and high complexity, the prediction effect is not ideal.

We present the history of the development and some research results in the field of stock forecasting in Section 2, followed by the model structure and its definition in Section 3, the process of constructing the model, the experimental results and the analysis of the results in Sections 4 and 5. Finally, the experimental results are summarized in Section 6.

2. RELATED WORK

In the field of studying stock trends, early approaches mainly used machine learning algorithmic models to analyze based on numerical values. Wen Fenghua et al [1] proposed a method using a combination of SSA and SVM for prediction. They use singular spectrum analysis to decompose stock prices into trends, market fluctuations, and noise with different economic characteristics over different time horizons, and then introduce these characteristics into a support vector machine for price prediction [2].

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 99-114, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122007

It can be found that in early research, deep learning is not commonly used in the field of stock prediction, but due to its powerful learning ability, more and more researchers find it very suitable for learning complex data such as stock prediction, traffic prediction, and pedestrian flow prediction. Khaled Althelaya evaluates and compares LSTM deep learning architectures for short and long term forecasting of financial time series. They considered bidirectional and stacked LSTM prediction models in their experiments and benchmarked them with shallow neural networks and simple forms of LSTM networks [3][4].

As the study progressed, researchers found that most research methods using single-factor data had poor predictive effects because the stock trading data itself had limited information [5][6]. Jian Wang used the moving average convergence/divergence and the improved historical volatility index as the evaluation indicators for buying or selling, and constructed a forecast based on the improved MACD indicator[7].The results show that the model based on the index after the introduction of the panic index, the buy and sell operations made more profit[8][9].

Among the existing methods, the traditional RNN suffers from gradient explosion, and CNN is also considered not suitable for time series processing. To address this issue, this paper proposes a combined model based on Feature Attention TCN and Temporal Attention GRU.

The main features of the model are as follows.

1. In terms of data processing, a multi-source data fusion approach is used to fuse the numerical features of stock prices and the features of stock news, considering the influence of public opinion on stock prediction.
2. Since the neurons in RNN are invariant to replication, it brings the problem of gradient disappearance in training. We try to use CNN to process time series, but CNN is not suitable for time series learning. Therefore, this paper differs from the traditional use of RNN to process time series data by using a one-dimensional temporal convolution network to process the extracted features[10].
3. Based on TCN and GRU, a dual focus mechanism of features and time series is introduced to capture potential connections between different stock indicators and prediction targets for the relevant indicators of stock series. In terms of time nodes, the expression capability of output targets at key time points is enhanced.

3. MODEL DEFINITION

3.1. TCN Model

In general, CNNs are not considered suitable for solving time series problems. However, TCN, as a special kind of convolutional neural network, is more suitable for processing time series data. In this paper [11], the authors compare the performance of LSTM, GRU, RNN, and TCN using various sequence modeling tasks. In the sequence task on the MNIST dataset, TCN achieves an accuracy of 99%, at least 3 percentage points higher than the other models. There are three reasons for this: the existence of causality in the convolution in this architecture means that there is no information leakage from future to the past. Secondly, the convolutional architecture can map arbitrary length sequences to fixed length sequences. In addition, it uses residual modules and dilated convolution to construct long-term dependencies. In terms of performance comparison, TCN can parallelize the time series as vectors, which is faster than the point-by-point sequential computation of RNN. In addition, TCN can extend the input into a one-dimensional sequence, thus avoiding the need for time-point-by-time alignment of features [12].

The network structure of TCN consists of three main parts: causal convolution, dilated convolution and residual module.

3.1.1. Causal Convolution

Causal convolution is a strictly time-constrained structure. Each hidden layer has the same length as the input layer, and is padded with zeros to ensure subsequent layers have the same length. For the value of the previous layer at time t, it only depends on the value of the next layer at time t and its previous value. The causal convolution structure is shown in Figure 1.

$$F(t) = \left(x_{(*d)}f\right) = \sum_{i=0}^{k-1} f(i) \cdot x_{(t-di)} \quad (1)$$

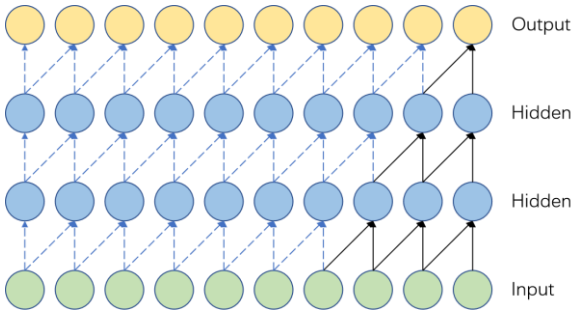


Figure 1. Causal Convolution

3.1.2. Dilated Convolution

There are still some problems with pure causal convolution, such as difficulty in capturing the dependencies between longer interval time points. Dilated convolution allows spaced sampling of the convolution time point input, and the sampling rate is controlled by the parameter Dilate [13]. The higher the level, the larger the Dilate used. Therefore, dilated convolution can expand the effective window, so that a larger receptive field can be obtained with a smaller number of layers.The dilate convolution structure is shown in Figure 2 and 3.

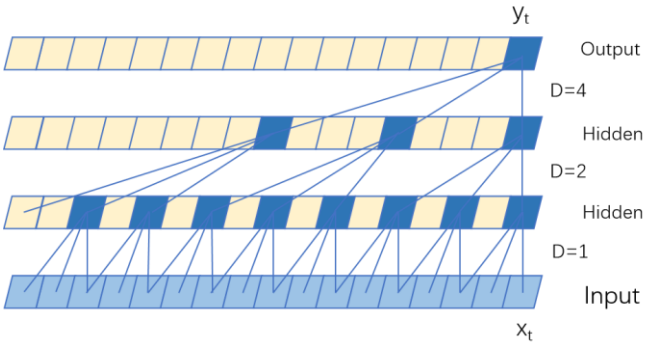


Figure 2. Dilated convolutional layer

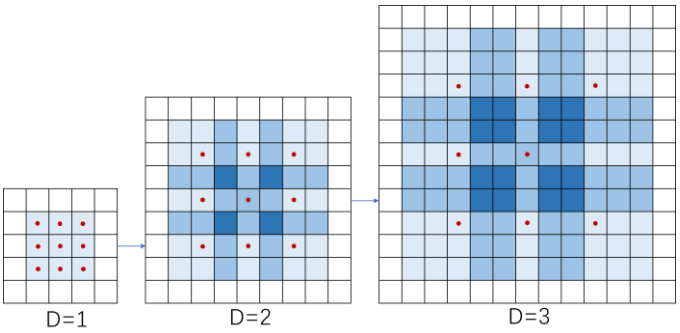


Figure 3. Feeling field for expanding convolution

3.1.3. Residual Module

The residual module of TCN consists of two layers of dilated convolution and ReLU function. The layer-hopping connection directly connects the feature maps of the lower layer to the upper layer[14]. The residual module ensures that each layer learns the relationship between mappings efficiently, which is very effective in networks with deeper layers[15]. As a result, the gradients of TCN are more stable and the problem of exploding or disappearing gradients can be effectively avoided. The structure of te residual block is shown in Figure 4.

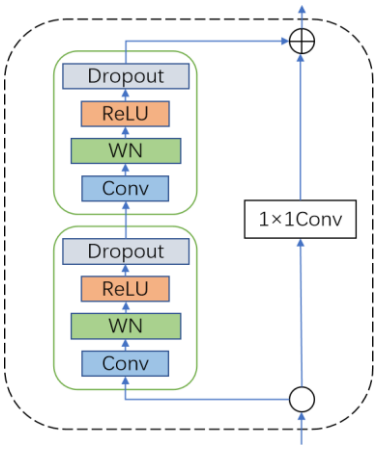


Figure 4. Residual Block

3.2. FATCN-TAGRU Model

By introducing the feature attention mechanism into the feature extraction process of the TCN model, the structure of feature attention is shown in Figure 5. Suppose the input time series is $X=[x_1,x_2,\cdots,x_T]=[x^{(1)},x^{(2)},x^{(3)},\cdots,x^{(6)}]^T$, The expansion can be represented as the following matrix:

$$X=\begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(6)} \\ x_2^{(1)} & x_2^{(2)} & \cdots & x_2^{(6)} \\ \vdots & \vdots & & \vdots \\ x_T^{(1)} & x_T^{(2)} & \cdots & x_T^{(6)} \end{bmatrix} \in R^{T\times 6} \quad (2)$$

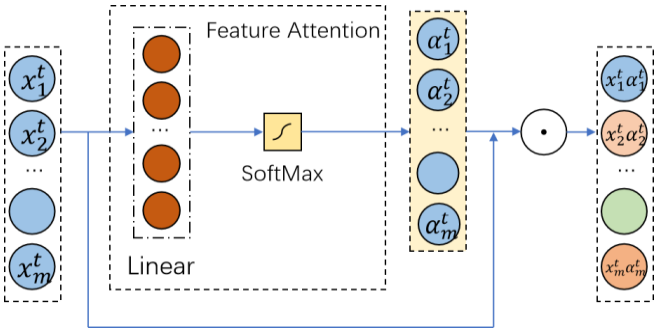


Figure 5. Feature Attention Model

In formula 2, $x_t = \left[x_t^{(1)}, x_t^{(2)}, \cdots, x_t^{(6)}\right](1 \le t \le T)$ is the set of 6 features at time t. $x^{(m)} = \left[x_1^{(m)}, x_2^{(m)}, \cdots, x_T^{(m)}\right](1 \le m \le 6)$ is the value of the m-th stock price-related variable at time t. Using a single-layer neural network to calculate the attention weight vector e_t, The formula is as follows:

$$e_t = tanh\left(W_e x_t + b_e\right) \qquad (3)$$

In formula 3, $e_t = \left[e_{1,t}, e_{2,t}, \cdots, e_{M,t}\right]$ is the attention weight coefficient combination corresponding to each input feature at the current time t; W_e is the training weight matrix; b_e is the bias vector for calculating the feature attention weight [16]. Since feature attention is located in the shallow layer of the model, and the input feature data is usually concentrated in a certain numerical range[17].

The structure of the time series attention mechanism is shown in Figure 6. The input $h_t = \left[h_{1,t}, h_{2,t}, \cdots, h_{k,t}\right]$ is the hidden layer state of the GRU network from the model iteration to time t, where k is the length of the input sequence time window. The time series attention weight vector l_t corresponding to each historical moment at the current time t is:

$$l_t = tanh\left(W_d h_t + b_d\right) \qquad (4)$$

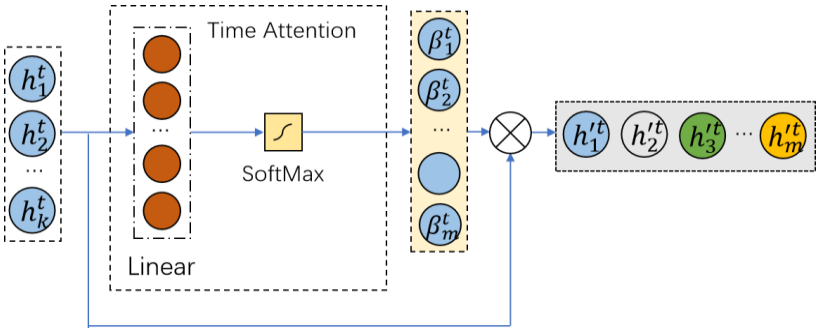


Figure 6. Time Attention Model

In the formula 4; W_d is the trainable weight matrix; b_d is the bias vector for calculating the temporal attention weight. The time attention weight $\beta_t = [\beta_{1,t}, \beta_{2,t}, \cdots, \beta_{\tau,t}, \cdots, \beta_{k,t}]$ is obtained, where $\beta_{\tau,t}$ is the attention weight at the τ th moment, which is weighted with the hidden layer state of each corresponding historical moment to obtain a comprehensive timing state $h_t^{'}$.

$$\beta_{\tau,t} = \frac{\exp(l_{\tau,t})}{\sum_{j=1}^{k} l_{j,t}} \quad (5)$$

$$h_t^{'} = \beta_t \otimes h_t = \sum_{\tau=1}^{k} \beta_{\tau,t} h_{\tau,t} \quad (6)$$

The FATCN-ATGRU network model is constructed by introducing the above feature attention and time series attention into the TCN layer and the GRU layer, respectively. The structure is shown in Figure 7. First, build the TAGRU network structure [18]. Compared with the TCN module, because the normalization part is already included in the attention mechanism, the Weight norm layer and the ReLU layer in the residual block are deleted. Use FATCN to mine the potential relationship between the input features and obtain the weighted input sequence; then extract the hidden time sequence correlation information from the weighted input feature sequence at the ATGRU layer, and mine the relevant feature time sequence information and current time data through the time sequence attention layer [19]. And assign time attention weights to it to enhance the expressive ability of key historical moment information, obtain weighted comprehensive time series information status, and finally send it to the fully connected layer to output future closing price predictions [20].

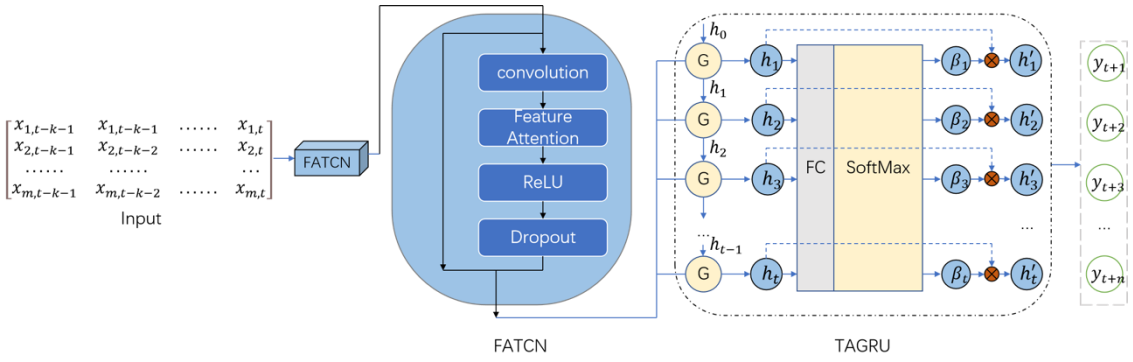


Figure 7. Model Structure

4. EXPERIMENTAL MODELING AND PROCESS

4.1. Data Collection and Pre-Processing

The stock data was downloaded from Yahoo Finance website. The time span is from June 26, 2017 to December 31, 2021. Stock market data for Apple, Google, Tesla, and Amazon are included. News text data was obtained by crawling the data of individual stock research reports from Eastern Fortune Website.

The downloaded stock numerical data includes six characteristic attributes: opening price, closing price, reweighted closing price, high price, low price, and trading volume. In this paper, closing price is used as the prediction label. News text features are obtained by processing research reports. Sentiment analysis is performed on the text data by using sentiment analysis API, and the processed sentiment indices are fused with the corresponding stock data as sentiment indices. The final fused input data has 7 feature attributes.

Data pre-processing consists of two specific steps.

1. Data cleaning: check the missing values and clean off the missing values, such as a day's data is missing a certain indicator, the data of that day will be deleted.
2. Normalization processing: This paper uses the MinMaxScaler normalization method, which is a linear transformation of the original data to eliminate the effects of differences between large and small units. The method deflates each feature to a given range, and after normalization, the indicators are in the same order of magnitude, which facilitates comprehensive comparison.

4.2. Model Parameter Settings

The initial time step size set in the experiment is 7, the learning rate is 0.01, and the parameters are updated with Adam optimizer and MSE loss function. Based on many previous studies and my replication and validation of experiments in the relevant literature, RNNs work best for time series tasks when the number of hidden layers is set to 2, and the number of neurons in each layer is 64. In addition, the effects of the number of residual blocks and the number of model iteration cycles in the TCN that need to be compared and analysed[21].

Table 1. Experimental results for some parameters Comparison.

Blocks	Epoch	R2	MAPE
2	500	0.981	0.0322
2	1000	0.974	0.0337
2	2000	0.983	0.0330
3	500	0.950	0.0338
3	2000	0.978	0.0378
4	500	0.974	0.0379
5	500	0.970	0.0382

It can be seen that when the residual block is set to 2 and the number of iterations is set to 500 or 2000, the prediction of the model is better. But when the number of iterations is set to 2000, the processing efficiency of the model is too low. So we choose 2 and 500.

5. EXPERIMENTAL RESULTS AND ANALYSIS

5.1. Evaluation Indicators

The evaluation indicators use RMSE, MAE, R2 Score, and MAPE. where N is the total number of samples, and the parameter i represents the i-th sample. $\overline{y_i}$ is the predicted value, $\overline{y_i'}$ is the actual value. Calculated as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_i')^2} \tag{7}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - y_i^{'}\right| \tag{8}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - y_i^{'})^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2} \tag{9}$$

$$MAPE = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{(y_i - y_i^{'})}{y_i}\right| \tag{10}$$

5.2. Comparative Analysis of Different Models

5.2.1. Single Model Comparison

In order to better compare and analyse the effect of the model, the traditional RNN and its variants and combined models are used for test comparison. The single model prediction effect is shown in Table 2 and Figure 8,9,10,11.

Table 2. Single Model Scoring Comparison.

Model	RMSE	MAE	R2	MAPE
RNN	0.0614	0.0380	0.8375	0.0598
LSTM	0.0484	0.0306	0.9161	0.0516
GRU	0.0479	0.0304	0.9141	0.0477
TAGRU	0.0365	0.0212	0.0934	0.0338

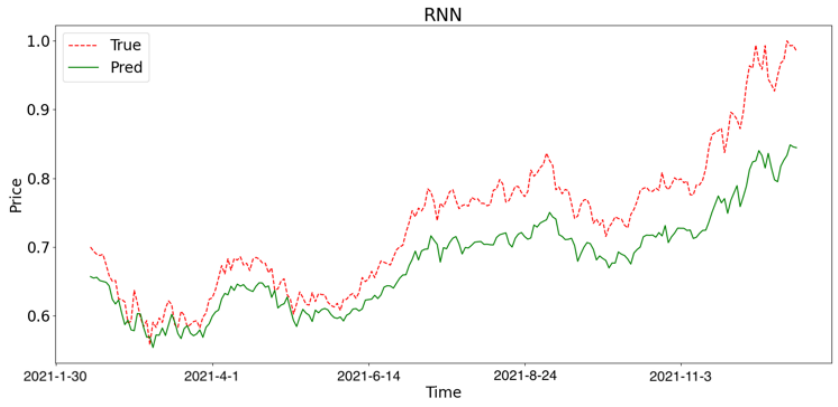


Figure 8. Prediction results of the RNN model

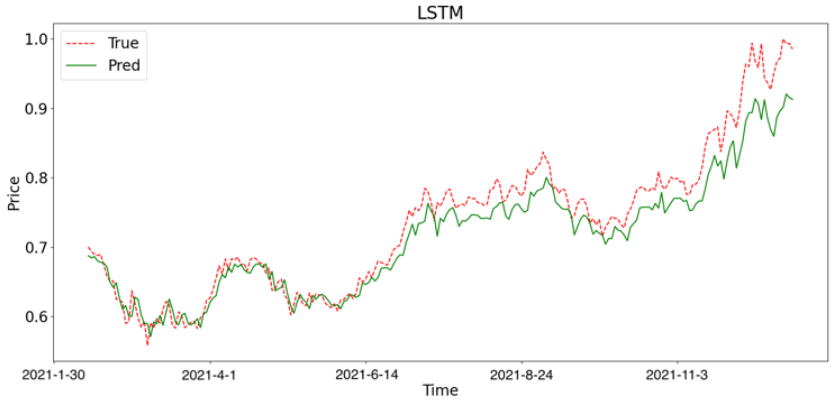


Figure 9. Prediction results of the LSTM model

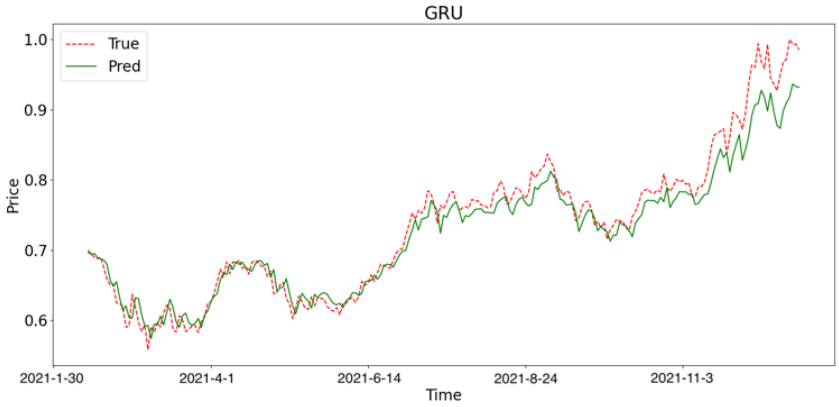


Figure 10. Prediction results of the GRU model

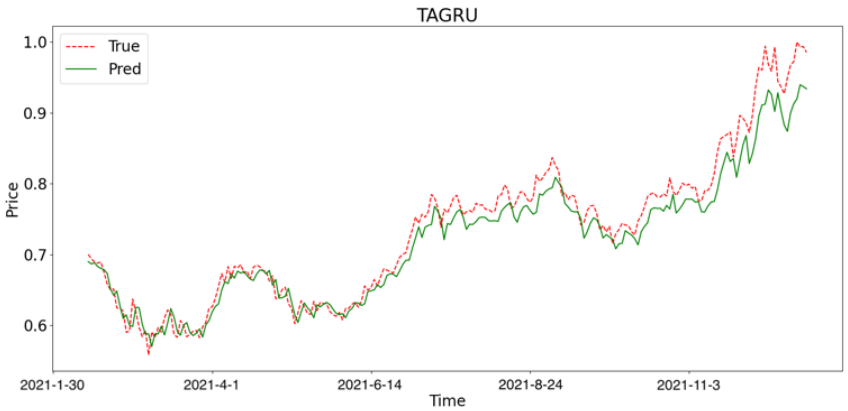


Figure 11. Prediction results of the TAGRU model

It can be found that the experimental results without using TCN network are poor, the fitting degree does not exceed 94 precent, and the predicted results have a certain lag phenomenon. The reason is that a model with a gated structure has the ability to retain memory to capture long-term relationships, and has a mechanism to reduce vanishing gradients, while a simple RNN erases and rewrites the entire memory at each update. Between LSTM and GRU, the gap between the

two is small, and the fitting effect of GRU is only 0.2 precent better than that of the LSTM network. The reason is that GRU has the advantage of simple structure and less computational effort than LSTM. The above experimental results illustrate that the attention mechanism weights can help the model to better capture the complex patterns between different time points and help to deal with irregular time series.

5.2.2. Combination Model Comparison

The prediction results of introducing the FATCN module are shown in Figure 12,13,14,15. By comparison, the model with the introduction of FATCN performs better in each error index, indicating that the features processed by TCN have a better effect in prediction. By comparing the prediction effect with or without introducing attention, it can be found that the model predicts better by introducing the attention mechanism. The reason is that different time points and different features in the stock data have different effects on the predicted time points. The stock data closer to the prediction point has a greater impact on the prediction point.

Table 3. Combination Model Results Comparison.

Model	RMSE	MAE	R2	MAPE
FATCN-RNN	0.0479	0.0269	0.9496	0.0357
FATCN-LSTM	0.0361	0.0274	0.9584	0.0478
FATCN-GRU	0.0300	0.0196	0.9840	0.0318
FATCN-TAGRU	0.0180	0.0129	0.9908	0.0222

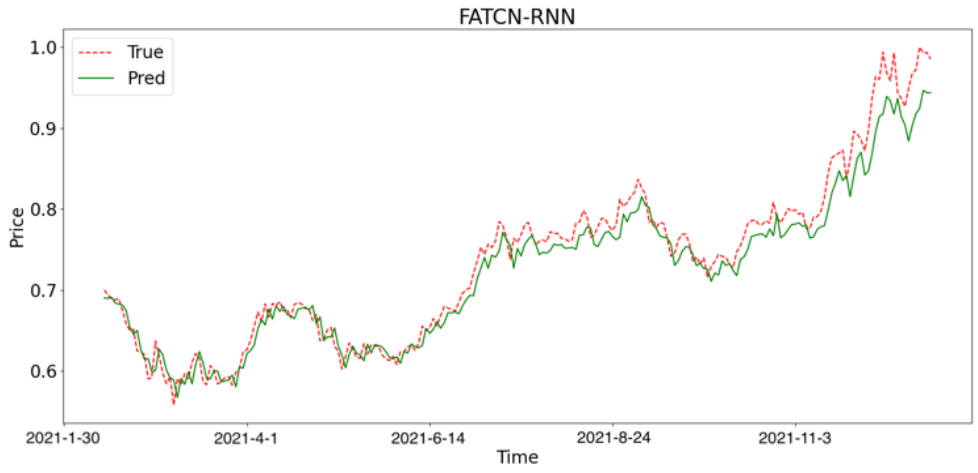


Figure 12. Prediction results of the FATCN-RNN model

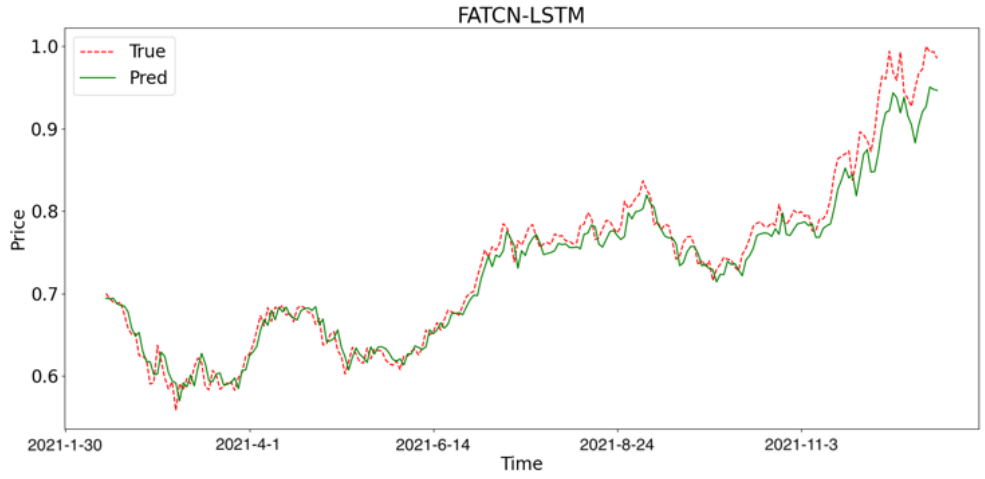


Figure 13. Prediction results of the FATCN-LSTM model

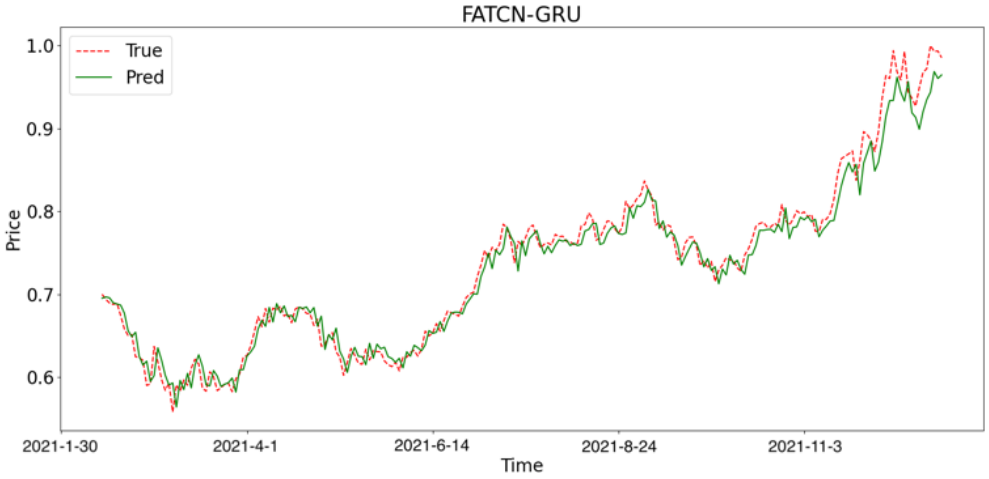


Figure 14. Prediction results of the FATCN-GRU model

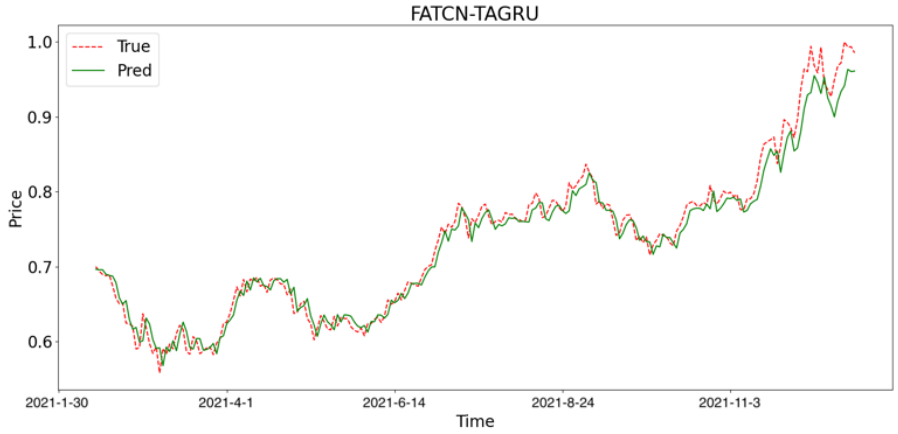


Figure 15. Prediction results of the FATCN-TAGRU model

5.3. Comparative Analysis of Different Stocks

In order to further examine the generalization ability of the model, three stocks of Apple, Tesla, and Amazon and their related news data are used to test the model.

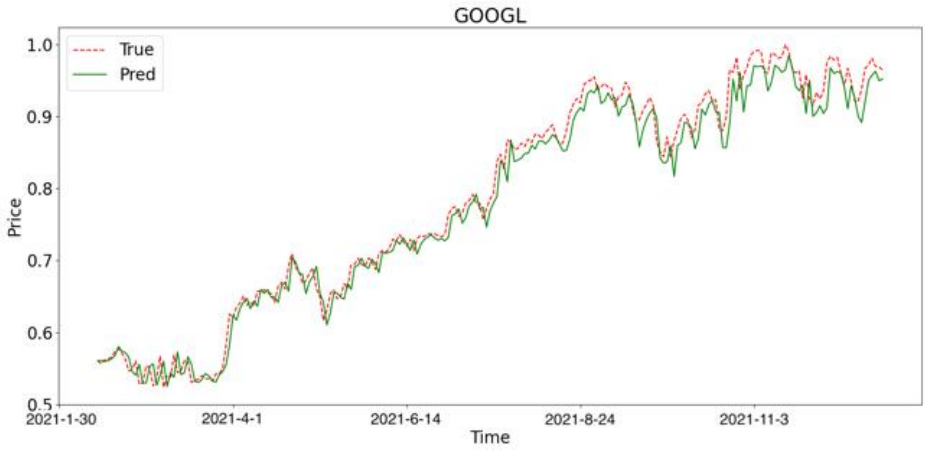


Figure 16. Results of an experiment using Google stock

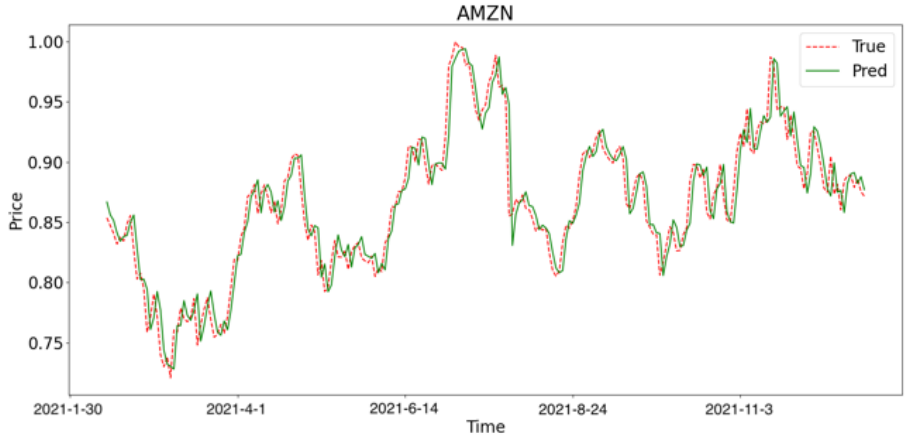


Figure 17. Results of an experiment using Amazon stock

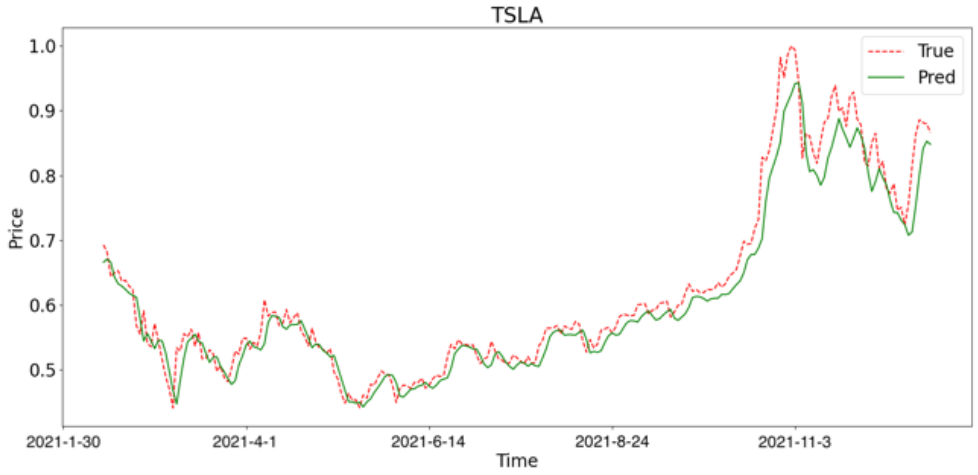


Figure 18. Results of an experiment using Tesla stock

It can be seen that after selecting the data of Apple, Amazon, and Tesla, the coefficient of determination of the model has reached more than 0.97. The three evaluation indicators also obtained good prediction results under other comparison models.

5.4. Interpretation Of Attention Weights

5.4.1. Feature Attention Weights Explained

Figure 19 shows a heatmap of the feature weights as the model converges. The vertical axis represents the characteristics of the stock data, and the horizontal axis represents the historical time point, in which time point 7 is the closest to the prediction time, and the depth of the color block represents the attention weight. In the convergence process, the weight of "closing price after resumption of rights" rises to 0.048 and the weight of "opening price" rises to 0.046, indicating that the closer to the prediction time point, the greater the contribution of the stock opening price and weighted closing price to the prediction result. The characteristics of "highest price", "minimum price" and "volume" have lower weights and have less influence on the prediction results. In the real stock trading market, the previous day's opening price and the reweighted closing price indicate the initial and final performance of the stock in the previous day's market, both of which directly affect the next day's movement [22]. The high and low prices represent only the two extreme points during the opening period and do not represent the final trend and performance of the stock. Therefore, the domain knowledge is consistent with the above experimental results and explains the rationality of the model from the perspective of feature importance.

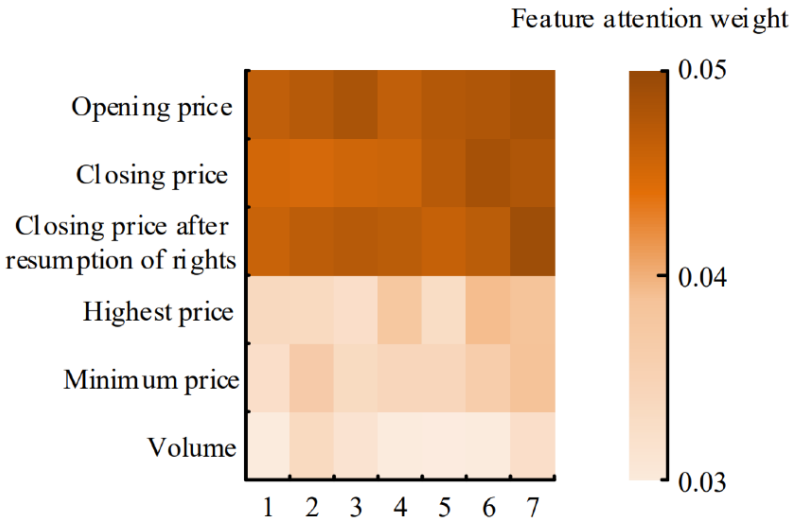


Figure 19. Feature Attention Weights

5.4.2. Time Attention Weights Explained

Figure 20 shows a sample of the temporal attention weight heat map. From the figure, it can be seen that the color blocks near the prediction time point are darker in color and have higher weights. It indicates that the model mainly focuses on the time steps closer to the prediction time. The analysis results in the above figure are consistent with the domain knowledge, which verifies the reasonableness of the prediction model in this paper from the perspective of temporal importance.

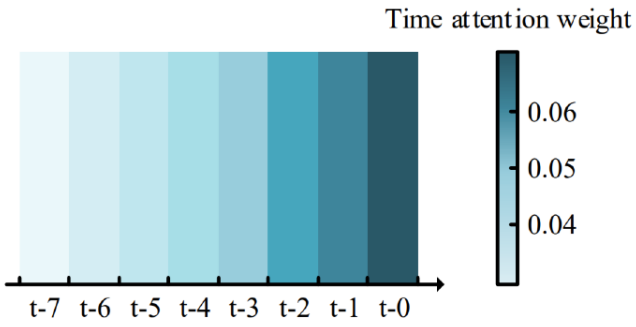


Figure 20. Time Attention Weights

6. CONCLUSION

In this work, we propose a TCN that introduces feature attention and a GRU framework that introduces time series attention. FATCN is responsible for processing time series data to extract deep features, and a feature attention mechanism is introduced to focus on important features, and then TAGRU is responsible for predicting frame. Finally, by comparing with other models and testing multiple stock data sets, better results are achieved. The results obtained by the model in this paper can provide a certain reference value for stock market investors on the microscopic level. On the macroscopic level, if risks can be predicted in advance, economic losses can be

avoided in advance. In the future, we will try to introduce more sentiment indicators and test the effect of the experiment using a two-way structure in the prediction model.

ACKNOWLEDGEMENTS

This work was partially funded by the Science and Technology Project of Jiangxi Provincial Education Department, grant number GJJ200824.

REFERENCES

[1] Y. Lin, H. Guo, and J. Hu, "An svm-based approach for stock market trend pre-diction," in The 2013 international joint conference on neural networks (IJCNN), pp. 1–7, IEEE, 2013

[2] W. Fenghua, X. Jihong, H. Zhifang, and G. Xu, "Stock price prediction based on ssa and svm," Procedia Computer Science, vol. 31, pp. 625–631, 2014.

[3] K. A. Althelaya, E.-S. M. El-Alfy, and S. Mohammed, "Evaluation of bidirectional lstm for short-and long-term stock market prediction," in 2018 9th International Conference on Information and Communication Systems (ICICS), pp. 151–156, 2018.

[4] A. Sethia and P. Raut, "Application of lstm, gru and ica for stock price prediction," in Information and communication technology for intelligent systems, pp. 479–487, Springer, 2019.

[5] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," Journal of computational science, vol. 2, no. 1, pp. 1–8, 2011.

[6] T. H. Nguyen, K. Shirai, and J. Velcin, "Sentiment analysis on social media for stock movement prediction," Expert Systems with Applications, vol. 42, no. 24, pp. 9603–9611, 2015.

[7] J. Wang and J. Kim, "Predicting stock price trend using macd optimized by historical volatility," Mathematical Problems in Engineering, vol. 2018, 2018.

[8] Y. Xu and S. B. Cohen, "Stock movement prediction from tweets and historical prices," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1970–1979, 2018.

[9] H. Kutlu and E. Avcı, "A novel method for classifying liver and brain tumors using convolutional neural networks, discrete wavelet transform and long short-term memory networks," Sensors, vol. 19, no. 9, p. 1992, 2019.

[10] H. Liu and Z. Long, "An improved deep learning model for predicting stock market price time series," Digital Signal Processing, vol. 102, p. 102741, 2020.

[11] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," arXiv preprint arXiv:1803.01271, 2018.

[12] L. Gong, M. Yu, S. Jiang, V. Cutsuridis, and S. Pearson, "Deep learning based prediction on greenhouse crop yield combined tcn and rnn," Sensors, vol. 21, no. 13, p. 4537, 2021.

[13] C. Yang, J. Zhai, and G. Tao, "Deep learning for price movement prediction using convolutional neural network and long short-term memory," Mathematical Problems in Engineering, vol. 2020, 2020.

[14] H. Wang, J. Wang, L. Cao, Y. Li, Q. Sun, and J. Wang, "A stock closing price prediction model based on cnn-bislstm," Complexity, vol. 2021, 2021.

[15] F. Xu and S. Tan, "Deep learning with multiple scale attention and direction regularization for asset price prediction," Expert Systems with Applications, vol. 186, p. 115796, 2021.

[16] J. Fan, K. Zhang, Y. Huang, Y. Zhu, and B. Chen, "Parallel spatio-temporal attention-based tcn for multivariate time series prediction," Neural Computing and Applications, pp. 1–10, 2021.

[17] J. Qiu, B. Wang, and C. Zhou, "Forecasting stock prices with long-short term memory neural network based on attention mechanism," PloS one, vol. 15, no. 1, p. e0227222, 2020.

[18] P. Hewage, A. Behera, M. Trovati, E. Pereira, M. Ghahremani, F. Palmieri, and Y. Liu, "Temporal convolutional neural (tcn) network for an effective weather forecasting using time-series data from the local weather station," Soft Computing, vol. 24, no. 21, pp. 16453–16482, 2020.

[19] D. Li, C. Lin, W. Gao, Z. Chen, Z. Wang, and G. Liu, "Capsules tcn network for urban computing and intelligence in urban traffic prediction," Wireless Communications and Mobile Computing, vol. 2020, 2020.

[20] Y. Wang, J. Chen, X. Chen, X. Zeng, Y. Kong, S. Sun, Y. Guo, and Y. Liu, "Short-term load forecasting for industrial customers based on tcn-lightgbm," IEEE Transactions on Power Systems, vol. 36, no. 3, pp. 1984–1997, 2020.
[21] H. Li, Y. Shen, and Y. Zhu, "Stock price prediction using attention-based multi input lstm," in Asian conference on machine learning, pp. 454–469, PMLR, 2018.
[22] L.-C. Cheng, Y.-H. Huang, and M.-E. Wu, "Applied attention-based lstm neural networks in stock prediction," in 2018 IEEE International Conference on Big Data (Big Data), pp. 4716–4718, IEEE, 2018.

AUTHOR

Fu Yifeng (1997 -), male, from Nanchang City, Jiangxi Province, China, is studying for a master's degree in computer technology at Jiangxi University of Technology. His main research fields are in-depth learning and quantitative trading.

LAYER-WISE RELEVANCE PROPAGATION FOR ECHO STATE NETWORKS APPLIED TO EARTH SYSTEM VARIABILITY

Marco Landt-Hayen[1], Peer Kröger[2], Martin Claus[2] and Willi Rath[1]

[1]GEOMAR Helmholtz Centre for Ocean Research, Kiel, Germany
[2]Christian-Albrechts-Universität, Kiel, Germany

ABSTRACT

Artificial neural networks (ANNs) are powerful methods for many hard problems (e.g. image classification or time series prediction). However, these models are often difficult to interpret. Layer-wise relevance propagation (LRP) is a widely used technique to understand how ANN models come to their conclusion and to understand what a model has learned. Here, we focus on Echo State Networks (ESNs) as a certain type of recurrent neural networks. ESNs are easy to train and only require a small number of trainable parameters. We show how LRP can be applied to ESNs to open the black-box. We also show an efficient way of how ESNs can be used for image classification: Our ESN model serves as a detector for El Niño Southern Oscillation (ENSO) from sea surface temperature anomalies. ENSO is a well-known problem. Here, we use this problem to demonstrate how LRP can significantly enhance the explainablility of ESNs.

KEYWORDS

Reservoir Computing, Echo State Networks, Layer-wise Relevance Propagation, Explainable AI.

1. INTRODUCTION

Machine learning (ML) provides powerful techniques in the field of artificial intelligence (AI) to discover meaningful relationships in all kinds of data. Within machine learning, artificial neural networks (ANNs) in shallow and deep architectures are found to be promising and very versatile. While these models considerably push the state-of-the-art solutions of many hard problems, they tend to produce black-box results that are difficult to interpret even by ML experts. Consequently, the question of enhancing the explainability of complex models ("explainable AI" or "xAI") has gained a lot of attention in the AI/ML community and stimulated a large amount of fundamental research [1], [2].

In its basic form layers of perceptrons [3] are stacked on top of each other to create a multilayer perceptron (MLP) [4]. These models are usually trained using some form of stochastic gradient descent (SGD) [5]. The aim is to minimize some objective or loss function. More sophisticated architectures e.g. make use of convolutional neural networks (CNNs) [6] or long short term memory (LSTM) [7] units to have recurrence in time in so-called recurrent neural networks (RNNs).
In this paper, we focus on geospatial data, which typically feature non-linear relationships among observations. In this szenario, ANNs are good candidate models, since ANNs are capable of handling complex and non-linear relations by learning from data and training some adjustable

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 115-130, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122008

weights and biases [8]. In recent years these methods have been used in various ways on geospatial data [9], [10], [11].

The problem with using ANNs on data of the Earth system is that we often only have relatively short time series to predict on or a small number of events to learn from. Using sophisticated neural networks encounters a large number of trainable parameters and these models are prone to overfitting. This requires a lot of expertise and effort to train these models and prevent them from getting stuck in local minima of the objective function. Famous techniques are dropout, early stopping and regularization [12], [13], [14].

In this work we overcome these problems by using Echo State Networks (ESNs) [15]. ESNs are a certain type of RNNs and have been widely used for time series forecasting [16], [17]. In its basic form an ESN consists of an input and an output layer. In between we find a reservoir of sparsely connected units. Weights and biases connecting inputs to reservoir units and internal reservoir weights and biases are randomly initialized. The input length determines the number of recurrent time steps inside the reservoir. We record the final reservoir states and only the output weights and bias are trained. But opposed to other types of neural networks, this does not encounter some gradient descent methods but is rather done in a closed-form manner by applying linear regression of final reservoir states onto desired target values to get the output weights and bias.
This makes ESN models extremely powerful since they require only a very small number of trainable parameters (the output weights and bias). In addition to that, training an ESN is easy, fast and leads to stable and reproducible results. This makes them especially suitable for applications in the domain of climate and ocean research.

But as long as ESNs remain black-boxes, there is only a low level of trust in the obtained results and using these kinds of models is likely to be rejected by domain experts. This can be overcome by adopting techniques from computer vision developed for image data to climate data. Layer-wise relevance propagation (LRP) is a technique to trace the final prediction of a multilayered neural network back through its layers until reaching the input space [18], [19]. When applied to image classification, this reveals valuable insights in which input pixels have the highest relevance for the model to come to its conclusion.

Toms, Barnes and Ebert-Uphoff have shown in their work [20] that LRP can be successfully applied to MLP used for classification of events related to some well-known Earth system variablity: El Niño Southern Oscillation (ENSO).

This work is inspired by [20] and goes beyond their studies: We also pick the well-known ENSO problem [21]. ENSO is found to have some strong zonal structure: It comes with anomalies in the sea surface temperature (SST) in Tropical Pacific. This phenomenon is limited to a quite narrow range of latitude and some extended region in terms of longitude. We use ESN models for image classification on SST anomaly fields. We then open the black-box and apply LRP to ESN models, which has not been done before - to the best of our knowledge.

SST anomaly fields used in this work are found to be noisy. For this reason, we focus on a special flavour of ESNs, that uses a leaky reservoir because they have been considered to be more powerful on noisy input data, compared to standard ESNs [22]. With the help of our LRP application to ESNs, we find the leak rate used in reservoir state transition to be a crucial parameter determining the memory of the reservoir. Leak rate needs to be chosen appropriately to enable ESN models to reach the desired high level of accuracy.

Our models yield competitive results compared to linear regression and MLP used as baselines. However, ESN models require significantly less parameters and hence prevent our model from

overfitting. We even find our reservoirs to be robust against random permutation of input fields, destroying the zonal structure in the underlying ENSO anomalies.

This opens the door to use ESNs on unsolved problems from the domain of climate and ocean science and apply further techniques of the toolbox of xAI [23].

The rest of this work is structured as follows: In Section 2 we briefly introduce basic ESNs and focus on reservoir state transition for leaky reservoirs. We then sketch an efficient way to use ESN models for image classification. Section 3 outlines the concept of LRP in general before we customize LRP for our base ESN models by unfolding the reservoir recurrence. The classification of ENSO patterns and the application of LRP to MLP and ESN models is presented in Section 4. Our models are not only found to be competitive classifiers but also reveal valuable insights in what the models have learned. We show robustness of our ESN model on randomly permuted input samples and visualize how the leak rate determines the reservoir memory. Discussion and conclusion are found in Section 5, followed by technical details on the used ESN and baseline models in the Appendix.

2. ECHO STATE NETWORKS

An ESN is a special type of RNNs and comes with a strong theoretical background [15], [24], [25]. ESN models have shown outstanding advantages over other types of RNNs that use gradient descent methods for training. We use in this work a shallow ESN architecture consisting of an input and output layer. In between we find a single reservoir of sparsely connected units. The weights connecting input layer and reservoir plus the input bias terms are randomly initialized and kept fixed afterwards. We find some recurrence within the reservoir and reservoir weights and biases are also randomly set and not trainable. Reservoir units are sparsely connected with sparsity usually in the range of 20-30%. Further constraints are put to the largest Eigenvalue of the reservoir weight matrix W_{res}. This is required for the reservoir to be stable and show the so-called Echo State Property [26].

Only the output weights and bias are trained by solving a linear regression problem of final reservoir states onto desired target outputs. A sketch of a base ESN model is shown in Figure 1.

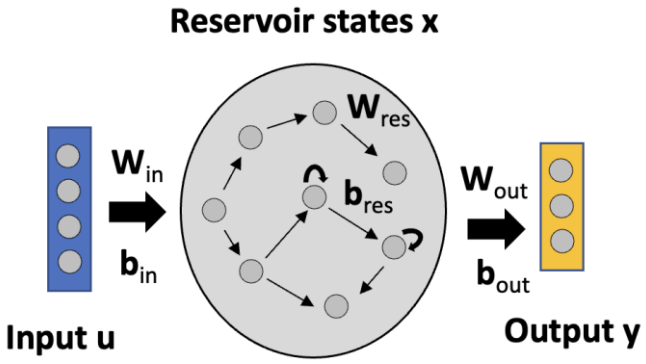


Figure 1. Sketch of base ESN: An input and an output layer, in between we find the reservoir.

In our ESN model, $u(t) \in \mathbb{R}^{D\times 1}$ denotes input values at time t with D input features. Inputs are fed into the model for T time steps, hence $t = 1..T$. Reservoir states at time $t = 1..T$ are denoted by $x(t) \in \mathbb{R}^{N\times 1}$, final reservoir states are obtained as $x(T)$. The final model output $y(T) \in \mathbb{R}^{M\times 1}$ at time T has M output values.

We then find input weights $W_{in} \in \mathbb{R}^{N \times D}$, connecting D input units to N reservoir units. Reservoir weights are given by $W_{res} \in \mathbb{R}^{N \times N}$ and output weights connecting N reservoir units to M output units read $W_{out} \in \mathbb{R}^{M \times N}$. In addition to weight matrices, we have bias vectors $b_{in} \in \mathbb{R}^{N \times 1}$, $b_{res} \in \mathbb{R}^{N \times 1}$ and $b_{out} \in \mathbb{R}^{M \times 1}$ for input, reservoir and output units, respectively.

We use a leaky reservoir with leak rate $\alpha \in [0, 1]$, as discussed in [22]. Leak rate serves as smoothing constant. The larger the leak rate, the faster reservoir states react to new inputs. In other words, the leak rate can be understood as the inverse of the memory time scale of the ESN: The larger the leak rate, the faster the reservoir forgets previous time steps' inputs. The reservoir state transition is defined by Equation 1.

$$x(t) = (1 - \alpha)\ x(t-1) + \alpha\ act[W_{in}u(t) + b_{in}\ + W_{res}x(t-1) + b_{res}] \qquad (1)$$

Here $act(.)$ is some activation function, e.g. *sigmoid* or *tanh*. From the initial reservoir states $x(t = 1)$ we can then obtain further states $x(t)$ for $t = 2..T$ by keeping a fraction $(1 - \alpha)$ of the previous reservoir state $x(t - 1)$. Current time step's input $W_{in}u(t) + b_{in}$as well as recurrence inside the reservoir $W_{res}x(t - 1) + b_{res}$ are added after applying some activation and multiplying with leak rate α. Reservoir states $x(t)$ are only defined for $t = 1..T$. This requires special treatment of $x(t = 1)$ as outlined in Equation 2.

$$x(t = 1)\ =\ \alpha\ act[W_{in}u(t) + b_{in}] \qquad (2)$$

The model output $y(T)$ is derived as linear combination of output weights W_{out} and biases b_{out} with final reservoir states $x(T)$, as shown in Equation 3.

$$y(T)\ =\ W_{out}\ x(T)\ +\ b_{out} \qquad (3)$$

This is a linear problem that can be solved in a closed-form manner with multi-linear regression minimizing mean squared error to obtain trained output weights and biases.

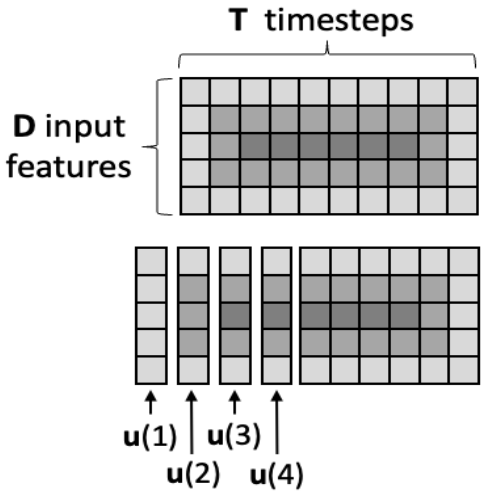


Figure 2. In the upper part we show a synthetic 2D input sample consisting of D input features and T time steps. Feeding the sample column by column into the base ESN model requires breaking the sample into columns. In the lower part we show inputs for the first four time steps.

ESN models have been widely used for time series forecasting [16], [17]. The idea is to feed a single signal or multiple time series of a specific length T into the model. In our work we want to

use 2D image data as input samples. This can be done in various ways. One possibility it to flatten 2D image data to obtain a one-dimensional vector and then couple each input to one reservoir unit in the first time step. Without adding additional inputs, the reservoir then swings for some time steps to unfold its dynamics [27]. In this approach the number of reservoir units is directly linked to the number of input units. For high dimensional input data reservoirs can hence become quite large. As mentioned above, we need to put some constraint on the largest Eigenvalue of the reservoir weight matrix W_{res} for stability reasons. Getting the largest Eigenvalue becomes computationally intensive for huge reservoirs and we therefore chose a different approach:

Here we transform images into a temporal signal. This is done by transforming one of the spatial dimensions (longitude) to a temporal one and passing 2D images column-wise into a base ESN model [28]. This is sketched in Figure 2.Feeding an image with dimensions $D \times T$ into a base ESN model is equivalent to having D input time series with length T. This allows using ESN models for image classification.

3. LAYER-WISE RELEVANCE PROPAGATION

LRP was first introduced by Bach et al. in 2015 [18]. LRP aims at understanding decisions of non-linear classifiers like ANNs. It can be used on classification and regression problems. This technique opens the black-box by visualizing the contributions of single input units to model predictions. Resulting relevance scores for an individual input sample can be presented as a heat map and give an intuitive understanding of which parts of the input sample have the highest relevance.

LRP has been successfully applied to various network architectures including MLP, CNN or LSTM models [20], [29]. But to the best of our knowledge, LRP has not been used for ESN models. In this section we will briefly repeat the general idea behind LRP before we customize this technique for using it on base ESN models.

3.1. General idea of LRP

LRP, as presented in [18], does not provide some closed-form solution but rather comes as a set of constraints. Used on image data it serves as a concept for achieving a pixel-wise decomposition of the final model output $y(T)$, as stated in Equation 4.

$$y(T) = \sum_n R_n^{(1)} \qquad (4)$$

The model output $y(T)$ is taken as the final or total relevance. The ultimate goal is to decompose the final relevance and find the contributions $R_n^{(1)}$, also referred to as relevance score of each of the n input pixels. Here superscript (1) refers to the first layer, which is the input layer.

To achieve that goal the relevance is traced back from the output layer all the way through lower layers until we finally reach the input layer. In addition to Equation 4 the second constraint is stated in Equation 5.

$$y(T) = \cdots = \sum_j R_j^{(l+1)} = \sum_i R_i^{(l)} = \cdots = \sum_n R_n^{(1)} \qquad (5)$$

This framework guarantees total relevance to be preserved in each layer. For calculating the relevance map for an individual input sample, the trained model weights and biases are fixed. We

then start with the model output as final relevance. A common approach for tracing relevance back through lower layers is by taking only positive contributions of pre-activations into account. This clearly satisfies constraints in Equations 4 and 5. An example is sketched in Figure 3.

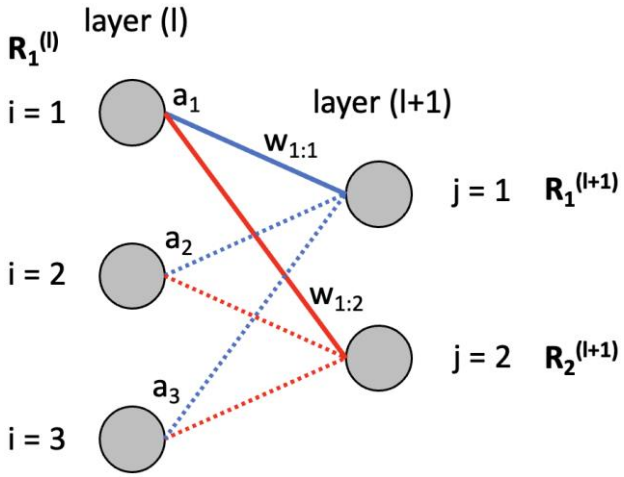


Figure 3. Illustrating the general idea behind LRP: Relevance is traced back from higher to lower layers.

Assume units $j = 1$ and $j = 2$ of layer $(l + 1)$ have known relevance scores $R_1^{(l+1)}$ and $R_2^{(l+1)}$, respectively. This relevance is now distributed on units $i = 1, 2, 3$ of layer (l). Unit $i = 1$ ends up having relevance $R_1^{(l)}$ from two contributions, as stated in Equation 6: One from unit $j = 1$ and one from unit $j = 2$ of layer $(l + 1)$, indicated by solid blue and red lines in Figure 3, respectively.

$$R_1^{(l)} = \left(\frac{a_1 w_{1:1}}{a_1 w_{1:1} + a_2 w_{2:1} + a_3 w_{3:1}}\right) R_1^{(l+1)} + \left(\frac{a_1 w_{1:2}}{a_1 w_{1:2} + a_2 w_{2:2} + a_3 w_{3:2}}\right) R_2^{(l+1)} \qquad (6)$$

Here a_1, a_2 and a_3 denote activations of units $i = 1,2,3$ of layer (l), respectively and $w_{i:j}$ denotes the weight connecting some unit i from layer (l) with some unit j from subsequent layer $(l + 1)$. This can be simplified using $z_{ij}^{+} = \max(a_i w_{i:j}, 0)$, where $+$ denotes that we only consider positive contributions. Relevance $R_{i=i_0}^{(l)}$ for a unit i_0 of layer (l) is stated in Equation 7.

$$R_{i=i_0}^{(l)} = \sum_j \frac{z_{i_0 j}^{+}}{\sum_i z_{ij}^{+}} R_j^{(l+1)} \qquad (7)$$

3.2. LRP customized for ESN models

Applying LRP to ESNs requires extending the basic methodology described in Section 3.1. Our ESN model consists of an input and an output layer. In between we have the reservoir with recurrence in time. Before we can apply LRP, we need to unfold the reservoir dynamics. Feeding an image consisting of T columns into a base ESN model leads to T time steps to be treated as individual layers. Accordingly, we have inputs $u(t) \in \mathbb{R}^{D \times 1}$ for time steps $t = 1..T$ with D input features.

In Section 2 we introduced our base ESN model including a leaky reservoir with leak rate α. Reservoir state transitions for time steps $t = 2..T$ have been stated in Equation 1. The initial reservoir states $x(1)$ are somewhat special, since there are no previous time step's reservoir states

$x(0)$, as we have seen in Equation 2. Figure 4 shows unfolded reservoir dynamics. In addition to that we decomposed reservoir state transitions to visualize distinct contributions separately. As soon as we have trained our base ESN model, the sketch in Figure 4 can be used to understand how the model output is calculated by forward passing an input sample through the network: In this example we have $D = 5$ input features in every time step (shaded yellow), denoted as $u(t)$ for $t = 1..T$. For simplicity we further assume to only have six reservoir units providing reservoir states $x(t)$ for $t = 1..T$ (shaded red). Reservoir states multiplied with $(1 - \alpha)$ contribute to subsequent time step's reservoir states, as sketched in the lower track of Figure 4. The second contribution is given by $\alpha\, act(.)$. Here $act(.)$ (shaded blue) is some appropriate activation function (e.g.*sigmoid* or *tanh*) and takes as argument the current time step's input $W_{in}u(t) + b_{in}$plus incorporates the recurrence inside the reservoir $W_{res}x(t-1) + b_{res}$. Once we calculated final reservoir states $x(T)$ we obtain model output $y(T)$ as seen in Equation 3 using trained output weights and bias.

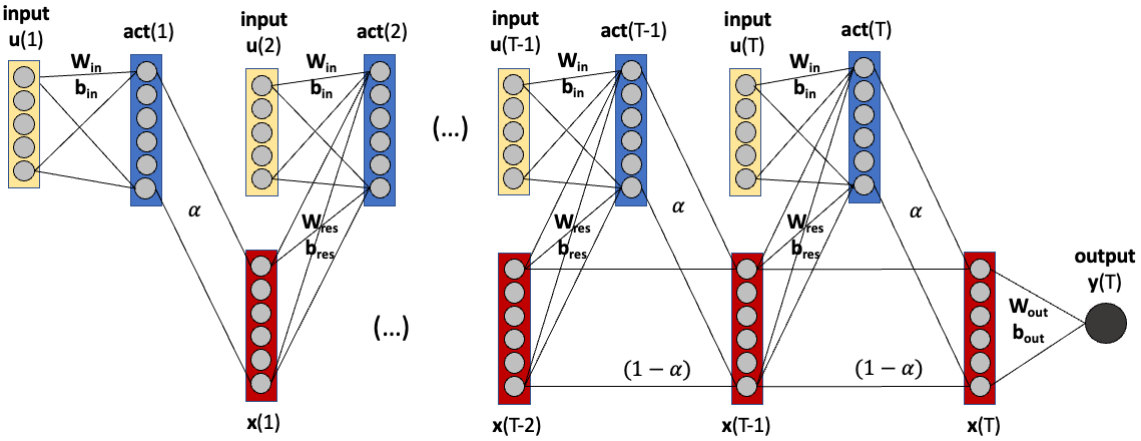


Figure 4. Unfolding our base ESN model in time.

But Figure 4 also illustrates how LRP works for our base ESN model. As usual, we pick an individual input sample and take the model output as final relevance. We then move backwards through all time steps. Opposed to the general concept of LRP, total relevance is not constant from time step to time step. Instead, a part of the total relevance is attributed to each time step's input $u(t)$ and only the remaining relevance is passed on until we reach the initial input $u(1)$. The initial input is special in a way, that it absorbs all residual relevance. For D input features we have $u(t) = (u_1(t), u_2(t), .., u_D(t))^T \in \mathbb{R}^{D\times 1}$ for each time step $t = 1..T$. And accordingly, we obtain relevance scores $R^{(t)} = (R_1^{(t)}, R_2^{(t)}, .., R_D^{(t)})^T \in \mathbb{R}^{D\times 1}$. These column vectors of relevance scores $R^{(t)}$ need to be combined to get the final relevance map $R \in \mathbb{R}^{D\times T}$, which can be visualized as heat map having the same dimensions as the input samples. Thus, total relevance is still preserved if we customize LRP to ESN models. However, Equations 4 and 5 need to be modified and can be combined to Equation 8.

$$y(T) = \sum_t R^{(t)} = \sum_t \sum_d R_d^{(t)} \qquad (8)$$

The final model output $y(T)$ is taken as total relevance and equals the sum of relevance scores $R_d^{(t)}$ with $t = 1..T$ and $d = 1..D$. But as mentioned above, the initial input $u(1)$ absorbs all residual relevance. The residual relevance itself depends on the amount of relevance, that has already been attributed to all other time steps' inputs $u(2), .., u(T)$. The speed of decay for total relevance, as it is passed through the layers in a descending order, is controlled by leak rate α. The role of α as memory parameter has been discussed in Section 2. If α is chosen too low, we

find an unreasonably high amount of residual relevance to be assigned to the initial input $u(1)$. To overcome this problem, we add a dummy column of ones as initial column to all input samples. This does not affect model performance, since the additional column is identical for all samples. The relevance $R^{(1)}$ attributed to the dummy column is meaningless in the overall relevance map R and can be omitted.

4. APPLICATION TO ENSO

In this section we will briefly recap the main characteristics of ENSO. Additional details on ENSO can be found e.g. in [20], [21]. We then show results from using MLP and our base ESN model for classifying 2D input samples and open the black-box by applying LRP as described in Sections 3.1. and 3.2. We intentionally choose ENSO as well-known problem to gain confidence in our model and methodology to open the door for applying LRP and further xAI techniques with ESN models on unsolved problems in the context of Earth system and climate research.

4.1. ENSO Patterns

For our studies we use measured monthly mean SST for the years 1880 through 2021, provided by US National Oceanic and Atmospheric Administration. Raw data comes in a 2° by 2° latitude-longitude grid. Each sample consists of 89×180 grid points.

There are several indices used to monitor the sea surface temperature in the Tropical Pacific. All of these indices are based on SST anomalies averaged across a given region. Usually, the anomalies are computed relative to a seasonal cycle estimated from some reference period (climatology) of 30 years (here 1980 through 2009). For our purpose we use SST anomalies averaged over the most commonly used Niño 3.4 region (5°N–5°S, 120–170°W), normalized by its standard deviation over the reference period to obtain a SST anomaly index used to define El Niño and La Niña events, which are associated with anomalous warm and cold SST, respectively. The index is shown in Figure 5.

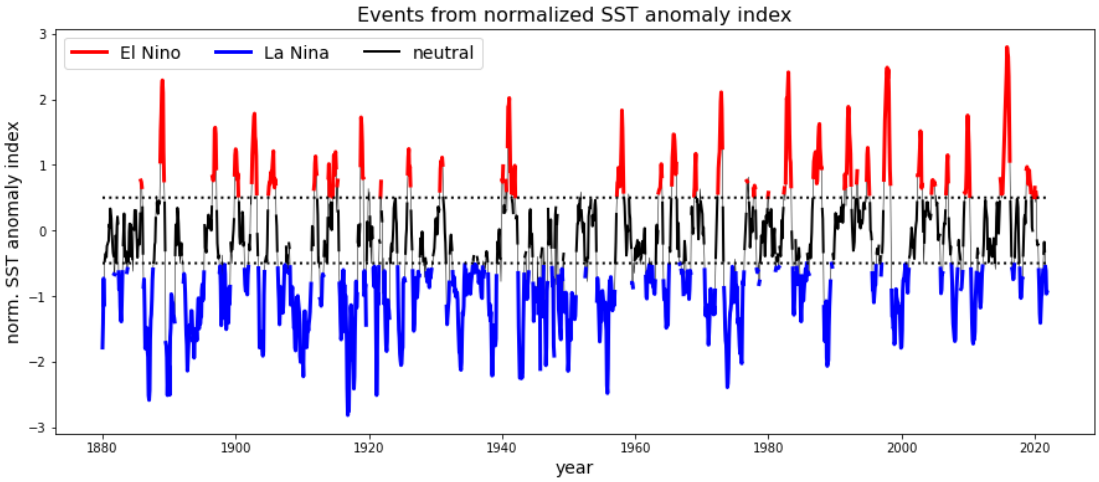


Figure 5. SST anomaly index used to define El Niño and La Niña events.

El Niño is referred to index values ≥ 0.5 (red), whereas La Niña events are referred to index values ≤ -0.5 (blue). In between we find *neutral* states, which are not considered here for classification. The SST anomaly index is used for labelling input samples and also as a single continuous target. In the time span from 1880 through 2021 we have a total number of 1,041

samples and split data into train and validation samples, using the first 80% for training (832 samples) and remaining 20% for validation (209 samples).Composite average SST anomaly patterns for El Niño and La Niña are shown in Figure 6.

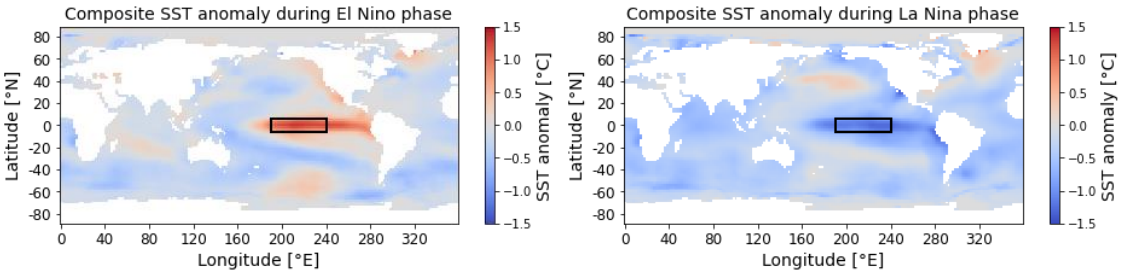


Figure 6. Composite average SST anomaly patterns for El Niño (left-hand side) and La Niña (right-hand side) events. Niño 3.4 region is highlighted by a black rectangle.

4.2. Classification and LRP

As described in Section 4.1 we train our models on 832 SST anomaly fields, where each input sample has dimensions 89×180 (latitude x longitude). SST is not defined over land masses. This reduces the number of valid grid points. Raw data shows some unreasonably high or low values: Here we limit SST anomalies to the range of $[-5°C, 5°C]$. Values exceeding these limits are set to upper and lower bound, respectively.

For our baseline models (linear regression and MLP) we vectorize valid grid points as inputs. SST anomalies are scaled to $[-1, 1]$. In any case we use the normalized SST anomaly index shown in Figure 5 as single continuous target. We then transform this regression problem to fit our classification problem by creating binary predictions from model output: Positive predictions refer to El Niño, whereas negative predictions refer to La El Niña events.

With this setup we easily reach 100% classification accuracy on both, El Niño and La Niña samples from train and validation data. This perfection was expected, as already shown in [20] and is due to the simplicity of the underlying problem.

For the base ESN model, we do not flatten input samples, as done for the linear regression and MLP approach. Instead, we feed 2D SST anomaly fields into our model and use longitude as time dimension. In other words, we have 89 input features, each consisting of 180 time steps. We deal with invalid grid points by setting SST anomalies to zero after scaling to inputs to $[-1, 1]$. Again, we use normalized SST anomaly index as single continuous target and create binary predictions from model output. Reservoir's leak rate is set to $\alpha = 0.01$.

This also leads to perfect accuracy on El Niño and La Niña, at least on train data. Validation accuracy is found to be 99% for both, El Niño and La Niña. We then focus on El Niño, for which we show the mean relevance maps obtained from MLP and our base ESN model, averaged over all train samples in Figure 7.

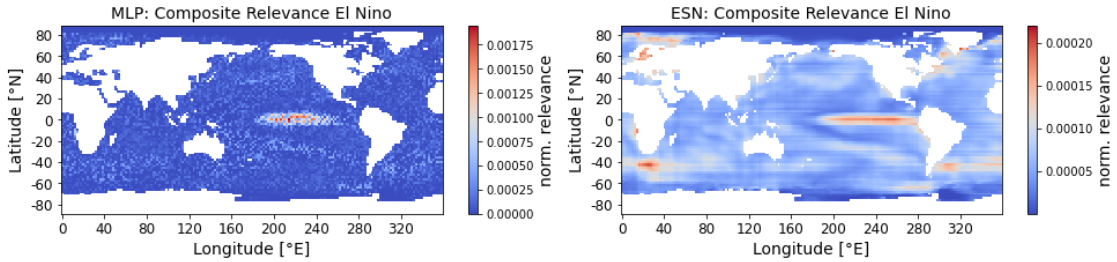


Figure 7. Mean relevance (normalized, unitless) obtained from LRP with MLP (left-hand side) and our base ESN model (right-hand side) on El Niño train samples.

We find the MLP to put its focus only on some narrow, elliptical region inside the Niño region in the Tropical Pacific. This was also found in [20] and appears to be reasonable and efficient to discriminate El Niño from La Niña samples. Compared to that, the mean relevance map obtained from our base ESN model also emphasizes the same spot to come to its conclusion. But in addition to that, we find significantly more structure in mean relevance highlighting other spots outside Niño region to be relevant. High relevance scores are attributed to the area between South Africa and Antarctica.

4.3. Random Permutation

ENSO patterns show some strong *zonal* structure: SST anomalies for both, El Niño and La Niña, are concentrated on some narrow range in latitude and some extended region in longitude. If we want to use our base ESN model to unknown problems, we need to make sure that this approach is also working without having such characteristic zonal structure present. To proof this, we apply some random (but reversible!) permutation on the columns of *all* input samples before training our model, to *shuffle* the order in time. The result is shown in Figure 8.

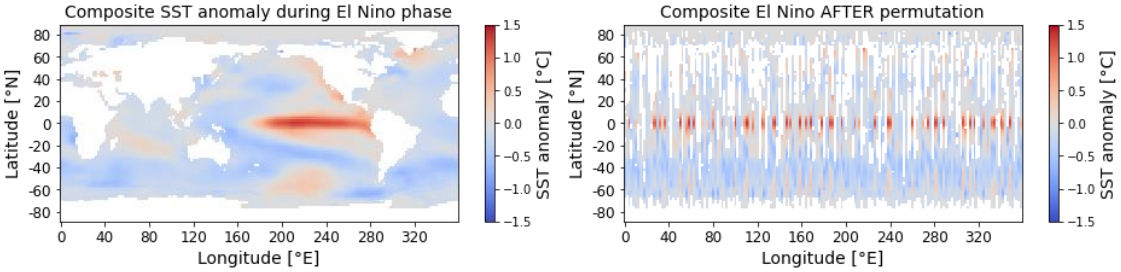


Figure 8. Composite average SST anomaly patterns for El Niño (left-hand side) and the same average SST anomaly AFTER some random permutation of columns (right-hand side).

We then train our base ESN model with unchanged parameters and apply LRP on permuted inputs. The obtained mean relevance map calculated on all El Niño train samples is shown in Figure 9. To restore some more familiar mean relevance map, the permutation needs to be reversed. The result is also shown in in Figure 9.

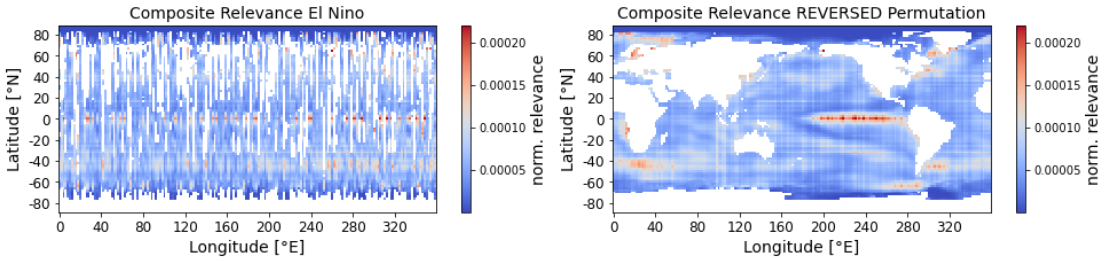


Figure 9. Mean relevance (normalized, unitless) obtained from LRP with base ESN model on PERMUTED El Niño train samples (left-hand side). And restored mean relevance after REVERSED permutation (right-hand side).

We find the *restored* mean relevance map to resemble the *original* mean relevance map, shown in Figure 7. This clearly proofs that our approach to pass 2D image data into base ESN models does not rely on the underlying structure in the input data. We also find the same accuracy for base ESN models trained *with* or *without* permuting input columns. This empowers Echo State Networks to be used on unknown problems in the context of climate and ocean science in combination with xAI techniques.

4.4. Fading Memory

In Section 2 we introduced the reservoir state transition as defined by Equation 1. Leak rate α is found to be a crucial parameter. It determines the memory of the reservoir and can be seen as the inverse of the memory time scale of the ESN: The larger the leak rate, the faster the reservoir forgets previous time steps' inputs. Here we use 2D input samples with $T = 180$ time steps for our base ESN model. In other words, we feed a 2D input sample column by column into the model, starting on the left-hand side. This procedure requires α to be chosen low enough to enable the reservoir to remember inputs from all time steps. This is especially important if we apply our method to unknown problems, since we do not know in advance which time steps are most relevant for achieving optimal performance.

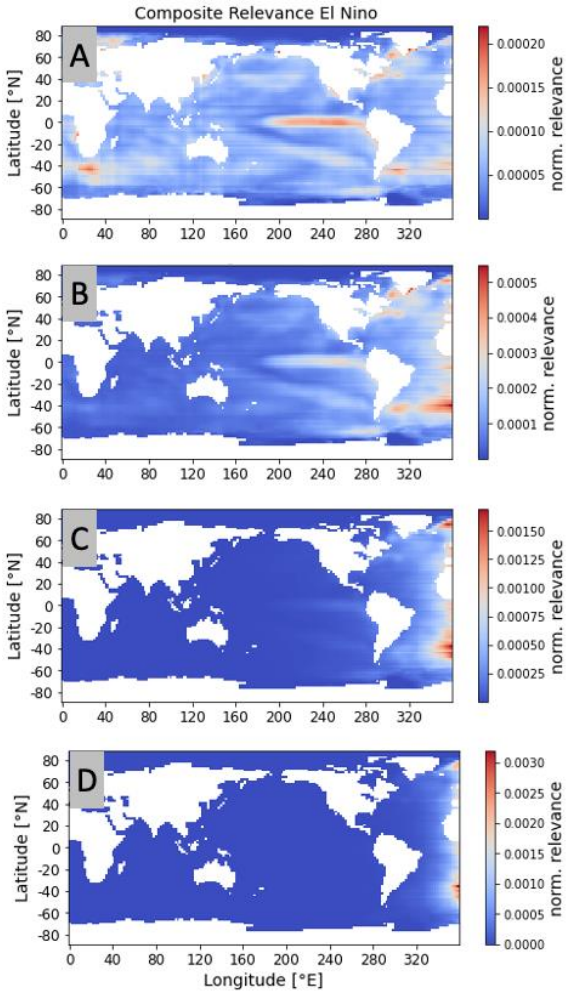


Figure 10. Fading memory effect: Mean relevance (normalized, unitless) obtained from LRP with base ESN model on El Niño train samples for α = 0.01 (A), 0.05 (B), 0.2 (C) and 0.4 (D), respectively.

With increasing leak rate, the reservoir memory fades. This effect is visualized in Figure 10. Here we show mean relevance maps for El Niño obtained from ESN models trained with four different leak rates $\alpha = 0.01, 0.05, 0.2$ and 0.4, respectively. For $\alpha = 0.01$ and 0.05 we find classification accuracy on train samples to be 100%, while validation accuracy reaches 99%. Accordingly, we observe high relevance in the Tropical Pacific region, as seen in relevance maps (A) and (B) in Figure 10. This appears to be reasonable for discriminating ENSO patterns. With further increasing$\alpha = 0.2$ and 0.4 the validation accuracy drops to 95% and 58%, respectively. Mean relevance maps (C) and (D) in Figure 10 explain this decline in model performance: The reservoir simply loses its memory of former input time steps and we find nonzero relevance concentrated on the right-hand side of the relevance maps, representing later time steps. For $\alpha = 0.4$ the model fails to distinguish between El Niño and La Niña samples. An accuracy of only 58% is close to random guessing.

5. DISCUSSION AND CONCLUSION

In this work we successfully used ESNs for image classification and applied LRP to this special type of RNNs, which has not been done before. This enabled us to look inside the model and understand, what the model has learned. LRP is a well-known approach and belongs to the xAI

toolbox. Using this technique on a reservoir with $T = 180$ time steps is challenging, but possible. Our proposed LRP customized for ESNs also empowers to study the effect of leak rate α. We found out that α needs to be chosen appropriately to allow the model to take inputs from all time steps into account.

Compared to MLP, mean relevance maps obtained from our ESN model reveal additional structure in terms of high relevance scores outside the Niño region. This needs to be further investigated and could point out existing teleconnections and help to find, where else ENSO leaves its footprint.

We find accuracy to be competitive compared to baseline models (linear regression and MLP). The advantage of ESNs is the low number of trainable parameters, which makes them fast and, thus, easy to train. In addition, our permutation experiments show, that ESN models yield reproducible and stable results. This even holds true if we only have limited train data, as often in the domain of Earth system and climate research. So, we can combine the advantages of ESN models with the power of the broad xAI toolbox. Further techniques to be applied to ESN models on similar problems may be backward optimization, sensitivity analysis or salience maps.

Beyond application to geospatial data, similar ESN models could be used for time series prediction: Instead of feeding 2D images into the model, we may pass a certain number of climate indices with specific input length to an ESN model and LRP could serve as an alternative for the temporal attention mechanism often used in the context of LSTM sequence-to-sequence models. In this way ESN models have good prospects to help understanding known teleconnections in atmospheric science or to find new relationships.

APPENDIX: MODEL DETAILS

In this section we briefly present some technical details on the multilayer perceptron used as baseline model and on our ESN model. The MLP was trained on vectorized SST anomaly fields, where we only considered valid grid points. In this case we worked with 10,988 input values for each sample. The input layer of the MLP consists of the same number of input units. We then have two hidden layers of 8 units each and finally one output unit. For a fully connected MLP we end up with 87,993 trainable weights and biases. We used a linear activation function (identity) for all layers and the Adam optimizer [30] with constant learning rate $lr = 0.0005$. The model was trained over 30 epochs with a batch size of 10. Since we have a regression problem using continuous SST anomaly index as single target, we took the mean squared difference of model output and ground truth as loss function, also referred to as mean squared error loss.

For our base ESN model, the number of reservoir units is set to $n_{res} = 300$. Input and reservoir weights and biases are drawn from a random uniform distribution in $[-0.1, 0.1]$. Reservoir units are only sparsely connected with $sparsity = 0.3$. After initialization the reservoir weights are normalized: The largest Eigenvalue of the reservoir weight matrix is set to 0.8. Leak rate is set to $\alpha = 0.01$. As activation in the reservoir state transition, we use tanh. With this setup our base ESN model only requires 300 trainable output weights plus one output bias, which is significantly less compared to 87,993 trainable parameters for the MLP model.

Raw data used in this work has been uploaded to Zenodo [31]. Annotated Python code can be found in our GitHub repository [32].

ACKNOWLEDGEMENTS

This work was supported by the Helmholtz School for Marine Data Science (MarDATA) funded by the Helmholtz Association (Grant HIDSS-0005).

REFERENCES

[1] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael A. Specter, and Lalana Kagal, "Explaining Explanations: An Approach to Evaluating Interpretability of Machine Learning," CoRR abs/1806.00069, 2018. http://arxiv.org/abs/1806.00069

[2] Marco T. Ribeiro, Sameer Singh, and Carlos Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," CoRR abs/1602.04938, 2016. http://arxiv.org/abs/1602.04938

[3] Frank Rosenblatt "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," Psychological Review vol. 65, pp. 386–408, 1958.

[4] Hassan Ramchoun, Mohammed Amine, Janati Idrissi, Youssef Ghanou, and Mohamed Ettaouil, "Multilayer Perceptron: Architecture Optimization and Training," International Journal of Interactive Multimedia and Artificial Inteligence, vol. 4, pp. 26–30, 2016.

[5] Sebastian Ruder, "An overview of gradient descent optimization algorithms," CoRR abs/1609.04747, 2016. http://arxiv.org/abs/1609.04747

[6] Keiron O'Shea and Ryan Nash, "An Introduction to Convolutional Neural Networks," CoRR abs/1511.08458, 2015. http://arxiv.org/abs/1511.08458

[7] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," Neural Computation vol. 9, no. 8, pp. 1735–1780, 1997.

[8] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville, "Deep Learning," MIT Press, Cambridge, MA, USA, 2016. http://www.deeplearningbook.org

[9] Ankur Mahesh, Maximilian Evans, Garima Jain, Climateai Mattias, Castillo Climateai, Aranildo Lima, Climateai Brent, Lunghino Climateai, Himanshu Gupta, Climateai Carlos, Gaitan Climateai, Jarrett Climateai, Omeed Tavasoli, Patrick Brown, and V Balaji, "Forecasting El Niño with Convolutional and Recurrent Neural Networks," In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada, pp. 8–14, 2019.

[10] Arvind W. Kiwelekar, Geetanjali S. Mahamunkar, Laxman D. Netak, and Valmik B.Nikam, "Deep Learning Techniques for Geospatial Data Analysis," CoRR abs/2008.13146, 2020. https://arxiv.org/abs/2008.13146

[11] Jinah Kim, Minho Kwon, Sung-Dae Kim, Jong-Seong Kug, Joon-Gyu Ryu, and Jaeil Kim, "Spatiotemporal neural network with attention mechanism for El Niño forecasts," Scientific Reports, vol. 12, no. 7204, 2022.

[12] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," The Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.

[13] Yingbin Bai, Erkun Yang, Bo Han, Yanhua Yang, Jiatong Li, Yinian Mao, Gang Niu, and Tongliang Liu, "Understanding and Improving Early Stopping for Learning with Noisy Labels," CoRR abs/2106.15853, 2021.https://arxiv.org/abs/2106.15853

[14] Anders Krogh and John A. Hertz, "A Simple Weight Decay Can Improve Generalization," In Proceedings of the 4th International Conference on Neural Information Processing Systems (NIPS 1991), San Francisco, CA, USA, pp. 950–957, 1991.

[15] Chenxi Sun, Moxian Song, Shenda Hong, and Hongyan Li, "A Review of Designs and Applications of Echo State Networks," CoRR abs/2012.02974, 2020. https://arxiv.org/abs/2012.02974

[16] Taehwan Kim and Brian King, "Time series prediction using deep echo state networks," Neural Computing and Applications, vol. 32, 2020

[17] Claudio Gallicchio, Alessio Micheli, and Luca Pedrelli, "Deep reservoir computing: A critical experimental analysis," Neurocomputing, vol. 268, pp. 87–99, 2017.

[18] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek, "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation," PLoS ONE, vol. 10, 2015.

[19] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller, "Explaining nonlinear classification decisions with deep Taylor decomposition," Pattern Recognition, vol. 65, pp.211-222, 2017.
[20] Benjamin A. Toms, Elizabeth A. Barnes, and Imme Ebert-Uphoff, "Physically Interpretable Neural Networks for the Geosciences: Applications to Earth System Variability," Journal of Advances in Modeling Earth Systems, vol. 12, no. 9, 2020.
[21] C. F. Ropelewski and M. S. Halpert, "North American Precipitation and Temperature Patterns Associated with the El Niño Southern Oscillation (ENSO)," Monthly Weather Review, vol. 114, no. 12, p. 2352, 1986.
[22] Herbert Jaeger, Mantas Lukosevicius, Dan Popovici, and Udo Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," Neural Networks: The official journal of the International Neural Network Society, vol. 20, p. 335, 2007
[23] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps," arXiv:1312.6034, 2014.
[24] Herbert Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note,", Bonn, Germany: German National Research Center for Information Technology GMD Technical Report, vol. 148, p. 34, 2001.
[25] Herbert Jaeger, "Adaptive Nonlinear System Identification with Echo State Networks," In Proceedings of the 15th International Conference on Neural Information Processing Systems (NIPS 2002), Vancouver, Canada, pp. 609–616, 2002.
[26] Herbert Jaeger, "Echo State Network," Scholarpedia, vol. 2, no. 9, p. 2330, 2007.
[27] Alexander Woodward and Takashi Ikegami, "A Reservoir Computing approach to Image Classification using Coupled Echo State and Backpropagation Neural Networks," In Proceedings of the 26th International Conference on Image and Vision Computing, Auckland, New Zealand, p. 543, 2011.
[28] Nils Schaetti, Michel Salomon, and Raphaël Couturier, "Echo State Networks-Based Reservoir Computing for MNIST Handwritten Digits Recognition," In Proceedings of the IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) and 15th International Symposium on Distributed Computing and Applications for Business Engineering (DCABES), pp. 484–491, 2016.
[29] Yeon-Jee Jung, Seung-Ho Han, and Ho-Jin Choi, "Explaining CNN and RNN Using Selective Layer-Wise Relevance Propagation," IEEE Access vol. 9, pp. 18670–18681, 2021.
[30] Kingma, Diederik P. and Ba, Jimmy,"Adam: A Method for Stochastic Optimization," In Proceedings of the 3rd In- ternational Conference for Learning Representations, San Diego, CA, USA, 2014. https://arxiv.org/abs/1412.6980
[31] Marco Landt-Hayen, "NOAA Extended Reconstructed SST V5 as Supplementary Data for Publication," 2022. https://doi.org/10.5281/zenodo.6517186
[32] https://github.com/MarcoLandtHayen/ESN_Classification_LRP_ENSO

AUTHORS

Main author: M.Sc. **Marco Landt-Hayen**
Ph.D. student at Helmholtz School for Marine Data Science (MarDATA) in Kiel
Project on attribution of climate events and focus on exploring methods of explainable AI

Co-author / supervisor: Prof. Dr. **Peer Kröger**
Head of Information Systems and Data Mining group at Christian-Albrechts-Universität zu Kiel

Co-author / supervisor: Prof. Dr. **Martin Claus**
Junior Professor for Ocean Dynamics at GEOMAR Helmholtz Centre for Ocean Research in Kiel and Christian-Albrechts-Universität zu Kiel

Co-author / supervisor: Dr. **Willi Rath**
Data Science technician at GEOMAR Helmholtz Centre for Ocean Research in Kiel

UTILIZING DEEP MACHINE LEARNING TO CREATE A CONTEXTALLY AND ENVIRONMENTALLY AWARE APPLICATION TO PREVENT SPINALTENDONITIS

Barry Li[1] and Yu Sun[2]

[1]Northwood High School, 4515 Portola Pkwy, Irvine, CA 92620
[2]California State Polytechnic University,
Pomona, CA, 91768, Irvine, CA 92620

ABSTRACT

Recently, we have discovered when a person is using their computer, they often begin to lean forward toward the screen without noticing. Leaning forward can cause many problems in their body, especially to the back bone known as spinal tendonitis, and the problem can spread throughout the entire body [1][2]. I created an app to warn usersto sit up straight when they lean toward the screen too much, effectively protecting them from damaging their back bone. This app uses deep learning to calculate the body posture, and draw an imaginary triangle between the shoulders, hips, and knees [3]. The point at the hips is most vital in calculating the angle of the body. This app takes pictures in a given interval of time (by default 30 second), when the body leans forward, this angle decreases, and when the angle becomes lower than a given amount (by default 30 degrees), it will send a warning message to ask the user to fix their sitting posture [4].

KEYWORDS

Machine Learning, Application, Spinal Tendonitis.

1. INTRODUCTION

Individuals who use computers consistently are more likely to notice their tendons and muscles, primarily in the back and neck, deteriorate over time. These problems are just the beginning as many start with pain in one region of the body that then spreads to different parts of their bodies affecting an individual's overall productivity. Correcting sitting posture has recently been popularized. In order to prevent leaning, someone needs to keep track of how the computer user is sitting and fix his or her posture when leaning forward. However, no person can monitor the computer user for a long time, as they may have some other tasks to do. Therefore, it is important to develop an app able to detect the sitting posture and send a warning when bad sitting posture has been detected. The benefits of using this app are being able to correct bad sitting posture and having a better understanding of how often a person leans forward while using their computer. However, the consequences of using this app include misdetection when the app sends a warning message when the subject is not leaning forward. In addition, the app cannot always catch all cases of the person leaning forward. This can result in-app notification spams if the person does not fix the sitting posture which could in turn affect the user's working environment.

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 131-137, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122009

Some pose estimation techniques and systems have been proposed to detect the sitting posture; however, these proposals assume the user is sitting on a chair upright, which is rarely the case in practice. The estimation implementations are also limited in scale, with samples only given for sitting on the ground or bending knees upward. Other techniques, such as calculating the angle between the imaginary triangle, is used to determine if the person is leaning or not. To reduce complexity, the current algorithm used in these techniques cannot be too sophisticated and often results in errors in processing the picture and the angles of the sitting position. A second practical problem is that some users find it hard to understand how to use these methods, such as where to put the camera and the optimal application for maximum effectiveness of the detecting method.

In this paper, we follow the same line of research by finding how to detect a person's sitting posture [5]. Our goal is to figure out if a person is leaning too far forward or not. Our method is inspired by deep learning to detect the pose of a person. In order to do this effectively and efficiently, the program will locate points of the sitting person's body, mainly by machine learning to determine where the person usually sits and the location of their body parts. Second, it will locate the person's ears, hips, and knees and remember those points. Third, the app forms a triangle with the three points and calculates the angle of the point at the hips to get the angle measure result. Therefore, we believe that the calculation of the angle can help detect the sitting posture of a person and decide if he/she is leaning or not. This helps by figuring out how a person is sitting in their seat and if they are bending or not. If a person is found leaning forward, by notifying the angle is low enough, it will send a warning to tell the user to fix their sitting posture [6].

In two application scenarios, we demonstrate how the above combination of techniques increase the accuracy of finding out the sitting posture and the effectiveness of the detecting system. First, the usefulness of our approach will be proven by a comprehensive case study on the evolution of AI and deep learning to define where each part of the body of the user is located [7]. Second, we will analyze the evolution of motion detection in calculating the sitting posture of the person sitting in his/her chair. The coordinates of the points marked are accessible using the API provided and readable by code. An equation is used to find out what angle the person is leaning. The equation contains division ensuring a zero (0) is not allowed in the denominator. If the denominator produces a zero, the calculation for this image is automatically canceled and a warning message is returned to the end user. Whenever the AI fails to mark all of the points required for calculation, the equation will be automatically voided and will assume the user did not lean too much forward even if the person does. The AI still have some problems of detecting the person's body components and often result in not marking the points correctly due to environmental factors that can misunderstand the AI's detecting knowledge, such as too bright or dark, similar color the the person's clothing to the objects around, and objects blocking the camera's vision. We are still finding ways to overcome these obstacles and increase the accuracy of the camera. The camera we used does not have a high resolution and makes the AI somewhat harder to see things correctly.

This paper will be describing the project in more detail and will be organized in the following structure: The next section described the details on the challenges that we met during the experiment and designing the sample; Section 3 is focused on our solutions to overcome the challenges that we encountered and mentioned in Section 2; Section 4 presents how each challenges are overcome step by step, and followed by presenting the related work made by other people in Section 5. And, finally, Section 6 concludes this paper by briefly going over the contents on the paper again, and pointing out the future works that we be do to improve the project.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Building, installing, and setting up the Raspberry Pi

The first challenge I encountered was building, installing, and setting up the Raspberry Pi. The first step was to buy all the parts necessary to build the Raspberry Pi: A Raspberry Pi computer with an SD card or micro SD card, a monitor with a cable and HDMI adapter, a USB keyboard and mouse, and a reliable power supply. After reading a quick tutorial, the build went pretty smoothly which solved most of the hardware issues. Unfortunately, the software installation was much more challenging. I quickly found out that I was unfamiliar with the Linux Debian-based operating system [8]. I used YouTube tutorials to guide me through how to use the Linux operating system, and, after some basic command prompts and reboots, the single board computer was ready to use. In order to set it up, I need to set it up with the remote control. I installed VNC viewer, an app on Windows computers to access Raspberry Pi remote control using wifi.

2.2. How to use AI

Another challenge I faced was researching how to use AI to detect where each body part of the user is located in the image. After capturing an image of the subject's entire body, the app automatically displays the coordinates of the triangular points on the image. Lines are then virtually drawn between the coordinates of the points given. Only the points that are very necessary are connected, and these points are then calculated in order to determine the path of the lines. The detection gives the computer data to analyze how the person is sitting and if the person is leaning too much forward. The module sometimes did not recognize the position due to the limited angle of the camera. An additional roadblock was the fact that if the color of the clothes was similar to the background then detection was often more inaccurate. We adapted the app to take a video and picked out the best frames from video instead of only taking a picture. The AI image recognition then produced results with much better accuracy during the real-time process.

2.3. Setting the programming environment up

The final and last challenge was setting the programming environment up. The operating system of this Raspberry Pi is Rasbian [9]. It is called Rasbian because it is an alternative version of the Debian system, and, therefore, most of the reliability libraries cannot be easily installed with pip. Pip is a library that contains codes that are used for taking images. In order to solve this problem,I need to look for different wheel files to identify any problems or errors that may occur during programming. Wheel files are used to determine the function and execution of each pip library. If there is a slight mistake in the coding of the wheel files, then the pip library will also fail, so it is essential to ensure the wheel files are set up correctly before continuing with any additional programming. For each step taken, I used research to find out how calculation and code works, and used my own programming skills to perfect the code.

3. SOLUTION

After successfully conquering the three challenges above, I was able to create a system that efficiently and effectively determined the sitting position of the user. At first, when the camera is set up, it is connected to Raspberry Pi and ready to take pictures. I then run the program in the

computer Linux terminal, but not the IDE terminal because TensorFlow does not work in IDE as the IDE terminal is not allowed to modify files and TensorFlow requires the plotting the dots and drawing the lines [10]. The TensorFlow function also returns the coordinates of the points marked relative to the image, which can be used for more purposes than just marking on the image. The dataset of the coordinates of the points marked are later put into the looping function that runs the calculation of the angle located at the hips. I found the equation online that results in the angle of one point, and assumed the hip to be that point and result in the angle. However, the dataset can sometimes not contain the coordinated required points either because it is out of image or blocked by another object in between the camera and the person. In response, I decided to implement the try function that will let the compiler verify if the code in this segment is able to run (contains all the points needed) or not. If any required points are found missing, the try function will determine it not to work and skip the function, I default this to a normal sitting posture if the equation will not run. If the angle is found to be less than 60 degrees, the alert message will appear, asking the user to realize their sitting posture and fix it back to the upright position.

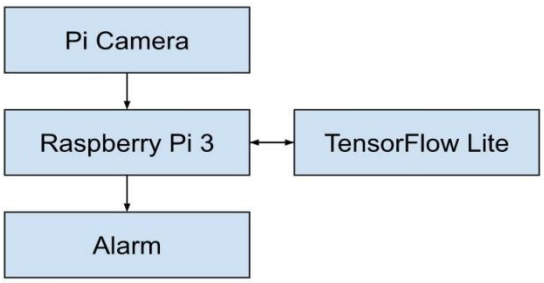


Figure 1. Overview of the solution

I used a Raspberry Pi Camera Module 2 and a code library so the images captured from the camera would automatically save onto the Raspberry Pi computer. In order to increase the accuracy of the position calculations I then instructed the camera to capture requires a camera installed to the Raspberry Pi to take videos instead of pictures then draw lines that connect the three major points plotted between the user's ears, hips, and knees. In addition to the hardware components such as the Raspberry Pi and Camera Module 2, I also used machine learning from TensorFlow to mark the body components that are used to define the angle of the person sitting. It can also mark other main components, such as eyes, nose, and feet, but those are not necessary for the sitting pose application. While the camera on my Raspberry Pi console is low and sometimes cannot see that well, it caused some problems while I'm testing my app. The program will send the message to the user's screen when leaning too much, prompting the user to fix their sitting posture. It currently only shows up in the computer's command console and does not always appear at the window in the top layer, it can be blocked by other windows. However, when used in modified applications, it is possible to show the message in a new window that pops out when bad sitting posture is detected. I developed this application using a python IDE canned Thonny Py, it is simple to use and, suitable for some advanced code development. Some obstacles in using Raspberry Pi are low performance, the demand of online searches, and unfamiliarity with the operating system. The low performances are natural to Raspberry Pi since it is a small computer that uses low-end hard-wares, this causes some limitation on the variety of the application that can be created, but it is suitable for the sitting pose detection. Online searches are done for the resources needed and how to build and use the code. They are essential for the creation of usable code and to modify the code to make it fit for the application. I'm also not familiar with Rasbian, or the operating system of Raspberry Pi, which I need to do research on how to use the operating system. Rasbian mostly uses the controls of Windows, but its terminal is based on Linux, which I'm not familiar with, for example, when I used ctrl+c and ctrl+v it

didn't work and I must use right click and select from the menu to operate copy and paste. The console can perform actions that cannot be done inside the programming IDE and are often run by a shortcut on the main screen.

4. EXPERIMENT

4.1. Experiment 1

At first, I used a Pi Camera to take the picture, but it was blurry and tinted reddish, this was because of the specification and the lens of the camera. I first took a picture of myself and found out that the lines are drawn incorrectly and some dots are missing because of my camera. However, if the person is visible enough, the program works as it is intended and draws lines along my body. It shows TensorFlow can actually be used to detect the posture of the person sitting and know the angle that the person is leaning too forward or backward.

After the computer draws lines on the image using TensorFlow, it appears to give out the coordinates of the points on the image. The coordinates can be used to fill in an equation to calculate the angle of the person is leaning. The equation is being used to calculate the angle of the three dots formed as a result of the angle taken on the hips. I first tested out sitting straight up and it showed my angle of tilt is about 90 degrees, which is the perfect angle. While also some pictures, since not all three points are marked, did not fulfill the inputs the equation needed, and the program crashed. The program often said that I cannot divide by zero and the program will not be working consistently. The error was due to the missing number of the points taken since some parts of the body cannot be analyzed.

I later rewrote the code to prevent the error from affecting the code running and stopping the code. I added the try method to catch the error if the code is interrupted by division by zero. With the new code in place, even if not all three dots are marked, the program will not crash, but rather not show the message if the person is leaning too much forward. It can keep the program running even after a point is missing, and prevent it from crashing. It also takes work from providing an alternative procedure for the code to execute when it is not able to operate properly.

When I run the test again, the program works as it is intended, it warns me when the angle is too low, or lower than 60 degrees. The program keeps running without crashing when something wrong with the marking point happens. It can be configured to show an error message on the console while still running, and notice other developers that something is wrong, such as not all required points are marked, asking them to consider testing in a better environment that is more free from being blocked by objects such as the chair. When I fix my back position to upright, the message disappears and is no longer sent in the console. However, the display window is still not implemented so the message still appears in the console, which can sometimes be blocked by other windows open. I'm still finding resources to help me implement the displaying of the warning message in a separate window.

When I find ways to build and install Raspberry Pi, I first build my Raspberry Pi computer with my instructor to install the parts of the computer into its provided shell. To set Raspberry Pi up, I followed the instructions given by my instructor and on the display screen of my Raspberry Pi to install the operating system. I later take some time to explore and get to know how to use the newly installed Raspberry Pi and its operating system. Once I started designing the application, I realized that I will need an AI library installed to take pictures and mark the body parts. I researched on the internet for an AI usable by Raspberry Pi to detect the person's sitting posture, and I found TensorFlow as my option. TensorFlow allows me to detect how a person's posture is,

like standing, sitting, or anything that involves moving. I finally researched for the equation to calculate the angle of the person sitting and made the code function as I tendered to.

5. RELATED WORK

Poor sitting posture has led to a variety of spinal disorders, per the research on A sitting posture recognition system based on 3 axis accelerometer [11]. It uses a tri-axial accelerometer to measure how a person is sitting by attaching the accelerometer on the back of the subject's neck to measure 5 types of sitting posture. It uses an accelerometer attached to the person instead of a camera taking pictures of the person which mine does. It has a better ability to detect the person as it is attached to the subject directly instead of taking a picture and evaluating how the person is sitting.

A method to predict 3D positions of body joints is from a research called Real-time human pose recognition in parts from single depth images [12]. It involves mapping the difficult pose estimation problem into a simple per-pixel classification problem. These methods also useimages to determine if a person is leaning too forward or not, and makes use of the camera's estimation of the person's body parts. However, it also involves turning the image into a simpler per-pixel detection than just marking points on a natural image. The program makes better use ofthe image detection and body mapping functions by giving the computer an easier reading of the image and the body.

Personalized services that improve the living environment of a person involve sitting posture, the project Sitting posture analysis by pressure sensors shows how to use pressure sensors on the seat to determine how a person is sitting [13]. The classification rate of an unknown person is about 93.9%, compared to about 98.9% for a known person. The project involves using a pressure sensor on the user's seat that receives information by the weight of the person sitting on it, the weight changes when the person is sitting in different postures and can detect 9 different postures. The body weight measurement tends to be less accurate than using camera detection because body weight pressure on the sensor may not determine the sitting posture in an understandable way.

6. CONCLUSIONS

I have created a sitting posture detection app that involves taking pictures and using Tensorflow to determine how a person is sitting using markings on the body skeleton and the angle of the hips with the intersecting lines of the head and lower spines and the knees to the hips. The main purpose of my program is to solve the back spine getting injured while using the computer. I used a Pi Camera to take pictures and TensorFlow Lite to analyze the sitting posture of the person [14]. If the person is leaning too much forward, the angle located at the hips of the person is lower and sends a warning message when this angle reaches below 60 degrees to tell them to fix their sitting posture. I applied this method by an equation that calculates the angle ratio between the three points and returns the result, which is the point that is marking the hips. TensorFlow is used to draw out the person's skeleton in the image and returns the coordinates of the dots marking in the image. The equation is researched on the internet and written separately from the libraries I downloaded, and involves using trigonometric functions. I also implemented the error catch when the equation attempts to divide by zero. The program sometimes does not work when at least one point that is required for the calculation is missing, often by something blocking the body from the vision of the camera. It can be effective at most times in detecting bad sitting postures but sometimes not doing so that well.

The limitations of my program includes the chance of a picture of the full person captured by the camera without anything blocking in the way, the detection of the lower spines are also critical toward a good sitting posture but cannot be implemented and well detected because of the limitation of the points drawn by TensorFlow, and the ability to launch by conventional ways, such as a shortcut on desktop or built into an application. The person can sometimes be blocked by other objects that reduce the proficiency of the camera detecting the person's sitting posture. And the built in app is not created because I lack the requirement of the knowledge of opening a new window with Python [15].

In the future, I will implement a better way of taking the picture taken and adding a pop-up window to warn the user to make the app easier to use. I should also recommend the distance between the camera and the person when the camera is detecting if a person is leaning or not. It makes the app much easier to use if these features are implemented and more functional to solve the health problem caused by leaning too forward toward the screen.

REFERENCES

[1] Nachemson, Alf. "The effect of forward leaning on lumbar intradiscal pressure." Acta Orthopaedica Scandinavica 35.1-4 (1965): 314-328.

[2] Artenian, D. J., et al. "Acute neck pain due to tendonitis of the longus colli: CT and MRI findings." Neuroradiology 31.2 (1989): 166-169.

[3] Deng, Li, and Dong Yu. "Deep learning: methods and applications." Foundations and trends® in signal processing 7.3–4 (2014): 197-387.

[4] Andersson, BJ Gunnar, et al. "The sitting posture: an electromyographic and discometric study." Orthopedic Clinics of North America 6.1 (1975): 105-120.

[5] Claus, Andrew P., et al. "Is 'ideal'sitting posture real?: Measurement of spinal curves in four sitting postures." Manual therapy 14.4 (2009): 404-408.

[6] Harbach, Marian, et al. "Sorry, I don' t get it: An analysis of warning message texts." International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2013.

[7] Serag, Ahmed, et al. "Translational AI and deep learning in diagnostic pathology." Frontiers in medicine 6 (2019): 185.

[8] Emm anouil, Emmanouil, and Angelo Sifaleras. "Implementation of an Open Source Optimization Software Package for Debian-Based Operating Systems."

[9] Kurniawan, Agus. "Programming on Raspbian OS." Raspbian OS Programming with the Raspberry Pi. Apress, Berkeley, CA, 2019. 79-96.

[10] Abadi, Martín. "TensorFlow: learning functions at scale." Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming. 2016.

[11] S. Ma, W. -H. Cho, C. -H. Quan and S. Lee, "A sitting posture recognition system based on 3 axis accelerometer," 2016 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2016, pp. 1-3, doi: 10.1109/CIBCB.2016.7758131.

[12] J. Shotton et al., "Real-time human pose recognition in parts from single depth images," CVPR 2011, 2011, pp. 1297-1304, doi: 10.1109/CVPR.2011.5995316.

[13] Kazuhiro Kamiya, Mineichi Kudo, Hidetoshi Nonaka and Jun Toyama, "Sitting posture analysis by pressure sensors," 2008 19th International Conference on Pattern Recognition, 2008, pp. 1-4, doi: 10.1109/ICPR.2008.4761863.

[14] Symon, Aslam Forhad, et al. "Design and development of a smart baby monitoring system based on Raspberry Pi and Pi camera." 2017 4th International Conference on Advances in Electrical Engineering (ICAEE). IEEE, 2017.

[15] Borcherds, P. H. "Python: a language for computational physics." Computer Physics Communications 177.1-2 (2007): 199-201.

A CRYPTOGRAPHICALLY SECURED REAL-TIME PEER-TO-PEER MULTIPLAYER FRAMEWORK FOR BROWSER WEBRTC

Haochen Han[1] and Yu Sun[2]

[1]Troy High School, 2200 Dorothy Ln, Fullerton, CA 92831
[2]California State Polytechnic University,
Pomona, CA, 91768, Irvine, CA 92620

ABSTRACT

P2P(peer-to-peer) multiplayer protocols, such as lockstep and rollback net-code, have historically been the cheaper, direct alternative to the Client-Server model. Recent advances in WebRTC technology raise interesting prospects for independent developers to build serverless, P2P multiplayer games on the browser. P2P has several advantages over the Client-Server model in multiplayer games, such as reduced latency, significantly cheaper servers that only handle handshakes, etc. However, as the browser environment does not allow for third-party anti-cheat software, having a secure protocol that catches potential cheaters is crucial. Furthermore, traditional P2P protocols, such as deterministic lockstep, are unusable in the browser environment because different players could be running the game on different browser engines. This paper introduces a framework called Peercraft for P2P WebRTC games with both security and synchronization. We propose two P2P cheat-proofing protocols, Random Authority Shuffle and Speculation-Based State Verification. Both are built on known secure cryptographic primitives. We also propose a time-based synchronization protocol that does not require determinism, Resynchronizing-at-Root, which tolerates desynchronizations due to browser instability while fixing the entire desynchronization chain with only one re-simulation call, greatly improving the browser game's performance.

KEYWORDS

Cyber Security, Anti-Cheat, Peer-to-Peer multiplayer, WebRTC.

MODULE 1, INTRODUCTION

Our framework is built on top of WebRTC, a browser peer-to-peer [1] technology that allows for both video/audio calls and the sending of arbitrary data[2], through UDP or TCP. Because each peer is hiding behind a NAT, a handshake server is required to perform hole punching[3]. This framework uses Peer.js, a high level wrapper for WebRTC API [4], so the hole-punching is handled automatically [5]. Users can also choose to implement their own Peer.js hole punching server, but this is out of the scope of this paper. Once a direct connection is established, the handshake server is no longer needed.

I, Contributions

The main focus of this paper will be about preventing cheating and maintaining synchronization while optimizing for a browser environment. Security protocol is much more important in browsers, as browser games rely on third-party anti-cheats [6]. It is also significantly easier to

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 139-155, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122010

manipulate browser games state as the codes are open source. For this purpose, we introduced two new protocols, Random Authority Shuffle and Speculation Based State Verification , to let the network catch cheaters. Random Authority Shuffle uses established cryptographic primitives(hash commitment scheme) to perform synchronized switching of the authority of clients. It ensures at a given tick, a player is only authoritative of the data of a random, remote player. This is combined with speculative state verification, where if a cheater attempts to manipulate the data based on its assigned authority over a random, remote player, non-cheating clients will compare their non-authoritative data and deny the manipulated data from the cheater, even if the cheater has authority. We also introduced a novel state synchronization technique that does not require determinism while minimizing the amount of re-simulation required due to desync, called Resynchronizing at Root.

II, Existing Methods

As of now, there are 3 major types of netcode implementation for browser multiplayer game:

1, Client-Server with authority server
2, Peer-to-Peer with one single authority “client host”
3, Peer-to-Peer with deterministic lockstep, input delay, and/or rollback

Peer-to-Peer style multiplayer, either with client host or lockstep, offers advantage in terms of lower cost and latency due to elimination of third party servers [7]. Deterministic lockstep, where clients broadcast its input rather than state, is commonly used in RTS(real-time-strategy) games as the sheer amount units makes sharing and synchronizing state impractical. Rollback, in which clients predict next remote input from previous remote input, is commonly used in fighting games, where the fluidity and low latency of peer-to-peer structure help promote better gameplay experience. Most multiplayer IO games, such as agar.io, uses Client/server method. As the number of connections needed for each client increases with the number of players, peer-to-peer networking tends to be less practical than client/server methods. Therefore, the focus of this paper and framework will be lobby-based multiplayer match games with less than 30 players rather than massively multiplayer IO games. The P2P single-authority client host methods are commonly used in games with private lobbies. Although convenient, it grants the host an unfair advantage as it has authority over all other clients. Both lockstep P2P and rollback P2P distribute authorities to each client. In other words, they are fully decentralized. Strictly deterministic lockstep aims to reduce the amount of data transmitted between clients by sending inputs from each player rather than state, and allow each client to calculate the future game state based on input. If the game is deterministic, then clients should always yield the same state. Network latency is solved through input delay, where clients issue a command with an execution time guarantee to be later than the package receive time on other clients. For example, client A issues a “move” command at tick 1, but specifies execution at tick 3. At tick 2, client B receives the command. At tick 3, both clients execute the move command.

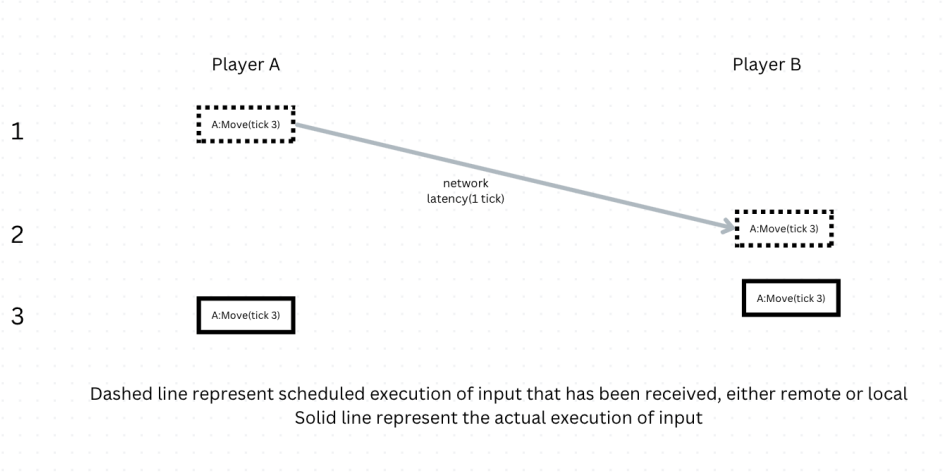


Figure 1. Figure of example

The unresponsiveness of input delay is usually solved with graphical feedback, in which local players' character immediately execute a command but run cause game-state altering at the synchronized execution time.

Another solution is using rollback to reduce the input delay interval. In rollback, clients predict movement of other players, often based on previous input. Inaccuracy in prediction is corrected when a synchronization packet from the other player arrives. There are also newer netcodes which completely eliminate the element of input delay, such as in Skullgirls, where clients are allowed to "steal frames" in order to catch up to inputs. Because human reaction time for visual stimuli is 250 millisecond when untrained, 2-15 frames of input delay is usually acceptable outside of fighting games(Reaction Time Study: https://www.scientificamerican.com/article/bring-science-home-reaction-time/) .

Existing lockstep ensures synchronization through waiting for the slowest client to send end-of-tick packets(often containing state). This is slow and impractical for many fast paced browser IO games,where it is not necessary to wait for a lagging player. Existing industry solution is deterministic lockstep + input delay. Some frameworks implement rollback to reduce the input delay time. The GGPO rollback SDK is currently the industry standard for implementing rollback in Peer to Peer games. This paper proposes a new framework with an alternative solution to rollback specifically targeting WebRTC and the browser environment [11]. There are several problems that we will solve in this paper. As the browser environment is nondeterministic, it is previously unrealistic to eliminate the lockstep wait through determinism. This paper proposes a new protocol called Timing Synchronization, in which it takes advantage of date/time on each device to heuristically ensure lockstep(without having to wait for the laggiest player), with regular state correction packets to eliminate simulation differences on each machines due to non-determinism. Secondly, most existing P2P multiplayer netcode does not provide an effective scheme for anti-cheat. This paper introduces a preliminary input/state verification system to allow players to catch cheaters. There are caveats to this system—- although universally applicable in all games in preventing unwarranted gameplay input or state manipulation on remote clients, it does not allow for private data. As all data are shared between all clients, this framework is more applicable to fast-paced shooter games or any games that do not include a de facto fog of war.

III, Paper structure

In Module 2, we explain the detail of Random Authority Shuffle and the cryptographic primitives behind it. In Module 3, we introduce the synchronization suite for this framework- time based lockstep, input delay, and Resynchronizing at Root. In module 4, we discuss the specific implementation detail in a browser environment, and network-related solutions such using UDP packets while maintaining reliability. In module 6, we reiterate the main ideas presented in this paper. In module 7, we suggest Zero-Knowledge-Proof as a future research direction for allowing cheat-proof private data in generalized use cases beyond TCG games.

One key concept to understand is that in this protocol, the term authority is always relative. To prevent state manipulation, if the client with authority deviates significantly from other players, it will be detected as a cheater. In our proposed Desynchronization Tracing method, a non-authority client can detect the data from an authority client is faulty due to authority client desynchronization, and thus ignore the data. The authority client's resimulation will then bring it to the same true state as the non-authority client.

Module 2, Cryptographical Cheat-Proof

A. Threat Model

In P2P, there are no impartial third parties. There are several potential ways in which a player could cheat.

1, Host Cheating - if a player is granted all authority as the de facto "server" in a pseudo client-server simulation, it could alter state to its will. As our protocol uses peer-to-peer state authority, this is not an issue.
2, Peer to Peer state manipulation - if state authorities are distributed among all peers, then one peer could report manipulated state to other peers
3, Data Peeking - in a game where "fog of war" exists, a player should not be able to see the secret data of the other player.
4, Probability Manipulation

#1 is not a problem as it only applies to pseudo client-server p2p. This paper introduces techniques, namely, Random Authority Shuffle and Speculation Based State Verification to address #2. This paper also uses a cryptographic commitment scheme to generate unbiased, synchronized random numbers.

I)Prevent State Manipulation

In a distributed environment, where each player has authority over their own data, it is very easy for a player to falsely report his/her own data to give himself/herself an unfair advantage.

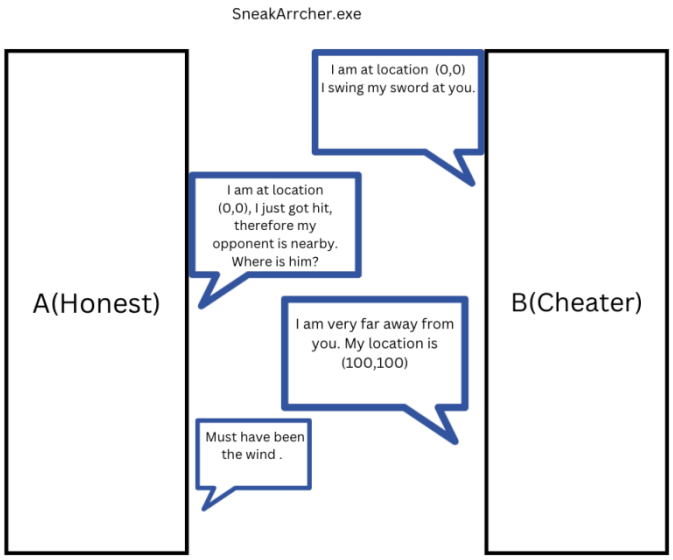


Figure 2. Sneak Arrcher

To catch state manipulation, the common approach is to implement a verification function for each state field and input to catch improbable reporting. For example, the game programmer could hardcode state rules like "positional change between tick cannot be greater than speed of a player, unless the previous 10 input stack has blink skill used". When these rules are violated, it is likely that the opponent is a cheater. For input, rules can be like "player cannot send attack input twice in one single tick". These rules, although successful in catching clear violations, often fall short in covering all potential cases. This is because cheaters can perform subtle manipulation on the state based on specific situations to get around rules. For example, when a cheater is about to get hit, being 0.1 meter away would result in avoiding the damage completely, thus giving the cheater a significant advantage while not being detected. To mitigate this, we propose two techniques.

1, Randomized Authority Shuffle
2, Speculation Based State Verification

Random Authority Shuffle

1)Logic

If a player has absolute authority over its data, the only way an honest player can catch a cheater is through careful inspection. To add in an extra layer of security, we propose Random Authority Shuffle protocol, which allows for three constraints

1, players can only be in "authority" of the state of a random remote player
2, before each tick, player cannot predict who's state it will be in charge of
3, the entire network, however, will reach a consensus on authority responsibility at that time.

Consider a hypothetical cheater. To give itself an advantage, the cheating client will report that its health is full 100/100, despite being hit. If the cheater has authority over its own data in every tick, and reports the manipulated data, then other remote clients will falsely assume that they

have desynchronized and thus change their state of B to 100/100. B will thus never take damage. However, in the Random Authority Shuffle, B will never be in authority of its own state. This limits the possibility of cheating into sabotaging the opponent's data. But since on every tick, the cheater is reassigned a random player's data to be in authority of, it is very hard for it to target down its opponent. Of course, the cheater could report manipulated state like health = 0 every tick for player A until the cheater is in charge of the state of player A. However, consider the previous example where the cheating player moves 0.1 meter to avoid getting hit. It is highly specific to the data of the tick. If the cheating player wants to perform this cheat by moving A away, it has to be lucky enough to be in charge of A's state, at the exact tick player A hits the cheater. If player A attacks once per second(15 ticks), In a lobby with n players, the probability of catching the attack is 1/(n-1)*1/15. This makes subtle, carefully tuned specific state manipulation unviable in a lobby with a large number of players. The cheater, forced to resort to brute force method such as reporting " player X(the player cheater happen to be in charge of) has health 0" on every tick, can be easily catched by the conventional method of Rule Based input verification, and our method of Speculation Based State Verification.

2)Cryptographic Commitment Implementation

For Random Authority Shuffle to work, the entire network needs to agree on an unbiased random value at a given tick. We will use a PRNG(pseudo random number generator) algorithm, because it is deterministic. For every 10 ticks, the decentralized network will generate an agreed-upon unbiased random seed, and use it for 10 ticks. We used a cryptographic commitment scheme to ensure the true randomness of the scheme [8]:

1.all players determine a number - this number can be generated through a true random number generation algorithm, or can be manipulated by a cheater. Once the number is determined, each player hashes the number and sends the hash string to every other player. This hash string will be "commitment"

a.SHA-256 is preferred as it avoids collision

2.Once all hashes have been received on all clients, players send the actual number to each other. Players can then hash these numbers to confirm that other players did not deviate from commitment.

3.All players then add up the numbers together. The sum will be used as the random seed for the next 10 ticks.

This commitment prevents a situation where a cheater intentionally delays sending out its number to receive all numbers from other clients first. Without commitment, the cheater can simply sum up other clients' numbers and send out a reverse engineer number that will result in a random seed that can give a favorable result for the cheater.

Once the random seed is synchronized, each player would then independently use that seed to determine the authority arrangement of the network. Since the random algorithm is deterministic, this is guaranteed to result in the same authority arrangement. If a cheater intends to cheat by claiming that it has authority over player A's state, player A and player B will simply ignore the cheater's authority state patch for player A.

1.Assume there are 3 players, A, B, C

2.each player calculate three(number of players in the network) random value, assign them to A, B, C, respectively in the same order

a.because the random algorithm is deterministic, and the seeds are the same, the three random value for A,B,C will be the same across all peers

3.Each player then compares the value for A, B, C and arranges them in Descending order. Let array P be the ordered arrangement of player, arrangement is determined as following

a.P[i] has authority over P[i+1]

b.if P[i] is the last player in the array(i+1 > OP.length), P[i] has authority over P[0]

As there are still chances for the cheating player to be lucky enough to manipulate the state of a specific player while also getting pass rule-based state verification, we use another technique, called Speculation Based State Verification.

Speculation Based State Verification

Since each player runs a near-deterministic simulation of all other players in our framework, it is possible to use the simulation of the remote player's state, also known as speculation, to detect cheating. If multiple players disagree with the potential cheater, then they reject the state patch from that potential cheater even if it has authority. See example

1.The cheater has authority over the state of client A after random assignment.
2.The cheater sends out a state patch to all clients, claiming that client A's health equals to 0.
3.Client A receives the authority state patch from the cheater for client A. It detects discrepancy.

a.client A's speculation for its own state is {health_of_A = 100}
b.the state patch from cheater is {health_of_A = 0}
c.client A request a Convention of Speculators to all clients
d.all clients will send to all other client their own speculation of client A's data
e.all clients then tally the result
f.each clients will discover that client A, B, C all speculated client A's health to be 100, while the cheater claim client A's s health is 0
g.the cheater is detected

II)Cryptographic Random Number Generation

The same algorithm for Random Authority Shuffle is also used for network random number generation. It is modified to satisfy the following constraints

1.cannot be manipulated

a.Since each client needs to send a commitment hash, they cannot modify their partial seed number after receiving other clients' partial seed number to make the seed sum favorable to them. The existing solution for Random Authority Shuffle already satisfies this constraint.

2.cannot be predicted

a.in Random Authority Shuffle, players reconvene every 10 ticks to repeat the seed generation process. This means players can only predict future states up to ten ticks. However, this is not good enough for turn-based games. Take the example of Monopoly. One player can gain a significant advantage by knowing what the next two dice rolls will be and prepare accordingly. Therefore, modification is needed

To make the game truly unpredictable, where a client cannot cheat by simply look ahead by running a copy of the PRNG model with the shared random seed, the seed should change, ideally every tick [9]. It is impractical to re-execute the same cryptographic commitment seed generation process on every tick. Therefore, we used a different technique basing off player input. Input data, unlike state, is not subject to indeterminism or desynchronization as we designed the UDP packet to guarantee arrival of input through redundancy(more on implementation detail section).

Therefore inputs for a tick in the past is guaranteed to be the same across all players. We utilize this to modify the seed in an unpredictable yet synchronized manner

1.At given tick X, find inputs that are executed at tick X-2 for all clients.
   a.Because we use input delay, inputs for tick X -2 are likely sent at tick X-4. This margin heuristically ensures that the inputs have arrived across all machines. The margin can be increased to compensate for network latency.
2.Convert the input names into ASCII numbers and add them together, resulting in number N.
3.The new seed will be the previous seed (S%N)*(N%S) * S.
   a.modulus is used to cause greater fluctuation while maintaining integer operations
   b.A Sin function could also be used, however, it must be implemented in a way that does not break determinism through floating point as small error in random seed can result in big difference in NRNG generated number.

Because player inputs are most likely unpredictable, no player can guess the new random seed to gain an advantage.

Module 3, Protocol

A.Designing for a Browser Environment

This section introduces the technique we used to achieve near-lossless synchronized multiplayer simulation on Peer to Peer WebRTC browser environment [12]. Browser environment are fundamentally different from PC game environment because are several constraints:

a.No guarantee on continuous simulation:

Browsers can pause a page's event loop and execution when deemed necessary. This means any framework need to account for such pause in order to be stable

b.inherently non-deterministic

Unlike PC games, developers cannot enforce determinism as players could play on different browser engines AND different computers.

Constraint (a) means that tick rate and packet arrival time is not-guaranteed to be consistent across different machines. In a traditional lockstep, players wait for tick packet confirmation to continue simulation. This delay is not feasible as browsers could freeze webpage when deemed necessary, stopping the simulation for the entire lobby. Traditionally, this requirement on waiting for tick updates is lifted by input delayed determinism. But there are two problems with input delay determinism on browsers: 1st, determinism is infeasible. 2nd, the tick rate is not guaranteed to be synchronized. We designed a novel solution that does not require determinism for synchronization nor waiting for player's tick advancement confirmation, while ensuring all clients run on the same tick. This framework can be divided into two components. First, time bound lockstep simulation with input delay. Second, synchronization protocol through state patch packets.

B.Time-Based lockstep

Ensuring that all connected clients run the same tick at any given time is crucial for synchronizing because the protocol performs catch-up/correction based on global tick. To achieve this, a frequent, consistent update loop is needed,preferably equivalent to 60fps. The

update loop is done through web-worker multi-threading, with detail explained in the framework implementation module.

We used a time-bound technique to ensure tick synchronization. This is under the assumption that browser javascript function Date.Time() returns an accurate relative representation of how much the time has passed. Note that Date.Time() does not have to be synchronized — it could differ based on different implementations. Our protocol have the following unique advantages

1.Synchronized Tick
2.Does not need to wait for lagging player to catch up- lagging player would automatically catch up without delaying the whole network

Below is the specific detail of the tick network synchronization protocol

1.The "host" client is responsible for distributing an accurate start time. Before the game starts, the host client sends ping packets, calculating the RoundTripValue.
a.Network latency for each connected client is calculated as NetworkLatency[i] = RoundTripValue[i]/2
b.to ensure accuracy, this process is repeated several time to find Median Value
i. MedianNetworkLatency[i] - the median network latency of client i
2. The host sends out a start packet with specified starting delay based on median network latency.
a. host A find the client M with biggest MedianNetworkLatency[M],
i. calculate global delay value as D = MedianNetworkLatency[M]+ padding(30ms)
b. send out start packet with specified starting delay(S)
i. for each client i, S(i) = (MedianNetworkLatency[M] - MedianNetworkLatency[i])+ D
c. when each client i receive the packet, start the game after delaying for S(i)
3. This ensures all clients start the game exactly D millisecond after host A sends out the start packet

After D millisecond, all clients would start at roughly the same same time, assuming the MedianNetworkLatency[i] is accurate. All clients will capture this time and use Date.Time() on their local machine to store the moment as InitialTime. Specific implementation of the delay will be discussed in the Framework section. Afterward, all clients jumpstart the tick loop. Some slight difference in framerate is likely unavoidable. For example, despite our protocol, there could still be a 10ms difference between the InitialTime of each client. To solve this, we designate a 15 tick per second tickrate. At any given moment T(fetched through Date.Time()), tick is equal to floor((T-InitialTime)/66). All inputs and state adjustment packets will be sent at the end of each tick and executed at the start of specified execution tick. This allows toleration for 66ms error, while not breaking the tick-to-tick synchronization.

C.Lag Catch-up

In certain scenarios, browser execution of the game may freeze temporarily. Although usage of worker thread prevents most of the issue, there could be situations in which players miss a few ticks. This is designed to ensure the simulation is lag-proof

1.At the start of each tick loop function, players fetch time T from date.time() and calculate its own tick based on floor((T-InitialTime)/66).
a.assume there is a 2 tick delay and player missed tick 3 and tick 4 , and is currently on tick 5

2.for each missed tick(3, 4), run the Tick Execution Sequence, fetching and running state patch and input for tick 3 and 4, instantly

a.input delay, documented later, ensures all input runs on a specified frame across all clients.

b.State Patch is explained in the Input/State Synchronization section

3.run tick 5

D.Input delay

Because ticks are ensured to be synchronized, we use input delay to make sure input executes at the same tick on all clients. The delay amount is heuristically calculated and would be discussed in the implementation section.

E.Concurrent Simulation/State Patch/Authorities

All examples here are simplified to not include Random Authority Shuffle.

First, some terminology/symbols should be specified.

- Xa = position of a
- Xb = position of b
- Ha = health of a
- Hb = health of b
- State Patch - a packet containing state of a remote player at a certain tick, and usually arrive at a tick later than the specified tick
- Speculation - players perform inputs received from remote clients and simulate their state for them. However, since no determinism is guaranteed, the state for remote player may not match the authority state on remote player and thus is mere speculation
- Local Authority state - the state of a player, on its machine.
- Resynchronize - when a state patch for tick X for player B arrives, player A check for its speculation at tick X for player B. if they are different, player A revert all state back to tick X with correct state for player B (based on state patch), and instantly rerun all of the input after tick X in order
- Tick Packet - each tick's input, state patch, and desync label is packaged and sent out in one UDP packet. This packet could be dropped. See Framework/implementation section for detailed solution to dropping frames.

Each client runs a simulation for players of self and other clients. The specific implementation of this structure is discussed in the implementation section. Each client only has authority on its own, local player data. For player data of remote clients, the local client maintains a speculative prediction called Speculation based on synchronized input simulation. In an ideal situation, where all clients maintain perfect determinism, this prediction is enough. Consider the following case:

- Player have 2 skills, move and attack(which only work if two players are adjacent)

In a simulation lacking determinism, the following could happen:

1.Player A and Player B start at Xa = 0, Xb = 2, respectively. For simplicity sake, there are only one dimension
2.At tick 1, Player A issues a move command(which would change Xa to 1), scheduled to execute at tick 3. (optionally, user of the framework could choose to extrapolate player to location without delay to improve responsiveness)
3.At tick 2, Player A issue attack command ,scheduled for tick 4
4.At tick 3, both player A and player B execute the move command. On Player A's machine, Player A ends up at Xa = 1. However, due to non-determinism, on Player B's machine, Player A ends up at Xa = 0.9.
5.At tick 4, both player A and player B execute the attack command. On A's A's attack will register, causing B to lose health. However, on B's machine, A's attack would not register, causing a desynchronization.

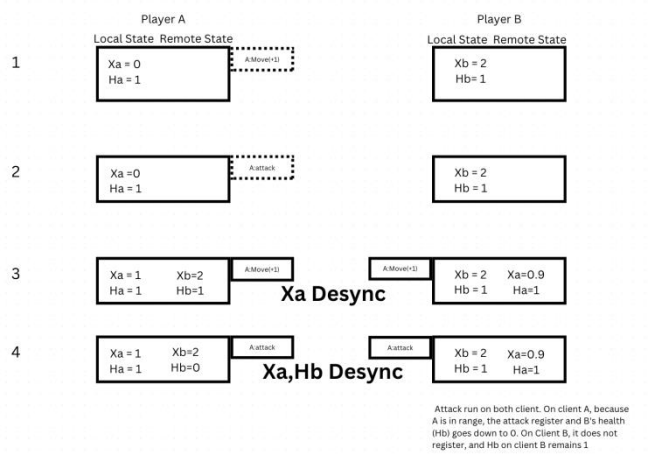


Figure 4. Player A vs Player 2

Note that in A's speculation of B is that B lost health, while on B's machine's authority data, A's attack missed, and B maintained full health. However, B's machine's calculation of full health is based on faulty simulation. Naive approach would simply rubber band A's speculation of B to B's true state. There are two problem, however:

1.A's speculation of B, now accurate, will be 2 ticks behind because B's state patch takes 2 tick to arrive
2.The root cause of this desynchronization is based on B's faulty speculation - addressing the most recent desynchronization by simply rubberbanding makes the game appear illogical to both parties.

Our approach is a variation of rollback. Every tick, the player sends a State Patch packet, which specifies the current true state of the local player at the current tick rate. This state patch packet will arrive to the other client around 2-3 ticks later. The following protocol ensures proper state adjustment if there are speculation errors on remote clients. This is based on the previous scenario

1.At tick 5, remote client B receives state correction packet of client A for tick 3
   a.Packet content: true position of A is Xa = 1 at tick 3
2.Player B compare the packet content with stored speculation for Player A at tick 3
   a.if they are the same, do nothing
   b.if they are not the same(as in this scenario), proceed with the algorithm

3.Player B waits for all player (player C, player D) state synchronization packets for tick 3 to arrive. While player B is waiting, it is also actively speculating.
4.Player B reverts game state for player A on its local machine to tick 3 based on authority value given by player A. Player B then instantaneously Resynchronize all the input from the point on.
a.Resynchronize tick 3 with all input from tick 3 to 5. this means now player 3 recognize the hit, the hit gets registered, and two players are in sync

Because this technique traces the root of desynchronization and Resynchronize, the first State Patch after desynchronization should correct all of the following desynchronization on the client receiving the state patch, thus matching it with the sender [10]. In the example above, player B receives a state patch for player A's position. After resimulation, player B's local authority state of health now matches player A's speculation of player B's health because now with the correct position for tick 3 for player A, player B can detect and register the attack from player A. In this technique, the local authority state is not necessarily always the true state of a player across the network. In the example, player B falsely calculates local authority state of health = 1, because it did not register a hit due to faulty speculation of A's location being 0.9 instead of 1. In its essence, local authority state may be subject to change if it is calculated based on faulty speculation of remote clients' authority state from previous ticks.

However, this presents a problem if a state patch is being sent on every tick. It could be that before B has corrected its state, B sends a faulty State Patch to A.

See a possible scenario

1.at tick 4, player B's local authority state for health is 1.
a.this is because player B had a wrong speculation of A's position at tick 3 and thus did not register the hit
b.player B sends state patch, stating that its local authority state is {hB = 1}
2.at tick 5, player B Resynchronize, changing its health to 0
a.player A's state patch packet arrives at tick 5, stating that player A's authority state at tick 3 is Xa = 1. Player B detects desync as its speculation of A's state at tick 3 is Xa = 0.9
b.player B sends state patch again, now local authority state is {hB = 0}(which is the same as player A's speculation)
3.at tick 6, player B's faulty state patch{hB= 1}() for tick 4 arrives on player A
a.player A Resynchronize
i.now at tick 6, player A change player B's speculative state to {hB=1} from {hB=0}
4. at tick 7, player B's packet for tick 5 arrives
a. player A Resynchronize
i. speculation for B changes back to {hB = 0}

There are several unnecessary resimulations. Beside costing resources, it could lead to further smaller desync. Player A could have non-deterministic results on each simulation, for example. Or player A could calculate a different local state because of a faulty desync, and send out that state, which will induce a chain reaction across all clients, letting them Resynchronize over and over for every few tick intervals. Although this is not an infinite loop(as resimulation only happens at most once per tick), the true state for each client may be lost in this constant resimulation. Theoretically, since all players only broadcast their own state, when the tick loop stop, and the resimulation for each client's state packet of last tick finish, all players would ultimately be resynchronized again, with the chain of events sorted back all the way to the

original desync. But this is highly improbable as the chain of resimulation expands exponentially with more players. Evidently, having all clients constantly sending state packets is not an efficient method and may result in highly complex resimulation based on delayed faulty state.

To ensure no client will receive faulty state packets when a remote client is desynced, we introduced a technique called Resynchronizing At Root. The technique guarantees that if a state patch causes a resimulation, it is at the very root of the desynchronization, and ignores all desynchronization that happened after the root because they will be automatically corrected by resimulation. In the example above, at tick 5, when player A's state patch for tick 3 arrives, player B would have a resimulation of tick 3 to correct the faulty speculation of Xa(from 0.9 to 1), which results in player B registering the attack input (scheduled for tick 4). On the other hand, at tick 6, when player B's faulty state synchronization packet for tick 4 arrive(falsely reporting that B's health Hb = 1 because of previous desync of Xa = 0.9), player A would know it is not the root of the desynchronization, and thus not trigger a resimulation. This is achieved by delaying the execution of a resimulation(from detecting difference between speculative state of remote player and state patch of remote player), until it is confirmed that all the previous tick up till the current tick is synchronized.

The protocol is as following

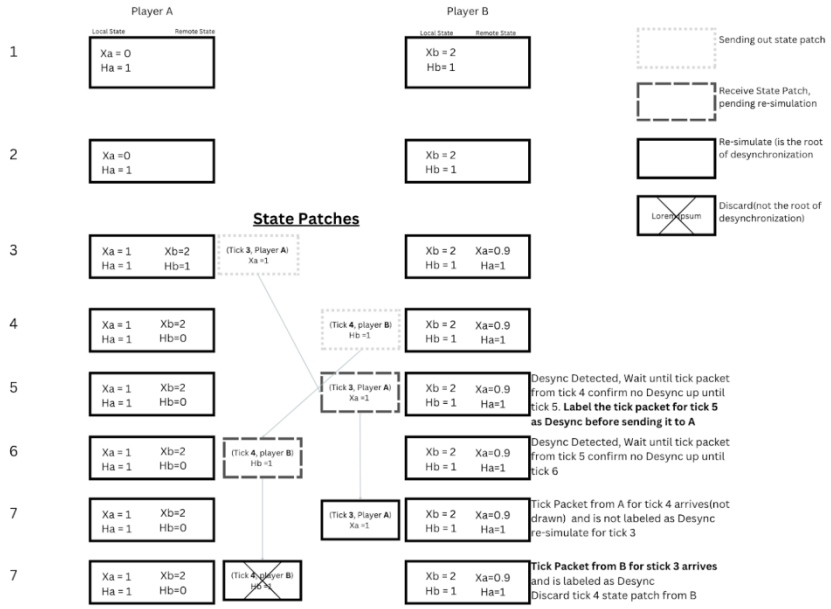


Figure 5. Protocol

Note: tick packet is sent for every frame and thus not drawn.

Note: desync label from player A's tick packet represents player A has a faulty speculation. Therefore, if player B receives a state patch from player A that shows player B has a faulty speculation, but then receives another tick packet indicating player A is already desynchronized, player B does not Resynchronize.

1.At each tick X, the local player A may receive a state patch(as part of tick packet) for tick Y from a remote player

a.tick Y is usually one or two tick behind tick X, such as X-2, depending how many tick it takes for a packet to arrive

2.local player A detect state patch for tick Y is different from its speculation for the past tick Y

a.does nothing. It is not determined whether or not tick Y is the root of desynchronization

b.mark tick packet for tick X with data {desync: tick Y}, this will arrive at tick Z.

c.if there are no difference between state patch and speculation, mark tick packet for tick X with data {synced:tick Y}

3.As the remote player B performs the exact same algorithm, player A is expected to eventually receive a tick packet containing the mark {synced: tick (Y-1)} or {desynced: tick (Y-1)}.

a.if {synced: tick (Y-1)}, Resynchronize as it means there is no desynchronization up till tick 3 thus it is the root cause of the desynchronization.

b.{desynced: tick (Y-1)}, does not Resynchronize – because player A's speculation may as well be the true state of the B, once B corrects its faulty speculation of A at tick Y-1 and Resynchronize up till tick Y.

If the local player receives a state patch that is different from speculation, we include in the current tick's tick packet an indicator that a previous tick is desynchronized. For example, if player B has a faulty speculation for player A's state at tick 3, and discover it at tick 5 after receiving tick 3 state patch from player A, player B would add the data along the line of "desync at tick 3" in its tick packet to indicate that it might Resynchronize because of a previous desynchronization. The flag Desync is for tick 3. It marks the tick in which the desynchronization happens, not the tick in which desynchronization is detected. In the previous example, the flag would specify there is a desync at tick 3 {Desync: 3}, while being packaged in the tick packet for tick 5. The flag allows player A to know that later state patches, like patch for tick 4 from player B, are not the root desynchronization and thus should not cause a resimulation. On the other hand, if player A received a state patch from player B that does not have desynchronized data, it will send out a tick packet with data {synchronized:3}, broadcasting that up till tick 3, there is no desynchronization. When player B receives this tick packet, it could safely assume that any desynchronization immediately after tick 3 is the root desynchronization. In the fringe case in which both players have speculation error at tick 3 for each other, it would also be root traced and corrected respectively.

Module 5, Implementation Details

The implementation will target browsers. Here we list down some specific key implementation details that make the framework efficient.

I, Web-Worker Multithreading

Main thread of a webpage in the browser is usually tied to the tab. This means when a tab goes idle, the main thread is subjected to being paused. Because of this, we implemented our framework in Web-Worker. Web-Worker provide the several advantages

1.separate from UI and game, thus will not be blocked by them
2.Can run on background, thus allowing update tick to run continuously

This median article https://medium.com/teads-engineering/the-most-accurate-way-to-schedule-a-function-in-a-web-browser-eadcd164da12

tested several different update loop methods in javascript [13].

- setInterval in main thread
- setInterval in web worker
- requestAnimation frame

It concludes that setInterval in web workers provides the most accurate delay based on the specified interval.

II, Managing UDP Packets

WebRTC's data channel API allows for both TCP and UDP packets to be sent across. However, we opted for UDP packet because there are several problems with TCP packets

- Ordered- causing jitter
  - ◇ packets arrive in irregular intervals due to network jitter. By forcing packets to arrive on the recipient's machine in order, TCP exacerbates the jittery effect.
- Reliable-causing lag
  - ◇ Because TCP waits for dropped packet to be resending before sending any new packet, it exacerbate the network jitter and also causing unnecessary lag

In situations where a continuous stream of data is favored over reliability, such as movie streaming, real-time fast paced games, UDP is the better choice because it aims to make the data arrive as fast as possible [15]. But raw UDP packets are unsuitable for our needs. Because our protocol does not require determinism, it is crucial for input packet and state patch for each tick to arrive on the remote player. Even one missed input could break both the simulation and the Random Authority Shuffle protocol. Therefore we packet previous ticks data into the current tick, until it is confirmed that they are received on a specific clients machine

1.At tick X, player A sends a tick packet to player B
   a.The packet contains input and state patches for tick X, tick X-1, and tick X-2.
   b.It also indicate that player A has received player B's packet up to tick X-1
2.Player B receives the tick packet (sent from step 1) at tick X +2.
   a.because the packet indicate player A has received player B's input and state up to tick X-1, player B only send a tick packet containing input and state of tick X, tick X+1, and tick X+2
   b.player B also indicate in its packet that it has received player A's packet up to tick X

This method allows for packets to arrive as fast as possible, while eliminating the unreliability of UDP. Because input and state are labeled for each tick, the packets can arrive in various order and still result in the same effect.

III, Input Delay

Our Protocol implements a variation of input delay. This paper will not dive into the detail of input delay as it has been thoroughly discussed in other literature. In our implementation of input delay, because our tick rate contains a 66ms interval, we allow for multiple input in order. This is because our tick runs at 15 ticks/second, while most browser updates at 60 frames/second. Therefore there could be multiple keyboard registers on the game engine in one tick(for example, a player pressing down W key for 60ms, resulting in 3 movement input).

Module 6, Discussion

In this paper, we introduced a framework for browser specific peer to peer multiplayer. Theoretically, the idea behind the protocol can work for desktop applications as well. But desktop applications usually would not need it as it has more security and anti-cheat options, such as anti-cheat software that detects RAM modification [14].

For security, we introduced two novel protocols to ensure no player can manipulate data to gain an unfair advantage: Random Authority Shuffle and Speculation Based State Verification. Random Authority Shuffle is a novel concept based on the core principle of using probability to reduce the success of targeted cheating by assigning each player to manage the authority of an unpredictable, random remote player. This way, no player can continuously target a specific player and any fine tuned cheating will likely fail due to the luck that it requires for cheater to be assigned with its target at the right time whereas any drastic cheating that significantly alters the state of remote player will be caught with Speculation Based State Verification. Speculation Based State Verification is a modified version of rule-based state verification, where if there is a significant discrepancy from data authority holder, players can convene and determine democratically if the data authority holder manipulated the data by simply checking if it differs from all other clients' simulation.

For Synchronization, we introduced a novel technique called Resynchronizing At Root , where it prevents unnecessarily re-simulation by ensuring that a re-simulation only happens if the tick to be Resynchronized is the root of desynchronization. Because re-simulation simulates the tick with all the input after it in order, it will remove the desynchronization chain that is caused by the root. We also used time based lockstep to ensure ticks are synchronized across all machines.

Module 7, Future Research

Although this paper addresses the state manipulation cheat from players, it is unsuitable for games that require secret data. We did not address how a client could hide private, strategic data while proving it is not cheating to other clients.For example, in a situation where a player wishes to hide its data(such as poker), how could a remote client make sure that player does not cheat by manipulating the private data? This could be solved with a Zero-Knowledge-Proof protocol.

ACKNOWLEDGEMENTS

I would like to express my gratitude toward Dr.Sun (Cal Poly Pomona) for his generous support and encouragement for this paper and my research into Cryptography.

REFERENCES

[1] Y. Dodis and A. Yampolskiy, "A verifiable random function with shortproofs and keys," in Public Key Cryptography (PKC'05). Springer,2005, pp. 416–431.
[2] D. Chaum, "Blind signatures for untraceable payments," in Advances incryptology. Springer, 1983, pp. 199–203
[3] El Saddik, Abdulmotaleb & Dufour, A.. (2003). Peer-to-peer suitability for collaborative multiplayer games. 101- 107. 10.1109/DISRTA.2003.1243003.
[4] Denning, T., Lerner, A., Shostack, A., & Kohno, T. (2013). Control-Alt-Hack: the design and evaluation of a card game for computer security awareness and education. Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security.
[5] Hincapié-Ramos, Juan David & Henao, Andres. (2007). P2P Game Network Framework - Communications and Context Framework for Building P2P Multiplayer Games (Intro).

[6] Blanchet, Bruno. "Security protocol verification: Symbolic and computational models." International Conference on Principles of Security and Trust. Springer, Berlin, Heidelberg, 2012.
[7] El Rhalibi, Abdennour, Madjid Merabti, and Yuanyuan Shen. "Aoim in peer-to-peer multiplayer online games." Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology. 2006.
[8] Kamara, Seny, and Kristin Lauter. "Cryptographic cloud storage." International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2010. G., and Richard P. Mislan. "Mobile device analysis." Small scale digital device forensics journal 2.1 (2008): 1-16.
[9] Wortman, Paul, et al. "P2M- based security model: security enhancement using combined PUF and PRNG models for authenticating consumer electronic devices." IET Computers & Digital Techniques 12.6 (2018): 289-296.
[10] Pfurtscheller, Gert, and FH Lopes Da Silva. "Event-related EEG/MEG synchronization and desynchronization: basic principles." Clinical neurophysiology 110.11 (1999): 1842-1857.
[11] Sredojev, Branislav, Dragan Samardzija, and Dragan Posarac. "WebRTC technology overview and signaling solution design and implementation." 2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO). IEEE, 2015.
[12] Johnston, Alan, John Yoakum, and Kundan Singh. "Taking on webRTC in an enterprise." IEEE Communications Magazine 51.4 (2013): 48-54.
[13] Jensen, Simon Holm, Anders Møller, and Peter Thiemann. "Type analysis for JavaScript." International Static Analysis Symposium. Springer, Berlin, Heidelberg, 2009.
[14] Powers, Scott, et al. "RAM, a gene of yeast required for a functional modification of RAS proteins and for production of mating pheromone a-factor." Cell 47.3 (1986): 413-422..
[15] Gu, Yunhong, and Robert L. Grossman. "UDT: UDP-based data transfer for high-speed wide area networks." Computer Networks 51.7 (2007): 1777-1799.

AUTHOR

Haochen Han

Haochen Han is a student living in Orange County with a passion for Computer Science, Networking Programming, and Cryptography. Author of Treasure of Forrealm, a multiplayer 2D PVP party game.

ENTERPRISE MODEL LIBRARY FOR BUSINESS-IT-ALIGNMENT

Peter Hillmann, Diana Schnell, Harald Hagel and Andreas Karcher

Department of Computer Science,
Universität der Bundeswehr, Munich, Germany

ABSTRACT

The knowledge of the world is passed on through libraries. Accordingly, domain expertise and experiences should also be transferred within an enterprise by a knowledge base. Therefore, models are an established medium to describe good practices for complex systems, processes, and interconnections. However, there is no structured and detailed approach for a design of an enterprise model library. The objective of this work is the reference architecture of a repository for models with function of reuse. It includes the design of the data structure for filing, the processes for administration and possibilities for usage. Our approach enables consistent mapping of requirements into models via meta-data attributes. Furthermore, the adaptation of reference architectures in specific use cases as well as a reconciliation of interrelationships is enabled. A case study with industry demonstrates the practical benefits of reusing work already done. It provides an organization with systematic access to specifications, standards and guidelines. Thus, further development is accelerated and supported in a structured manner, while complexity remains controllable. The presented approach enriches various enterprise architecture frameworks. It provides benefits for development based on models.

KEYWORDS

Enterprise Architecture, Model Library, Business-IT-Alignment, Reference Architecture, Enterprise Repository for reusable Models.

1. INTRODUCTION

Over the years, companies and their employees aggregate a lot of expertise and experience. This ranges from structured processes to technical skills. Explicit expert knowledge can be well mapped by documentation. In contrast, implicit experience is difficult to explain or write down [1, 2, 3]. Furthermore, the totality of all information with its variety and diversity can lead to information overload and in-transparency. This problem is addressed by the approaches of model formation and *Enterprise Architecture* (EA), see Figure 1. Thereby, the central element is the Enterprise Architecture Library (EAL), in which the information is structured and manageable. Despite capability-based frameworks like *The Open Group Architecture Framework* (TOGAF), *IT Infrastructure Library* (ITIL), and *NATO Architecture Framework* (NAF), there is no practical template or content structure of an EAL. According to the definition, an EAL organizes models with reference character and user application according to uniform aspects [4]. It allows a systematic access on the knowledge of the business specific perspectives with e. g. value chain, processes, methods, applications and resources. In addition, a holistic EA landscape includes a comprehensive method and strategic orientation to the application of the models.

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 157-170, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122011

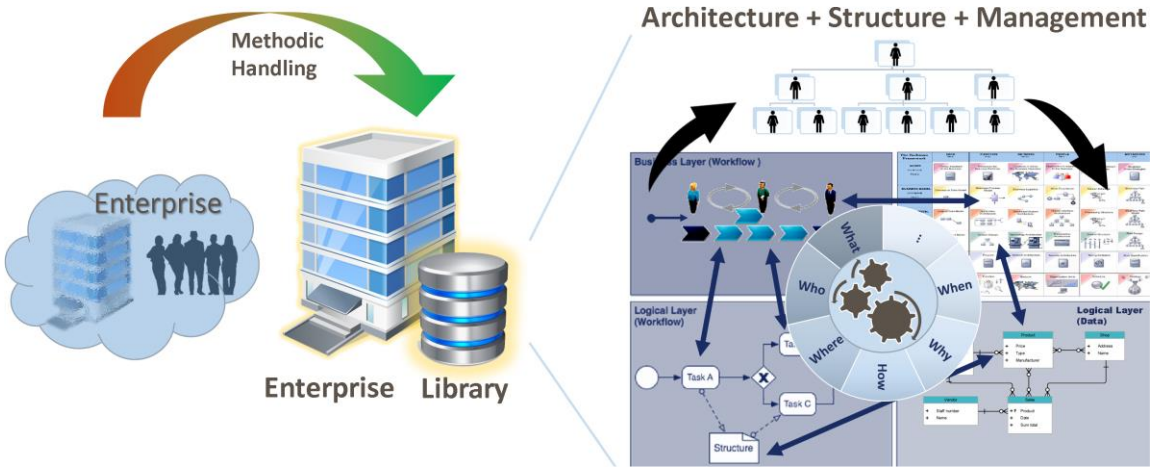


Figure 1. Reference design for an EAL of enterprise models and their maintenance

Thereby, the facts of an enterprise are illustrated via models. It is an adequate medium to describe *good practices* for complex systems, processes, and interconnections. Models provide a clear overview of interrelationships and allows them to be managed. Furthermore, the models have the goal of representing responsibilities, established methods and valid solutions. Thus, the broad knowledge of various company divisions become structured and made available for communication as well as reuse. Considering, especially reference models provide support with a recommendation character [5]. In the design process, best practices such as ITIL, COBIT, or IT4IT frameworks also serve as a guide for the modeler. These frameworks can be part of a company's own repertoire of reference materials. Therefore, a catalog organizes this information according to be defined criteria. According to Kiehl [6], the use of reference models shortens the development times of concrete models, while their model quality increases and the costs of modeling are reduced. Furthermore, it is hardly possible to feed back knowledge gained during the practical usage of a model [7].

To drive an enterprise forward, well-functioning Business-IT-Alignment is indispensable today. A central EAL of business models represents the possibility to counteract the information in-transparency and discrepancy. Currently, there is no structured and detailed approach for the design of an EA model repository. In this paper, an EAL for models is developed to capture the wealth of information about an enterprise. It includes the reference design of the data structure for managing, processes for administration and possibilities for usage. In addition, our approach supports the maintenance of EA models. Thereby, relationships between entries, models, and architecture building blocks are shown. The EAL allows a targeted application and re-usability of already coordinated knowledge. To include an improvement possibility, the EAL supports the process with additional meta information and feedback options to an entry. So, it forms a knowledge base for the enterprise and their value streams. The structured preparation of established EA models is essential for it and influences the retrieve-ability of the generally valid solutions. It allows systematic access to existing knowledge, especially for new employees, special cases, and audits. The operation of an EAL supports the strategic management of the business. Generally valid solutions in the corporate environment are accordingly reusable to avoid work already done.

For clarification, we see a repository in this context as an unstructured collection of EA models. In contrast, an EAL provides structured access to content via a catalog as well as further functionality for application and usage.

This paper is structured as follows: In Section 2 we describe a typical scenario and the requirements for a reference EAL. Section 3 provides an overview of the current state of the art in that application area. The main part in Section 4 describes our concept of a reference EAL for models. Subsequently, we elaborate first experiences on prototypes and evaluate fundamental properties of the presented system in Section 5. The last section summarizes our work and provides an outlook.

2. SCENARIO AND REQUIREMENTS

The need for an easy-to-use EAL is illustrated by the following typical scenario. *YOUrRobots* is a mid-sized company and offers autonomous robots for the maintenance of green areas. TOGAF is already established to cyclic improve the interaction between IT infrastructure and the implementation of business goals. According to model-driven development, the technical models and documents can be reused as reference other business areas with the knowledge already gained. The methods and processes are harmonized for complexity reduction and integrated Business-IT-Alignment.

To increase customer satisfaction, new robots are to be equipped with additional services by plugin extensions. The smartification includes remote administration for planning and control as well as online software updates. In addition, it offers a regular evaluation of usage data for maintenance depending on actual wear. Therefore, the company will operate several cloud services for remote administration by customers. In order to reduce the maintenance effort and the operational costs for setting up services, their structure should be identical. Appropriate referral models offer flexibility and support integration into existing structures. Also, the management requires support in the IT alignment of the business units. Therefore, the introduction of an EAL is pushed with the re-usability of the adapted solutions and strategic management.

The challenge is to provide the knowledge gained in a structured way, especially for new employed systems engineers. In addition, the aligned relationships between reference approaches and adapted solutions need to be supported. For cooperation of different departments, the information has to be interconnected with adequate interfaces. The diverse structures and processes require various types of data, from simple descriptions to visual models and complex specifications.

Generally, most important requirements for an EAL and its contents are the following:

- **R1**: The EAL has to provide sufficient documentation of model purpose, model context, problem structure and design aspects, so that communication is improved and interpretation is limited.
- **R2**: The EAL shall support storage and re-usability of architecture components of different levels to simplify the development of new models.
- **R3**: The approach shall provide discoverability via typology and feature search, factual logical access hierarchies, and classification by feature description.
- **R4**: The EAL entries shall have a recommendation character and therefore be configurable, adaptable, instantiable and extensible.
- **R5**: The approach shall support linkage between versions and variants of EA components and models built on top from them.

Based on the mentioned requirements, we consider the following research questions to be important:

- **Q1**: How can an EAL be realized to support a sustainable approach through reuse of architecture components?
- **Q2**: What properties must an EA component have in order to have reference character and actually be reusable?
- **Q3**: How can interrelationships and dependencies between components be made available to support and guide a modeler?
- **Q4**: What is the process for integrated application of models via library throughout a life-cycle?

3. RELATED WORK

The requirement for reference modeling has existed since 1980 [8]. Due to the diversity, the need for a repository has also arisen. The term reference comes from etymological and means recommendation. According to this, a reference model has recommendation character or it is referenced [9].

According to ISO/IEC 42020 definition, the architecture repository hosts the baselines of architecture elements produced or updated by architects. It includes different kinds of architectures models and architecture elements like patterns and building blocks. It is a place where work products and the associated information items can be stored for preservation and retrieval. In alignment with the TOGAF enterprise continuum, it describes elements with increasing detail and specialization.

Various references to an architecture repository can be found in the literature. In the context of the EA Framework TOGAF, the consortium *The Open Group* roughly describes an EA repository with its components [10]. It contains out of six main classes of architectural information: Architecture Meta-model, Architecture Capability, Architecture Landscape, Standards Information Base, Governance Log, and Reference Library. In addition, the concept includes connections to a *Solution and Requirements Repository* for a suitable interplay in the application domain. Based on the mere existence of such a repository, its added value and usefulness are made clear. The main comparative component in relation to our scenario is the *Reference Library*. However, the description is an empty shell with a superficial description of the possible content. A detailed description is missing regarding the structuring, process and procedure of application. According to given templates of *The Open Group*, only a folder structure on file basis is suggested [11]. ArchiMate also recommends a file-based storage in the scenario for banking industry [12]. This is also accompanied by a specific model registry for logistic [13].

The newer approach of the NATO Architecture Framework (NAF) describes a comparable reference library with the Architecture Landscape [14]. This contains a basic set of assets that can be reused by architects. Lang [15] distinguishes here three kinds of model libraries, in order to separate the conceptually different reference models. It is differentiated into the following libraries: domain-neutral, domain-specific, and enterprise-specific. A detailed description for a repository is not given.

For a corresponding classification and reuse of models, there are first approaches of a taxonomy according to application areas [16]. Another classification is based on structural and model aspects [17]. Both approaches are on a theoretical level. In addition, the rough methodical procedure for reference modeling are given with the top-down or bottom-up strategies [18, 19]. Besides this, an abstract infrastructure for a model management system is illustrated. Furthermore, it should be noted that there are different types of reference modeling [20, 21]: Analogy Construct, Specialization, Aggregation, Instantiation, and Configuration. This results in

three possibilities of construction for the application of models in the specific case: variants, versions, and re-construction.

In the ITIL approach, several databases with similar goals to a repository are described [22]. Their Service Knowledge Management System is intended to centrally manage the essential assets of a company. Nevertheless, no detailed description is provided.

In addition, the Gamma et al. [23] design patterns provide detailed modeling and descriptions of recommended solutions to typical problems in information technology. However, these do not relate to enterprise design and no repository structure is given.

Approaches for EA design can be found in the patterns of Perroud and Inversini [24]. Here, some patterns are described on the basis of an individual meta-model. Possible categorizations are provided for software and construction pattern, which can be partially adapted.

The comprehensive approach to a repository is provided by the German University of Saarbrücken [25]. It classifies many different models. Nevertheless, their reference model catalog focus on a tabular without individual adaptability and operational solution. According to our requirements it lacks structural coherence, graphical representation and practical usability.

The IT4IT framework clarifies adaptable information objects via the reference architecture [26]. According to the EA Functional Component data object, only the three attributes *ID*, *Component*, and *Diagram* are suggested.

In summary, there is so far no practical usable approach of an EAL for the management and usage of models in long-term.

4. CONCEPT OF A LIBRARY FOR ENTERPRISE MODELS

Our holistic approach to an EAL addresses these topics in detail with a domain-independent description of such a system. Figure 2 provides an overview of the main components. The focus is on the information and application management.

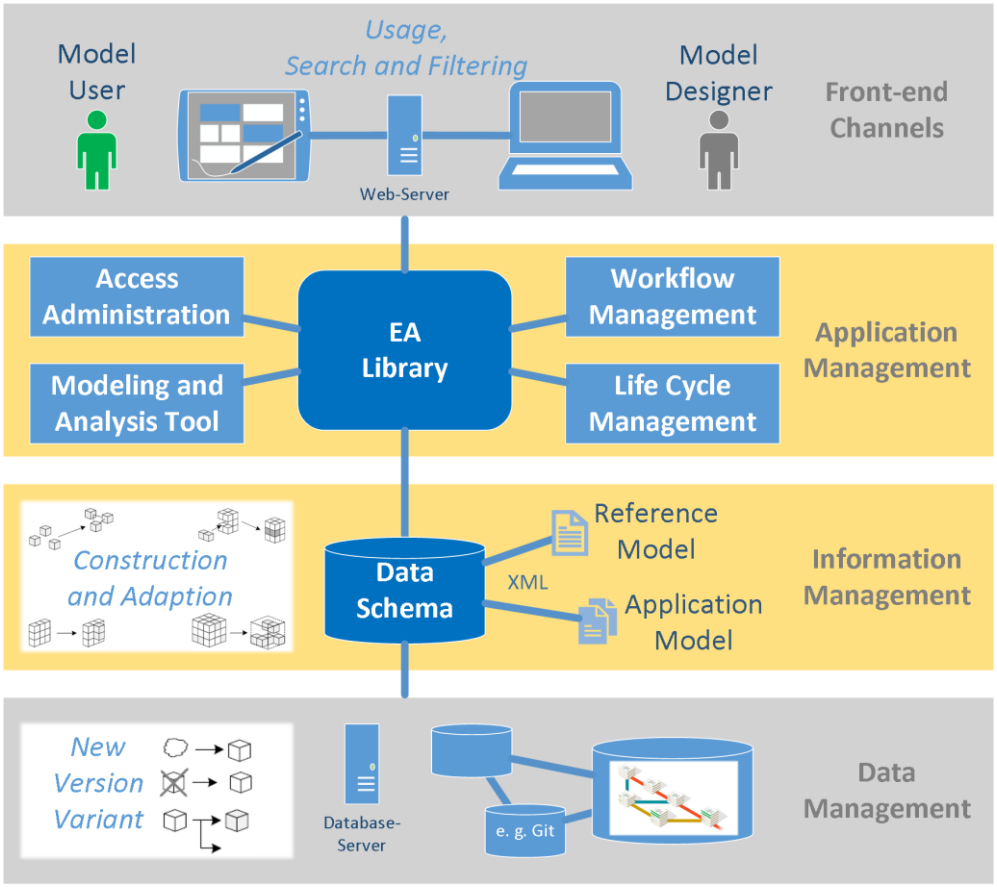


Figure 2. Overview of the Enterprise Architecture Library.

At the bottom line is the data management based on a vault for linking of versions and variants, see Section 4.2. It is setup on a database to manage the entries with meta-data, models, and optional content as well as access rules. A distributed file system is suitable here, which enables version and variant management with role-based access control, e.g. Git. This infrastructure supports collaborative work with non-linear workflows. Based on this, the structure for orderly information management is specified. Here, the storage of entries takes place in the form of a catalog, see Section 4.1. This includes the models with reference and application character. Furthermore, it offers architecture building blocks, which can be used for new constructions, further development, and flexible adaptation. For this, we use the standardized data format XML (based on *The Open Group ArchiMate Exchange File Format*).

The core of the EAL system lies in the application management. This consists mainly of four parts. The access administration manages the users with their roles and rights. It controls the access to the system and limits the modification of entries to responsive role. Here, the organizational structure can be used accordingly. The modeling and analysis tools support the designer during the construction and modeling. These depend on the individual needs and use case of the modelers and the company. Here, common meta-models with the favored tools are applied. In particular, building block-oriented modeling is an essential feature with regard to reusability. It also enables adaptation to a specific problem in a domain. For this purpose, the modeler must pay attention to clearly distinguishable building blocks. These must not be designed too specifically, especially in reference modeling, and must be provided with clearly defined interfaces.

The flow of information is controlled by means of workflow management, see Section 4.3. A process engine is used for this purpose, which provides defined processes for modeling, application and usage. The workflows are supported semi-automatically by stringently following the process model and generating attributes of the entries. This also allows company-specific workflows to be mapped, such as change proposals. In connection with workflow management, life-cycle management is carried out for the entries and models. Models of different levels and perspectives can be entered via the targeted attribution in the library. An entry in the library follows a life-cycle, whereas the status is traced via specific attributes. The objective is a single source of information, whereas data linkage is enabled. Thus, separate directories for different continuum's are not mandatory in relation to model's abstraction level or purpose.

On the top level, the system is accessible via a web front-end. This guarantees an interaction across all devices in any situation. In summary, this creates a system for documenting and processing the knowledge that exists in the company in a structured manner. This counteracts the current obstacle of limited communication and makes the scope for interpretation manageable.

4.1. Information and Data Management

The following attributes are specified to meet the requirements for a structured storage of models and other information. The integration of a model in the library contains generated, mandatory, and optional parts to be filled.

The following attributes are **automatically generated**:

- **ID**: Unique identifier.
- **Version**: Ascending Number with creation date.
- **Status**: Current phase in the life-cycle with date of change.
- **Complexity**: Generated rating for syntax comprehensiveness.
- **Connectivity**: Generated score for semantic understanding.

The ID is mandatory for unique identification and referencing, equivalent to IT4IT. The tracking of improvements is done via appropriate versioning with sequential succession. The complexity and connectivity express how difficult the model is to capture and applicable. So far, there is no measurable metric with regard to "good" models. Experience has shown that a 3-level rating for both yardsticks is enough in practice, based on the number of elements and connections. By analogy for the comprehensibility of sentences, a corresponding measure is adapted here [27, 28]. For complexity, we rate models based on the sum of elements and connections. A model is easy with less than 20 components, moderate from 20-40 components, and complex with more than 40 components. For connectivity, we score models based on the average number of connections of an element. We see a model as simple with a score less than 2, average from 2-3, and difficult over 3. If a model is too complex or difficult, then it is suggested to subdivide the model into several sub-models or to structure it hierarchically [29].

The following **core attributes** are the key information for any entry to be included in the library:

- **Title**: Short and concise name of the model for the first relevance decision.
- **Category**: Kind of model with regard to structured reuse according to a taxonomy.
- **Layer**: Recommended level of application based on an enterprise hierarchy.
- **Abstract**: Description of the data set including pattern intent:
  context, problem statement, solution, and results.
- **Keywords**: Supports finding according to a feature list based on taxonomy.

- **Responsible Authors**: Central point of contact for questions and improvements.
- **Model**: Main content of an entry, which provides a solution approach for an use case.

The category represents an abstract high-level tag and is independent of the application area. It focuses on the enterprise continuum with domain-neutral, domain-specific, and company-specific aspects. The kind of the model is described with regard to possible reuse, e. g. building block, design pattern or application model. Especially, the design of reference models and generic building blocks require clear distinctiveness from each other and well-defined interfaces. The assignment to a layer addresses the subject of interest and describes the degree of abstraction as well as the level of detail. The specification of possible layers is oriented on the modeling framework, e.g. ArchiMate from the top with strategy to the bottom with physical systems. Typical areas are, e.g. Business Process, Application Stack, or Support Information-flow. The abstract describes the essential addressed contents of the model in the form of a succinct text. The keywords support the targeted retrieval via search and filtering. The specification of possible keywords is based on ontology, thesaurus, or glossary. According to the rights and roles model, there is a responsible author for the data set. This is the central point of contact for questions. The person is responsible for continuous improvement as well as maintenance and care. The essential content of an entry is given by the model, regardless of whether it is a reference, a building block, or an application. Based on a meta model, a solution approach is shown for an issue. This is not only used for documentation, it is mainly intended for reuse and further development.

Beside the key attributes, we suggest to added the following **optional information**:

- **Application** Context: Explanation of the circumstances when and if it should be used.
- **Stakeholder**: Matrix of important people to be addressed.
- **Capabilities**: Textual description of how to use the model and what skills must be met.
- **Limitations**: Listing of restrictions, requirements and possible challenges in application.
- **Dependencies**: References and connections to other facts, esp. for future developments.
- **Bricks**: Index of applied building blocks in this entry for traceability.
- **Variants**: Links to domain- or case-dependent modification or adaption; parallel existence.
- **Example**: Description of a concrete use case with application context for typical usage.
- **References**: List of sources for further information.

4.2. Structure and Library Administration

A vault is provided for structured storage of the hierarchical data. Therefore, the generic approach of a *Product Data Management Enabler* is adapted [30]. The Figure 3 shows the generic data schema of the back-end system. On the top level of the left part, the initial entry is created by means of the data element *Entry Master*. This describes the type of information that is stored in the library. It includes the automatically generated information and the obligatory meta-data. These are intended to describe management information and offer a flexible reuse. Both versions and variants are attached to the *Entry Master*. The actual data is stored under the *Entry Data* for a specific variant in a particular version. By definition, these describe the content and structure. It includes textual and graphical descriptions, building blocks and associated components. In addition, the data object can be supplemented with further information. Each entry can be assigned optional data and conditions on qualification, effectiveness and possible alternatives.

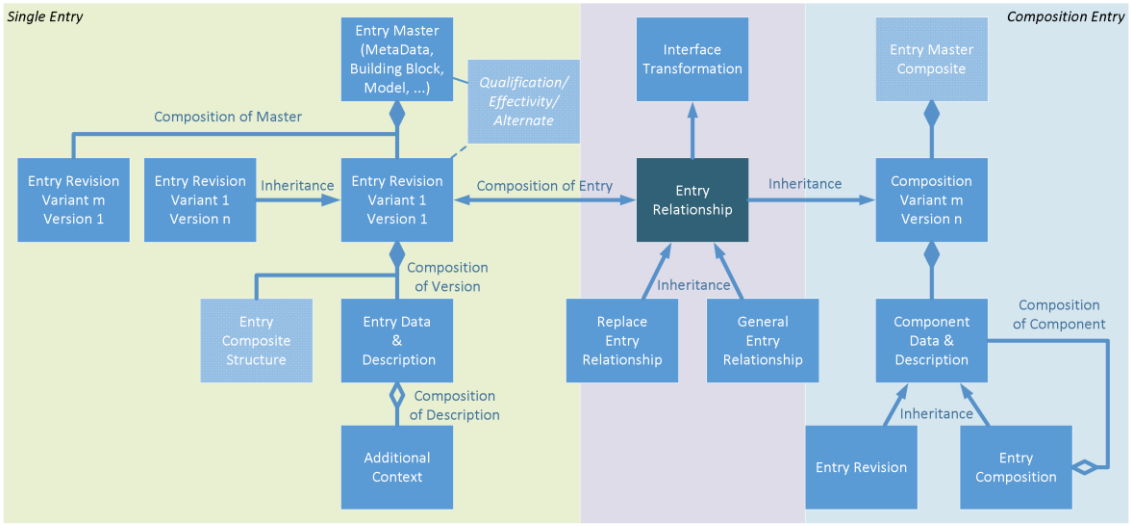


Figure 3. Hierarchical object-based data schema in the back-end of the vault.

For the reuse of models and building blocks a composition of entries is enabled, see right part of Figure 3. Analogous to the data structure of a general entry, the top element is called the master composite. This contains basic meta-data about the content. This is followed by the variant and version. According to the Design Pattern Composite a nesting of the data is made possible. A link to the original data record is established via a relationship so that no redundancies have to be maintained. A relation always refers to a specific variant and concrete version, so that possible incompatibilities in future versions must be checked in advance manually. The relation distinguishes between a general linkage and a replacement behavior. Depending on the compatibility, corresponding interfaces must be transformed.

4.3. Workflow of Usage and Customization

For the general use of the system, it provides an overview page with a grid structure according to aspects and subjects of the viewing angle, following [31]. So, it represents the different domains of the stored models and the content is grouped according to the taxonomic categorization [32]. In addition to a clear entry point, it offers easy accessibility. This architecture landscape already provides information about existing and open areas. In addition, there is the possibility of a search, filtering according to requirements.

In the system, the entries go through a life-cycle, see Figure 4. It can be divided in the two interconnected circle for reference modeling on the left side and application modeling on the right side. If there is not yet a reference for an use case, one can be created. First, requirements of the general scenario are analyzed and determined. Besides this, comparable entries with similar requirements have to be identified. These can serve as architecture building blocks for a composition in order to fulfill the deviating conditions. Thereupon either a new version or variant is to be created. The solution approach is to be described with all its peculiarities. The quality of the model is determined by the expertise of the modeler, the chosen notation and the procedure. Especially for compositions, the links with information on interface connections must be traced. Possible improvements are derived from the result of the analysis or feedback, which cause re-designs. As soon as this is completely stored in the system, the approval for further use takes place via a release. This is expressed accordingly by the attribute status. Subsequently, the model can be implemented and used in the domain. Appropriate monitoring is used to control the solution on the basis of metrics. Depending on different domains, parallel variants can be

instantiated based on this. Adaptations and improvements are traceable via successor relationships with versioning. Here over the status accordingly the relationship is produced. Over a feedback function comments can be left to specific entry. This serves a sustainable further development and is part of the continuous improvement and maintenance strategy. If the references are no more use due to technological progress, these are to be marked as invalid.

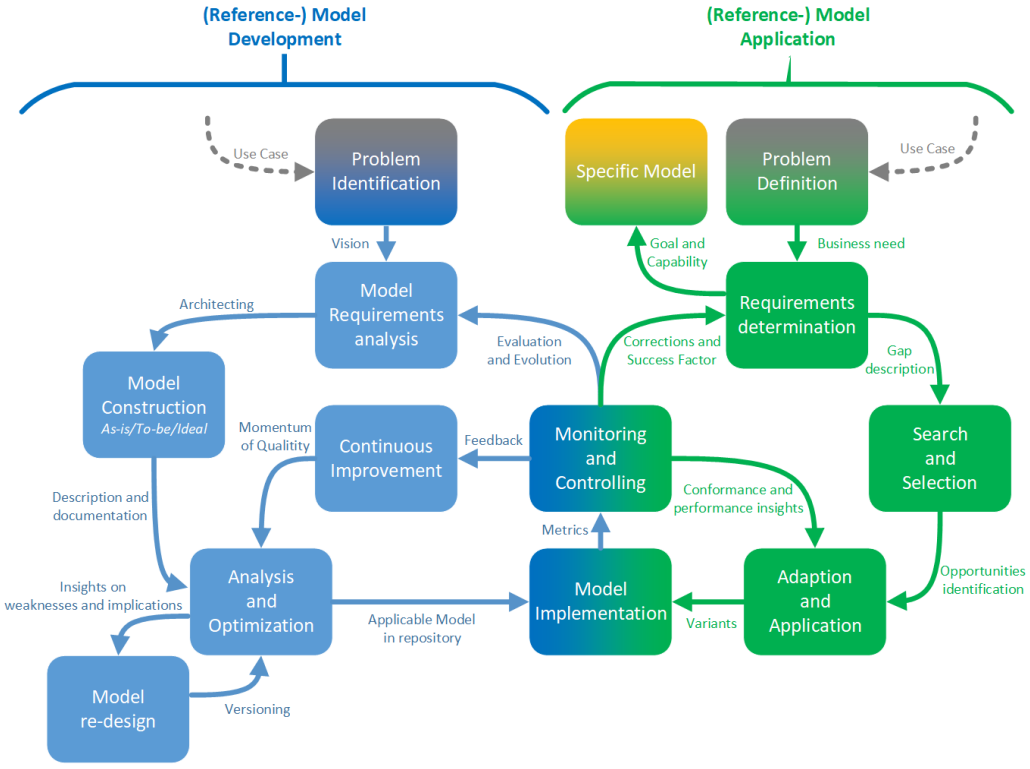


Figure 4. Life-cycle for creating, adapting, and maintaining models in the library.

In order to make the maintenance of models agile, attention must be paid to the documentation of dependencies between individual architectures. If an architecture is revised and modified, this automatically triggers an event to check the linked architectures. This behavior is traced by the attribute status and can create a cascading effect. A supporting system marks the affected areas and informs the responsible authors.

Other methodological suggestions for roles and governance collaboration depend on the specific deployment area. For the frameworks TOGAF and ArchiMate, we refer to the following article [33].

5. Case Study and Preliminary Results

Initial experience was gained in the course of an industrial project. A joint library was created between a smart product provider and an IT service provider with several data centers, in line with the example in Section 2. The companies work in a model-driven manner throughout, so that uniform structures via models can be expected to have a positive impact on operational business.

As test data set, the functional area *Incident Management* is selected according to the frameworks ITIL and IT4IT. It includes: reference models, architecture building blocks, data structures, responsible roles, checklists and many more. These specific models are realized in their own

tools depending on their meta-model, which maintains flexibility. The associated models are used as an example to run through the life-cycle in our library and serve as the basis for evaluation.

The first draft for a realization was based on MS-Excel and MS-Visio to obtain a first impression for a deep analysis. It is widely used and the functional scope is powerful, especially in terms of linking data. The tabular listing corresponds to the idea of Fettke and Loos [7], but quickly reaches its limits in terms of clarity and complexity. Figure 5 provides an impression of these prototypes.

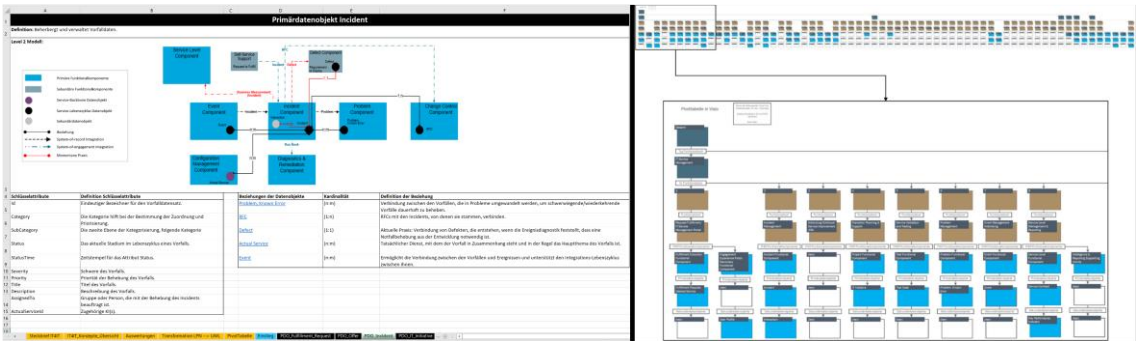


Figure 5. Prototypes in MS-Excel and MS-Visio for first experiences.

In the MS-Excel approach, a worksheet serves as an overview, with each element having an ID for better structuring and linking. The other worksheets are divided according to the attributes *Category* and *Layer*. This structuring allows an extension according to the categories of a domain taxonomy in the width. Here, the flat hierarchy is limited by the software and a missing tree structure. If a model or building block is used in multiple areas, this can be entered accordingly by linking. In this way, all relevant categories can be covered in a consistent and synchronized way. The workflow of usage and life-cycle support is realized manually. The frameworks ITIL and IT4IT categorize the data in over 54 functional units, which we are replicated in our library. Through correlation of models between reference and adapted allows rough statements about the degree of coverage and the fulfillment of specifications. This is done by existence relations with complex expressions in the software suit. Graphical models can be generated automatically in MS-Visio by pivoting the data elements.

Based on the experience of the first two prototypes, an improved software solution for the library as a knowledge base was evaluated. The tools *LexiCan*, *Brain*, and *CherryTree* turned out to be possible alternatives. The implementation in LexiCan can be seen in the Figure 6. It becomes clear that structuring the type layer and domain categories as a tree structure leads to much more clarity.

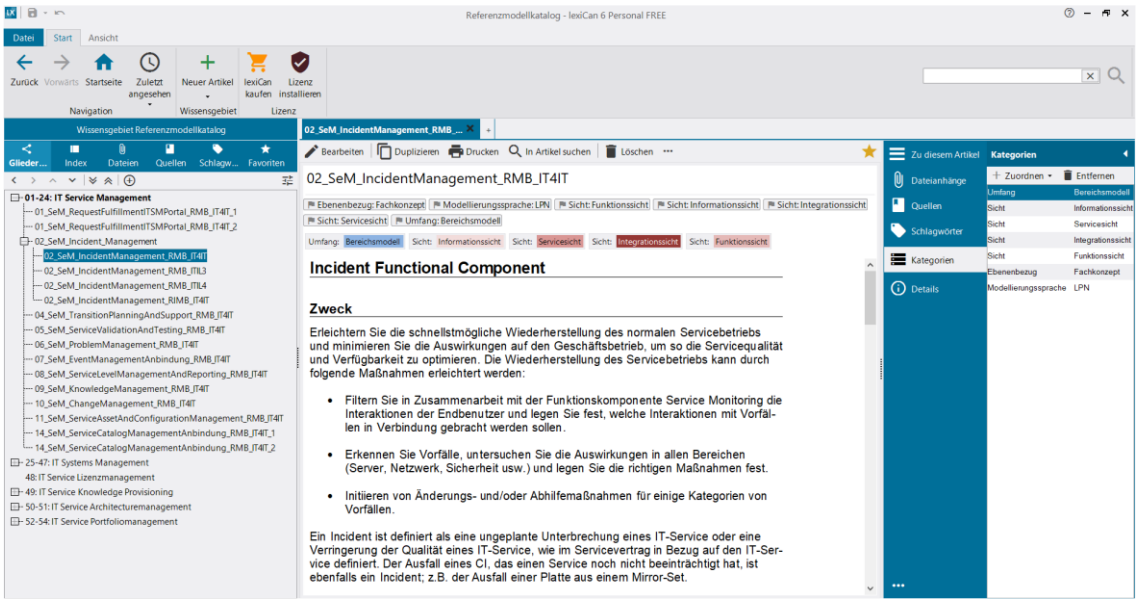


Figure 6. Prototype in the personal Wiki-Software LexiCan.

Nevertheless, also these solutions currently lack possibilities for the direct treatment of models as well as multi-user abilities. In addition, correlations of models and a combination of model components are only indirectly supported. Regardless of this, a clear added value in the daily work with models could already be achieved within the scope of the study.

6. CONCLUSIONS

In summary, the present approach describes essential foundations of a model library as a knowledge base with individual adaptability. The focus is on the management of Business-IT-Alignment with reference and application models, but it is not limited to it. Multiple frameworks like TOGAF, NAF, ITIL and IT4IT are adhered. It extends them in detail over multiple level where these frameworks do not provide information. Our case study shows that considerable added value can be achieved using simple standard software or specialized knowledge management tools. Furthermore, it was analyzed to what extent the mentioned standards actually help a modeler in information modeling.

In the future, we extend the library with further reference models and best practices. A next prototype will allow integrated treatment of models in different meta-models via plugins.

In a further expansion stage, the change management of models across several versions and variants is to be made graphically visible. Based on a standardized XML format for models, deviations can be easily identified and highlighted in visual representations. In addition, we investigate relationships and correlations between reference models and individual instances. This allows detailed statements about the degree of coverage and compatibility. Furthermore, an automated determination of reference models from multiple instance models is envisaged.

ACKNOWLEDGEMENTS

Special thanks to Mr. Lukas Köhler for his support in the development of the basic prototype and his passion for this research.

REFERENCES

[1] Frithjof Weber, M. W. (2016), "Expertise Transfer: A Case Study about Knowledge Retention at Airbus", https://www.researchgate.net/profile/Frithjof Weber/publication/266880066 Expertise Transfer A Case Study about Knowledge Retention at Airbus/links/5721092f08ae5454b230fcab/Expertise-Transfer-A-Case-Study-about-Knowledge-Retention-at-Airbus.pdf, accessed 2022-04-01.

[2] Rensing, C., and L. Després (2017), "Wie lässt sich dasWissen ausscheidender Mitarbeiter bewahren?", http://kommunikation-mittelstand.digital/content/uploads/2017/06/leitfaden-bewahrung-des-wissens.pdf, accessed 2022-04-01.

[3] Cogneon GmbH (2022), "Copedia: Expert Debriefing", https://wiki.cogneon.de/Expert Debriefing, accessed 2022-04-01.

[4] Vázquez, J. M. G., and H.-J. Appelrath (2010), "Energie-RMK - Ein Referenzmodellkatalog für die Energiewirtschaft", *Gesellschaft für Informatik e.V., OFFIS*.

[5] Pöhn, D., and P.Hillmann (2021), "Reference Service Model for Federated Identity Management", *International Conference on Evaluation and Modeling Methods for Systems Analysis and Development*:196–211.

[6] Kiehl, T. (2016), "Wiederverwendung von Geschäftsprozessmodellen durch Referenzmodellierung mit der SOM-Methodik", *Benker T., Jürck C., Wolf M. (eds) Geschäftsprozessorientierte Systementwicklung. Springer Vieweg, Wiesbaden*.

[7] Fettke, P., and P. Loos (2002), "Methoden zur Wiederverwendung von Referenzmodellen: Übersicht und Taxonomie".

[8] Thomas, O. (2005), "Understanding the Term Reference Model in Information Systems Research: History, Literature Analysis and Explanation", *Business Process Management Workshops*.

[9] Delfmann, P. (200), *Adaptive Referenzmodellierung, MethodischeKonzepte zurKonstruktion und Anwendung wiederverwendungsorientierter Informationsmodelle*, Volume 25, Advances in Information Systems and Management Science.

[10] The Open Group (2018), *The Open Group Architecture Framework (TOGAF) - Architecture Repository*, Volume 9.2., https://pubs.opengroup.org/architecture/togaf91-doc/arch/chap41.html, accessed 2022-04-01.

[11] The Open Group (2010), "TOGAF-9-Templates", https://github.com/Skulls-sky/TOGAF-9-Templates, accessed 2022-04-01.

[12] Derde, P., and M. Landkhorst (2020), "Banking Industry Architecture Network (BIAN)", https://www.bian.org/participate/blog/expressing-bian-reference-model-banking-industry-archimate-modeling-language/, accessed 2022-04-01.

[13] Schubel, A., M. Schneider, and C. Seel (2016), "Flexible Prozessstandardisierung: Produktionslogistikgestaltung auf Basis konfigurierbarer Referenzprozessmodelle", ZWF 111.

[14] Architecture Capability Team (2020), *NATO Architecture Framework*, Volume 4.

[15] Lang, K. (1997), *Gestaltung von Geschäftsprozessen mit Geschäftsprozessen*, Gabler Edition Wissenschaft.

[16] Fettke, P., P. Loos, and J. Zwicker (2005), *Business Process Reference Models: Survey and Classification*, 469–483, Business Process Management Workshops.

[17] Schwegmann, A. (1999), *Objektorientierte Referenzmodellierung: Theoretische Grundlagen und praktische Anwendung*, Gabler Edition Wissenschaft.

[18] vom Brocke, J. (2003), "Referenzmodellierung: Gestaltung und Verteilung von Konstruktionsprozessen", *Logos Verlag Berlin* 4.

[19] Fettke, P., and P. Loos (2007), *Reference modeling for business systems analysis*, Hershey PA: Idea Group Pub.

[20] Remme, M. (1997), "Konstruktion von Geschäftsprozessen: Ein modellgestützter Ansatz durch Montage generischer Prozesspartikel", *Gabler, Wiesbaden*.

[21] Becker, J., H. L. Grob, S. Klein, H. Kuchen, U. Müller-Funk, and G. Vossen (2002), *Referenzmodellierung: Methoden - Modelle – Erfahrungen*, Westfälische Wilhelms-Universität Münster.

[22] AXELOS (2019), *Information Technology Infrastructure Library (ITIL)*, The Stationery Office.

[23] Gamma, E., R. Helm, R. E. Johnson, and J. Vlissides (1995), *Design Patterns, Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional Computing.

[24] Perroud, T., and R. Inversini (2013), *Enterprise Architecture Patterns: Practical Solutions for Recurring IT-Architecture Problems*, Springer-Verlag Berlin Heidelberg.
[25] German Research Center for Artificial Intelligence GmbH (2021), "Reference Model Catalogs", Deutsche Forschungsgemeinschaft (DFG), http://rmk.iwi.uni-sb.de/catalog show.php, accessed 2022-04-01.
[26] The Open Group (2015), "IT4IT Reference Architecture", *Opengroup.org,* https://pubs.opengroup.org/it4it/refarch20/chap05.html# Toc431202751, accessed 2022-04-01.
[27] Nirmaldasan (2011), "Longer The Sentence, Greater The Strain", *Vidura, Press Institute of India,* https://strainindex.wordpress.com/2012/04/30/longer-the-sentence-greater-the-strain/, accessed 2022-04-01.
[28] Cohen, J. (1988), "Statistical power analysis for the behavioral sciences", *Lawrence Erlbaum Associates.*
[29] Kopetz, H. (2013), *Real-Time Systems*, Springer.
[30] Object Management Group (2000), Product Data Management Enablers, https://www.omg.org/spec/PDME/1.3/About-PDME/,accessed 2022-04-01.
[31] Object Management Group (2020), "Unified Architecture Framework".
[32] Becker, J., M. Rosemann, and R. Schütte (1999), "Referenzmodellierung: State of the art und Entwicklungsperspektiven", *Heidelberg: Physica-Verlag.*
[33] Maissel, Joe (2017), „Wanted – A Reference Architecture for Enterprise Architecture Repositories", Association of Enterprise Architects.

AUTHORS

Peter Hillmann is a postdoctoral researcher at the Universität der Bundeswehr München, Germany. He received a M.Sc. in Information-System-Technology from Dresden University of Technology (2011) and a Dr. rer. nat. (Ph.D. in science) degree in Computer Science (2018) from the Universität der Bundeswehr München. His areas of research are cyber security, distributed systems as well as enterprise architecture and optimization.

Diana Schnell is a research associate in the Department of Applied Computer Science at Universität der Bundeswehr München, Germany. She holds a M.Sc. in Business Administration and Engineering from Universität Duisburg-Essen. Her research interests include enterprise architecture management and an associated improvement in business alignment.

Harald Hagel is a senior researcher at the Universität der Bundeswehr München, Germany. He obtained a Dr.-Ing. degree in mechanical engineering (1988) from the Universität der Bundeswehr München. His research area is the application of hybrid and knowledge-based modeling techniques based on business process engineering in industrial practice.

Andreas Karcher holds the Chair for Software Tools and Methods for Integrated Applications in the Department of Applied Computer Science at Universität der Bundeswehr München, Germany since 2003. The Chair of Software Tools and Methods for Integrated Applications focuses its research and teaching on digital transformation on the basis of a model- and architecture-based design of integrated application system landscapes.

A FRAMEWORK TO PROTECT IOT DEVICES FROM ENSLAVEMENT IN A HOME ENVIRONMENT

Khalid Al-Begain[1], Murad Khan[1], Basil Alothman[1],
Chibli Joumaa[1] and Ibrahim Rashed[2]

[1]Kuwait College of Science and Technology, Kuwait
[2]Computer Engineering Department Kuwait University, Kuwait

ABSTRACT

The Internet of Things (IoT) mainly consists of devices with limited processing capabilities and memory. Therefore, these devices could be easily infected with malicious code and can be used as botnets. In this regard, we propose a framework to detect and prevent botnet activities in an IoT network. We first describe the working mechanism of how an attacker infects an IoT device and then spreads the infection to the entire network. Secondly, we propose a set of mechanisms consisting of detection, identifying the abnormal traffic generated from IoT devices using filtering and screening mechanisms, and publishing the abnormal traffic patterns to the rest of the home routers on the network. Further, the proposed approach is lightweight and requires fewer computing capabilities for installation on home routers. In the future, we will test the proposed system on real hardware, and the results will be presented to identify the abnormal traffic generated by malicious IoT devices.

KEYWORDS

Botnet, IoT, Malicious Activities, Abnormal Traffic Detection.

1. INTRODUCTION

The applications of IoT spread into different places such as homes, offices, buildings, and other environments. In addition, these devices provide several services such as controlling home appliances in a smart home, remotely monitoring an agricultural farm for humidity, temperature, etc., and industry for automating robots, manufacturing systems and controlling various hardware. As these devices have different services, ubiquity and inconspicuousness become two important characteristics of these devices [1]. Therefore, the attackers always target these devices to gain different advantages, such as launching a DDoS attack on different servers and then asking for ransom. In recent years, IoT devices proliferated in different fields of engineering, health, smart home, and smart cities, and it is estimated that 50 billion devices will be connected within the next few years. Therefore, sophisticated security mechanisms are needed to provide enough security to these devices.

Recently, several solutions to protect IoT devices from enslavement have been presented. For instance, some of these security mechanisms are based on employing deep learning models to train a neural network and then using it to detect the malicious activity pattern in the network traffic originating from the IoT devices. However, such solutions are favourable in those situations when the IoT devices are provided with high processing power and enough memory

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 171-178, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122012

[2]. One of the solutions is to make a cluster of the Raspberry Pi and then control the entire home network with these devices [3, 4]. However, in such situations, it would not be easy to maintain the management of the clusters. Lightweight security is a favourable solution for protecting IoT devices from enslavement. One of the reasons is that IoT devices have limited memory and processing power, and therefore, they can easily support lightweight security protocols [5, 6]. However, such solutions are mainly designed for IoT devices which again require extra management on the network layer. Therefore, modifying the network layer protocols requires additional work and management.

In order to handle the issues in neural networks, clusters, and lightweight-based security solutions for IoT, this article presents a security framework for a home environment. The working mechanism of the proposed scheme consists of two main parts; first, an attacker launches an attack to enslave an IoT device by installing malicious code. Similarly, the enslaved IoT device, a bot, infects the other IoT devices within the same network cluster with the same malicious code. Further, the attacker configures the bots to launch a DDoS attack on the victim server. Secondly, we devised a security mechanism to detect abnormal traffic from the bots on the edge router. The edge router is programmed to block abnormal traffic from the bots.

Further, any open port will be blocked to prevent future connections to the IoT devices. Finally, the edge router will share the information about the abnormal traffic with the rest of the routers attached to it.

The rest of the paper is divided into the following parts. Section 2 presents a thorough literature study of the current botnet and related security mechanisms. Section 3 presents the conceptual idea of the proposed scheme. Section 4 presents an overview and brief explanation of the experimentation study. Finally, the conclusions are given in Section 5.

2. RELATED WORKS

In recent literature, researchers suggested a number of techniques to protect IoT devices from malicious attacks. These techniques are based on a number of parameters, such as the processing power, memory, communication range, etc., of the IoT devices. However, it is still challenging to design a generic security system that can incorporate all the above-mentioned parameters. For instance, the authors in [7] focused on tackling the networks of devices that have been infected with malware by modelling the behaviour of malware spread, the classification of malicious traffic, and the analysis of traffic anomalies. This paper introduces a system for ANTicipating botnEts (ANTE) signals based on machine learning techniques. By learning to recognize different forms of botnets throughout their execution, ANTE's architecture allows it to adapt to a variety of circumstances. In order to maximize the accuracy of categorization, ANTE automatically picks the best optimal machine-learning pipeline for each type of botnet. A similar study is presented in [8] to detect botnets using the autoencoder machine learning technique. A large-scale IoT network traffic data is encoded using a long short-term memory auto-encoder in order to minimize the dimensionality of features (LAE). With the use of a deep bidirectional long-short-term memory (BLSTM), the authors propose that it is possible to categorize network traffic samples properly (BLSTM). To validate the efficacy of the proposed hybrid DL technique, extensive experiments were performed on the BoT-IoT data set. A weather station, smart fridge, motion-activated lighting, a remotely operated garage door, and a smart thermostat were among the IoT devices included in the testbed setup. Bot-IoT also includes millions of samples of IoT botnet attack traffic.

As machine learning techniques require a sufficient amount of processing power and memory, therefore, it is only appropriate for IoT devices that have such features and capabilities. For

lower-capability IoT devices that may exist in a variety of home appliances, It is necessary to use other techniques which are faster and require a limited amount of processing power and memory. For instance, a lightweight security scheme is designed to limit network access using segment units set with various device features, network information, and service types [9]. The architecture's capacity to prevent the propagation of threats on the IoT network was validated by the suppression of botnet creation in the botnet creation experiment testbed using the Mirai virus. In the same context, another scheme is presented using a fuzzy logic mechanism to prevent IoT devices from enslavement [10]. This paper proposes a unique approach capable of detecting IoT-Botnet attacks while avoiding the difficulties related to the limitations of knowledge-based representation and binary decisions. The contribution of this research paper is to present a detection method for the IoT-BotNet attack using Fuzzy Rule Interpolation (FRI). These advantages assist the Intrusion Detection System (IDS) in producing more realistic and comprehensive alerts. The suggested technique was used for an open-source BoT-IoT dataset from the Cyber Range Lab at UNSW Canberra Cyber. The proposed technique was assessed and found to have a detection rate of 95.4%, according to the study. Due to its fuzzy nature, it was able to successfully smooth the boundary between regular and IoT-BotNet traffic, and it was able to create the appropriate IDS warning in the case that the knowledge-based representation failed.
Intrusion Detection Systems (IDS) are widely used to handle abnormal traffic and block illegal access to an IoT device [11, 12]. For instance, in [13], the authors proposed a three-layer IDS that detects a variety of prominent network-based cyber-attacks on IoT networks using a supervised method. The system is capable of classifying the type, profiling the usual behaviour of each IoT device, and determining the sort of attack that has been launched. The system is tested in a smart home testbed that includes eight prominent commercially accessible devices. The effectiveness of the proposed IDS architecture is assessed by deploying 12 attacks from four major network-based attack categories. The system is also tested against four multistage attack scenarios with complicated event chains. As this system is based on supervised learning, there is a fair chance that the system will respond in a significantly large amount of time if a greater number of IoT devices are connected to the home network. One way to avoid the use of supervised learning is to perform IoT botnet isolation and detection on access-level routers that enable automated detection of unprotected IoT devices, isolation based on the access router's internal firewall, an update mechanism based on a CVE online service, and self-optimizing scanning [14]. The researchers used two testbeds to perform a quantitative evaluation of the proposed approach using both virtual and real hardware devices.

The paper [15] presents an IoT botnet testbed that emulates IoT devices in a DETER-lab-based infrastructure (NCL). It has all the ancillary services like a DNS, CNC, ScanListen/loader, and a victim server that a botnet needs for full operation. The paper showcased a virtualization method using (QEMU) by emulating a Raspbian OS with limited and fixed resources for each emulated device. There were 10 IoT-emulated devices connected to gateways and routers in a contained environment. The botnet used for experimentation was "Mirai", whose source code was altered to remove the errors faced in the emulated environment. All the technical details about setting up the testbed were shown, and validation of the testbed was achieved by using a UDP flood and TCP SYN flood attack on the testbed's victim server and monitoring the packet traffic to see disruption of the victim server upon being attacked.

The proposed approach presented in this paper is similar in principle to the system presented in the above-mentioned manuscript. However, the major difference between our proposed scheme and [15] is our scheme is a lightweight one. In addition, we also consider the home edge router's limited memory and processing power when scanning and detecting the network traffic. Also, the knowledge gained by a home edge router is published with the rest of the routers to reduce the time for processing and to detect abnormal traffic patterns.

3. PROPOSED SCHEME

The working of the proposed scheme is further divided into two phases: 1) DDoS Attack Phase and 2) DDoS Attack Detection and Prevention Phase.

3.1. DDoS Attack Phase

In the attack phase, an attacker infects a smart home device, such as a washing machine, refrigerator, etc., by scanning for open ports using or performing a brute force attack on random IP addresses. As soon as the attacker finds an open port, it establishes a connection to the smart home device to inject the malicious code. The attacker later uses this malicious code to launch a DoS attack from the infected device. Similarly, the infected devices distribute the malicious code with the devices attached to the same network. The spreading of infection to other devices is continued until all the devices on the home network are infected. The attacker also uses the switch/router attached to other switches and routers for spreading the infection to other home networks via an infected device. The infected devices have the functionality of automatically scanning the network for other devices, and as soon as they find any other device, they infect them by sending them the malicious code. Similarly, these infected devices, which are also called bots, can perform a number of activities, such as scanning other devices on the network, sending malicious codes to the infected devices, sending spam emails, and can launch an attack. The entire working mechanism of the DDoS attack phase is presented in Figure 1.

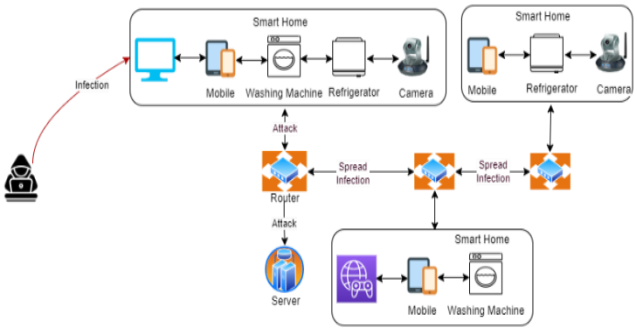


Figure 1. The attack scenario

Fig 1. shows the mechanism of how an attacker infects a device and then spreads the infection to the network.

3.2. DDoS Detection and Prevention Phase

In the prevention phase, the home router uses a set of mechanisms to block the DDoS attack launched by the bots. As we already discussed, it is difficult and time-consuming to add neural network models to identify any malicious traffic generated from home devices. Also, neural networks always require huge amounts of data to be trained to differentiate between legitimate and abnormal traffic. We also know that home routers are always available with limited processing and programming capabilities. Therefore, it would be difficult to program and configure them with neural network capabilities. In order to deal with the abnormal traffic generated by the bots, we divided the working mechanism into the following steps.

3.2.1. Detection

In order to detect the abnormal traffic generated by the bots, each packet distant from a server is checked for the destination IP address. Further, a counter on the destination IP addresses will be set. If, for a particular destination IP address, the counter exceeds a pre-defined threshold, then the traffic will be considered abnormal. In a similar context, each device that generated similar traffic will be blocked.

3.2.2. Filtration

One of the possibilities in the case of step 1 is that there are difficulties in differentiating between normal and abnormal traffic. In this regard, we developed a filtration mechanism based on defining various thresholds similar to step 1. However, the difference between the threshold used in step 2 is that there legitimate and abnormal traffic can be classified into two separate categories. Finally, the outcome of the filtration step will help in traffic classification and separation between normal and abnormal traffic.

3.2.3. Screening

In this step, all the logs and entries saved in the home devices and the home router are checked regularly to avoid any possible attack in the future. It also helps in identifying malicious actions by comparing the logs of normal and abnormal traffic.

3.2.4. Publishing

In order to reduce the processing time on the rest of the routers attached to the current router, the current router shares the malicious actions information with the routers attached to it via a wired connection. However, we also programmed the rest of the routers to repeat steps 1 to 3. Further, adding the information from the current router may also help in identifying the malicious code as quickly as possible. One of the main reasons for publishing the information to other routers is to deal with malicious traffic and prevent the DDoS attack in the early stages.

Finally, the mechanism of detection and prevention of the DDoS attack is given in Figure 2. The proposed idea of the prevention of the possible DDoS attack is simple and requires less amount of time to process the packets from infected devices. Also, the malicious network traffic generated from the bots is handled and blocked. These bots are also identified, and the open ports used by the attacker are closed for any suspicious connections.

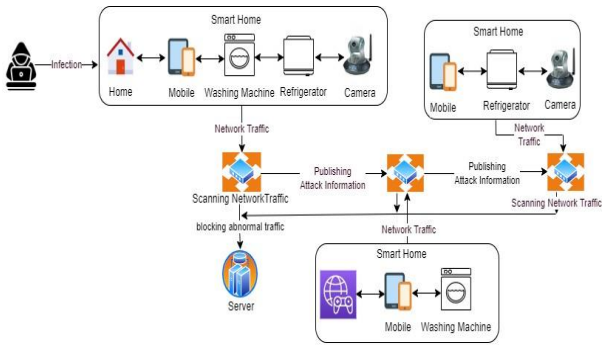


Fig 2. Prevention of DDoS attacks from infected home devices

4. RESULTS AND DISCUSSION

In this article, we present a conceptual idea of our proposed DDoS prevention system for home environments. To validate the proposed system, a fully isolated testbed is set up at the Network Security Laboratory at KCST to inject the malicious code into the home network by brute forcing all the available ports to establish a connection. For this purpose, we will set up different Raspberry PI with variable configurations connected to a home router to emulate the different IoT devices embedded in the different home appliances. The traffic generated from each home device will be scanned to detect abnormal traffic for a specific period and a specific destination address. If the home devices are sending data to a common destination IP address, then the traffic will be classified as abnormal traffic. The abnormal traffic will be discarded, and all the open ports will be closed. A snapshot of the system is presented in Figure 3.

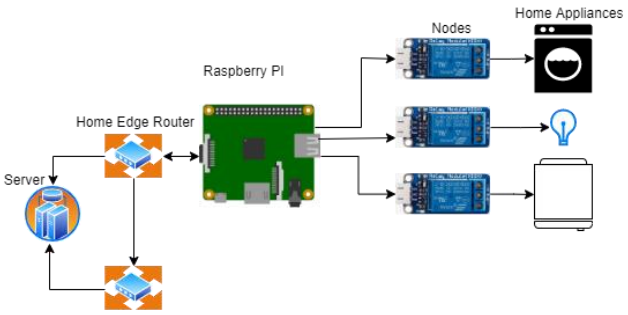


Fig 3. A testbed scenario of the proposed system.

5. CONCLUSION

This paper provides a framework for injecting malicious code into an IoT network to enslave home devices. Further, a detection mechanism is devised to scan the traffic originating from the home devices for possible malicious activities. As soon as a malicious activity, i.e., a DDoS attack, is detected by the home router, the traffic coming from the home devices is blocked, and the results are published to the routers connected to the current router. This is a light weighted solution that could be useful for home devices and routers with limited processing capabilities. In the future, we are planning to set up a similar structure using real hardware and testbed in the form of Raspberry Pi and home routers.

ACKNOWLEDGEMENT

The paper is a result of the project, which is fully funded by the Kuwait Foundation for Advancement of Science (KFAS) under Project No. (PR17-18QI-03).

REFERENCES

[1] B. N. Silva, M. Khan and K. Han, "Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities," *Sustainable Cities and Society*, vol. 38, pp. 697-713, 2018.

[2] B. Jan, H. Farman, M. Khan, M. Imran, I. U. Islam, A. Ahmad, S. Ali and G. Jeon, "Deep learning in big data analytics: a comparative study," *Computers & Electrical Engineering*, vol. 75, pp. 275-287, 2019.

[3] K. Doucet and J. Zhang, "Learning cluster computing by creating a Raspberry Pi cluster," in *Proceedings of the SouthEast Conference*, 2017.

[4] J. Coelho and L. Nogueira, "Enabling Processing Power Scalability with Internet of Things (IoT) Clusters," *Electronics,* vol. 11, no. 1, p. 81, 2021.
[5] R. Fotohi and H. Pakdel, "A lightweight and scalable physical layer attack detection mechanism for the internet of things (IoT) using hybrid security schema," *Wireless Personal Communications,* vol. 119, no. 4, pp. 3089-3106, 2021.
[6] V. Rao and K. V. Prema, "A review on lightweight cryptography for Internet-of-Things based applications," *Journal of Ambient Intelligence and Humanized Computing,* vol. 12, no. 9, pp. 88358857, 2021.
[7] A. B. de Neira, A. M. Araujo and M. Nogueira, " Early botnet detection for the internet and the internet of things by autonomous machine learning," in *16th International Conference on Mobility, Sensing and Networking (MSN)*, 2020.
[8] S. I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui and H. Gacanin, "Hybrid deep learning for botnet attack detection in the internet-of-things networks," *IEEE Internet of Things,* vol. 8, no. 6, pp. 4944 - 4956, 2020.
[9] J. Lim, S. Sohn and J. Kim, "Proposal of Smart Segmentation Framework for preventing threats from spreading in IoT," in *International Conference on Information and Communication Technology Convergence*, 2020.
[10] M. Al-Kasassbeh, M. Almseidin, K. Alrfou and S. Kovacs, "Detection of IoT-botnet attacks using fuzzy rule interpolation," *Journal of Intelligent & Fuzzy Systems,* vol. 39, no. 1, pp. 421 - 431, 2020.
[11] A. R. Khan, M. Kashif, R. H. Jhaveri, R. Raut, T. Saba and S. A. Bahaj, "Deep learning for intrusion detection and security of Internet of things (IoT): current analysis, challenges, and possible solutions.," *Security and Communication Networks,* vol. PP, p. 1, 2022.
[12] Y. Otoum, D. Liu and A. Nayak, "DL-IDS: a deep learning–based intrusion detection framework for securing IoT," *Transactions on Emerging Telecommunications Technologies,* vol. 33, no. 3, p. e3803, 2022.
[13] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos and P. Burnap, "A supervised intrusion detection system for smart home IoT devices," *IEEE Internet of Things,* vol. 6, no. 5, pp. 9042 - 9053, 2019.
[14] C. Dietz, R. L. Castro, J. Steinberger, C. Wilczak, M. Antzek, A. Sperotto and A. Pras, "IoT-botnet detection and isolation by access routers," in *9th International Conference on the Network of the Future (NOF)*, 2018.
[15] A. Kumar and T. J. Lim, "A secure contained testbed for analyzing IoT botnets," *International Conference on Testbeds and Research Infrastructures,* pp. 124 - 137, 2018.

AUTHORS

Khalid Al-Begain is the founding President of Kuwait College of Science and Technology. He served as the President of the European Council for Modelling and Simulation (ECMS) (20062018) and as President of the Federation of European Simulation Societies (EuroSim) (2010-2013). He was the co-founder and chairman of the first National Welsh Industrial Cyber Security Summit in Newport, UK, in 2014, organized jointly with Airbus, General Dynamics and the UK NCSC. He worked in many organizations and countries, including Hungary, Jordan, Germany and the UK, where his last position was as a Professor of mobile networking at the University of South Wales and Director of the Centre of Excellence in Mobile applications and Services. He won numerous awards, including the John von Newman Computer Award (1986) and the Inspire Wales Award for Science and Technology (2013). In 2006, he received Royal Recognition from Her Majesty Queen Elizabeth II for his contributions to the British scientific community. He supervised 28 successful PhD projects and examined 28 PhDs. Over the years, he led many major national projects securing over GBP20 million in funding for research and consultancy. He registered two granted patents, authored two books and edited 26 books, in addition to over 200 papers in refereed journals and conferences. He was the general chair of 31 international conferences.

Murad Khan received a BS degree in computer science from the University of Peshawar, Pakistan, in 2008. He has completed his Master's and PhD degrees both in computer science and engineering from the School of Computer Science and Engineering in Kyungpook National University, Daegu, Korea. Dr. Khan is currently

working as an assistant professor at the Kuwait College of Science and Technology, Kuwait. Dr. Khan also served as Brain Pool Fellow at Kyungpook National University, Daegu, Korea, from December 2019 to December 2021. Dr. Khan published over 100 International conference and Journal papers along with two book chapters and is an editor of books in Springer and CRC Press. Dr. Khan served as an editorial member of various special sections in world-renowned journals such as Computer & Electrical Engineering, Transactions on Emerging Telecommunications Technologies, etc. He also served as a TPC member in world-reputed conferences and as a reviewer in numerous journals such as IEEE Communication Magazine, Future Generation Computer Systems, IEEE Access, etc. His area of expertise includes ad-hoc and wireless networks, architecture designing for the Internet of Things, Communication Protocols designing for smart cities and homes, Big Data Analytics, etc. Email: m.khan@kcst.edu.kw URL: https://www.kcst.edu.kw/default/viewpeople?id=3

Basil Alothman joined Kuwait College of Science and Technology (KCST) as Assistant Professor at Computer Science and Engineering Department. Dr. Basil graduated from De Montfort University, Leicester, UK, with a PhD in Computer Science. He received his MSc in Computer Science from the University of Hertfordshire, UK and his BSc in Computing and Information Systems from the University of Dubai, UAE. Dr. Basil is mainly interested in cybersecurity science or, more specifically, computer and network security issues, mobile security, computer privacy, OSINT, reverse engineering, cloud and VM security, big data security, IoT security, and Botnet detection techniques.

Chibli Joumaa has been an Associate Professor of Computer Engineering in the Faculty of Engineering and Computer Science of the Kuwait College of Science and Technology (KCST) since April 2020. In addition to his role in the Computer Science and Engineering department, he oversees accreditation-related processes in collaboration with the coordinators, dean, and President. He has previously occupied several teaching, administrative, and managerial positions in reputed educational institutions. Dr. Chibli Joumaa holds a bachelor's and master's degree in electrical engineering from the University of Balamand in Lebanon in 2004 and a master's in Network Telecommunication and System Architecture from France in 2005. He received his PhD in Computer Engineering from the University of Technology of Belfort-Montbeliard, France, in 2010. He is the author of many publications and has attended many professional training and workshops for accreditation and academic advancement.

Dr. Ibrahim Rashed Ebrahim Alrashed received his PhD and MS degree in computer engineering from The University of Southern California, Los Angeles, CA in 1997 and 1993. Since 1997 he has been a faculty member in the department of computer engineering at Kuwait University where he is currently an associate professor. His research interests includes network security, mobile and wireless networks, cloud and edge computing and blockchain.

SENTIMENT CLASSIFICATION OF CODE-SWITCHED TEXT USING PRE-TRAINED MULTILINGUAL EMBEDDINGS AND SEGMENTATION

Saurav K. Aryal, Howard Prioleau and Gloria Washington

Department of Electrical Engineering and Computer Science,
Howard University, Washington DC, USA

ABSTRACT

With increasing globalization and immigration, various studies have estimated that about half of the world population is bilingual. Consequently, individuals concurrently use two or more languages or dialects in casual conversational settings. However, most research is natural language processing is focused on monolingual text. To further the work in code-switched sentiment analysis, we propose a multi-step natural language processing algorithm utilizing points of code-switching in mixed text and conduct sentiment analysis around those identified points. The proposed sentiment analysis algorithm uses semantic similarity derived from large pre-trained multilingual models with a handcrafted set of positive and negative words to determine the polarity of code-switched text. The proposed approach outperforms a comparable baseline model by 11.2% for accuracy and 11.64% for F1-score on a Spanish-English dataset. Theoretically, the proposed algorithm can be expanded for sentiment analysis of multiple languages with limited human expertise.

KEYWORDS

Code-switching, Sentiment Analysis, Multilingual Embeddings, Code-switch points, Semantic Similarity.

1. INTRODUCTION

Linguistic code-switching is the concurrent use of two or more languages or dialects in a conversation. According to The International Organization for Migration's (IOM) World Immigration Report, physical migration only accounts for 3.1% of the world population in 2020. However, growth in internet technology has introduced a new wave of unfiltered and cross-cultural-lingual interactions, which provide a fertile ground for linguistic code-switching. Therefore, sentiment analysis of linguistic code-switching is a critical task in the future of multilingual natural language processing and understanding.

Code-switching, in general, has been studied extensively in psycholinguistics and sociolinguistics [1,2,3]. While the psycholinguistic definition of code-switching is multi-faceted, this paper, for readability and convenience, will refer to only the linguistic variant as code-switching henceforth. While research in sentiment analysis has proliferated over the past decade, most research focused on utilizing monolingual textual data. Consequently, sentiment analysis of code-switched text is limited. Furthermore, sentiment analysis of code-switched data is a more complex task because two languages' unification creates a tertiary low-resource language. This low-resource language

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 179-186, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122013

maintains the two primary languages' features while creating a new topology that classification models must learn separately.

To further the work in code-switched sentiment analysis, we propose a multi-step natural language processing algorithm utilizing points of code-switching in mixed text and conduct sentiment analysis around those identified points. The proposed sentiment analysis algorithm uses semantic similarity derived from large pre-trained multilingual models with a handcrafted set of positive and negative words to determine the polarity of code-switched text. Additionally, the proposed algorithm can be adapted for sentiment analysis of multiple languages with limited human expertise.

2. RELATED WORKS

In this section, we will review literature related to transformer-based architectures for sentence embeddings and sentiment analysis of code-switched text.

Extracting embeddings is the process of mapping textual data to vectors of real numbers. These embeddings are learnt representations of sentences and can be used in semantic similarity. Bidirectional Encoder Representations from Transformers (BERT) [4] and related variants [5,6] have provided state-of-the-art embeddings for numerous natural language processing (NLP) tasks. While the seminal BERT does not compute sentence-level embeddings, researchers have trained BERT model for sentences [7,8,9]. However, BERT models are computationally expensive for semantic similarity searches, Siamese-BERT (SBERT) [10] enabled efficient comparisons. Various multi-lingual SBERTs have been applied code-switched text [11]. The findings of [11] suggest that a pre-trained multi-lingual model does not necessitate high-quality representations of code-switched text.

In another line of work, bilingual embeddings have been introduced to represent code-switching sentences [12,13,14,15,16]. However, these approaches are limited to fewer languages. Independently, much work has been done on extracting sentence embeddings [17,18,19,20]. Of these, Universal Sentence Encoder [20] was trained with the explicit goal of learning multilingual embeddings for downstream tasks.

Despite available embedding methods and accessible code-switched social media text, few labelled data sets have been released for sentiment analysis. These datasets include Malayalam-English and Tamil-English [21,22,23], Spanish-English [24], and Hindi-English [25].

Previous work has shown BERT-based models achieve state-of-the-art performance for code-switched languages in tasks like offensive language identification [26] and sentiment analysis [27]. Custom architectures and algorithms have also been proposed [24,28,29].

Of these approaches we chose [24] as our baseline of comparison since this model, to the best of our knowledge, is the current state-of-art for this specific dataset-task pair and has shown improved performance when compared to over 10 other approaches. Furthermore, other models have only explicitly focused on Dravidian languages which is likely easier than Spanish-English sentiment analysis due to difference in lexical relatedness [30,31].

In our work, we explore the possibility of exploiting multi-lingual embedding spaces toward sentiment analysis of code-switched text. Our proposed algorithm can be extended in a language-agnostic manner with nominal human involvement.

3. METHODOLOGY

3.1. Datasets

To analyze this project, we utilize two primary datasets. Firstly, the Spanish-English Code-switching Dataset [24] sourced from tweets that contained both Spanish and English. Secondly, the Word Level Sentiment Lexicon [32] dataset which provides a lexicon of negative and positive words for many languages. For this work, we only utilize the Spanish and English subsets.

3.2. Data Preprocessing

We convert the text data to lower case and remove, to the best of our abilities, all punctuation, white spaces, and sensitive information such as email addresses, URLs, numbers, and social media entities like emojis and hashtags.

3.3. Multilingual Models

In this work, we compare two standard, pre-trained, multilingual model for extracting sentence-level embeddings: SBERT-based model XLM-Roberta (XLM-RBASE) [10] and Universal Sentence Encoder (USE) developed by Google in [20]. Our algorithm's intuition relies on the hypothesis that sentence embeddings of segments that are negative sentence will be closer in vector space to a set of negative ground truth words/phrases and vice versa for positive sentences and their corresponding ground words despite their language of origin.

3.4. Code-switching Identification Algorithm

The proposed system in this work is composed of two primary parts. The first part is the detection of code-switch points in text. To implement this, we utilized a Naïve Bayes classifier [33] on our Spanish-English dataset to detect likely points where code-switching might occur. In their dissertation research, [33] seeks to detect code-switching points in several interviews where the interviewer and the interviewee switch between English and Kiswahili. For reference, an example of code-switching in Swahili-English points can be found in Table 1.

Table 1. Code-switching points represent point at which the language of the sentence switches from one language to another. This table show examples of code-switching points in various code-mixed language combinations.

Original Text	Translation	Code-Switching Points
Okay, Okay, na unafikiria ni important kujua native language?	Okay, and do you think it is important to know native language?	“okay,” “ni,” “important,” and “kujua”

Table 2. Pseudo Code for English-Spanish Text

pos_eng: The list of ground positive English words
neg_eng: The list of ground negative English words
pos_es: The list of ground positive Spanish words
neg_es: The list of ground negative Spanish words
 # compare the language-segments with negative and positive word sets
scores = cosineSimilarity(spanish_segments, pos_es)
neg_scores = cosineSimilarity(spanish_segments, neg_es)
eng_scores = cosineSimilarity(english_segments, pos_eng)

```
eng_neg_scores = cosineSimilarity(english_segments, neg_eng)
# get the total Positive and Negative score
neg = sum(neg_scores, eng_neg_scores)
pos = sum(scores, eng_scores)
# Thresholding
if pos >neg:
   return “Positive”
elif neg <pos:
    return “Negative”
```

3.5. Sentiment Analysis Algorithm

The code-switch points are passed into the second part of the overall system for sentiment analysis. Using these results, the code-switched text is segmented into constituent languages. Assuming the two languages involved are know, which for our dataset is English and Spanish, the segments embeddings are extracted independently using USE and XLM-RBASE. A set of positive and negative sentiment lexicons [32] for each language and their embeddings are also extracted. Segments were compared to their language-specific sets of negative and positive embeddings. The returned comparison from the four sets of comparisons from two languages each with two sets of curated embeddings is the totalled and the total negative score and total positive scores are used to determine the polarity of the overall codeswitched sentence. The pseudocode algorithm for detecting the sentiment post-segmentation into English and Spanish can be seen in Table 2. For clarity, we summarize the full proposed algorithm in Figure 1 below.

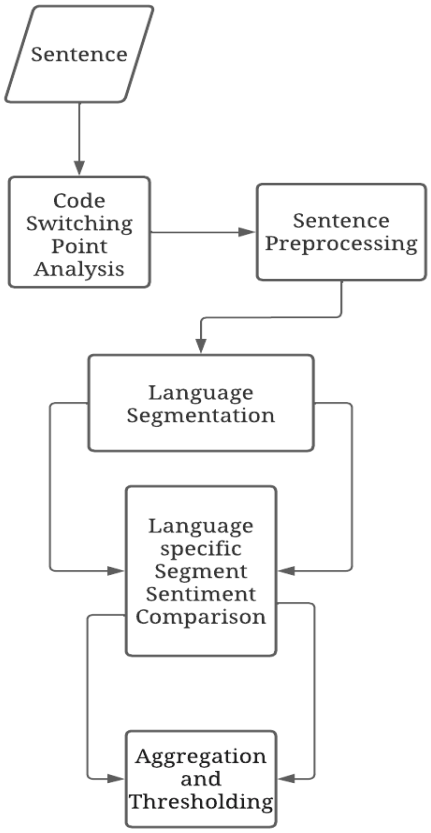


Figure 1. Full Proposed Algorithm

To enable future comparisons and ensure reproducibility, a link to access our source code has been provided in the *Appendix*.

4. EXPERIMENTAL RESULTS

Table 3. Comparison of results

Model	Accuracy	Macro F1	Precision	Recall
GIRNET 1L	63.3	62.4	63.4	61.8
GIRNET 2L	62.0	61.3	61.8	61.1
USE	71.9	71.7	**74.5**	73.4
XLM-RBASE	**74.5**	**74.0**	74.2	**73.8**

4.1. Sentiment Analysis Evaluation

Table 3 shows the sentiment analysis results via the semantic similarity task. This table compares our results with other methods for classifying the sentiment of the code-switched text. The first two rows show the results provided by the baseline [24]. The baseline results from the paper, which utilize a unified position-sensitive multi-task recurrent neural network (RNN), produce an accuracy of 63.3% and a macro-F1 score of 62.36% on the Spanish-English code-switched dataset. For context, in table 3, GIRNET 1L represents the architecture with one layer, whereas GIRNET 2L illustrates the architecture with two layers. The third and fourth columns show results from our algorithm when utilizing Universal Sentence Encoders (USE) and XLM-RBASE. For further analysis, we have provided the confusion matrix of the approach utilizing XLM-RBASE and USE in Tables 4 and 5 respectively. In terms of overall performance, we notice that XLM-RBASE slightly outperforms USE embeddings. However, USE embeddings are better for predicting positive sentiments than XLM-RBASE embeddings but much worse for negative sentiments. Furthermore, false positive and false negative predictions of the best-performing model have been provided the *Appendix* for further qualitative analysis.

Table 4. XLM-RBASE Model Confusion matrix

	True Positive	True Negative
Predicted Positive	429	201
Predicted Negative	160	**624**

Table 5. USE Model Confusion matrix

	True Positive	True Negative
Predicted Positive	**553**	77
Predicted Negative	321	463

4.2. Future Work and Limitations

While the proposed approach significantly improves the baseline's performance metrics for the task, the model's performance still leaves more to be desired. Furthermore, while our proposed approach works on English-Spanish, the work done explicitly on Dravidian languages or utilizing domain-specific dependence are not directly comparable. Additionally, even though these languages are prominently studied in code-switching literature, all current datasets only contain code-switching between English and one other language whereas other relevant language pairs such as French-Tamil, Nepali-English, Hopi-Tewa, Swahili-English, Hindi-Nepali, and Latin-

Irish currently lack datasets. Thus, much work needs to be done in creating datasets that go beyond the language pairs currently being studied.

The algorithm proposed still depends on models that can provide accurate representations across the languages considered and assume the languages involved have already been identified. This requirement can be a particular issue for other low-resource languages. Furthermore, while the given approach requires limited human intervention, we may still need human expertise to create positive and negative word sets for the language of choice. We hope that with increasing need and interest for this topic, larger and more challenging datasets and increased research funding will follow in the future.

Finally, the authors would like to acknowledge that the approach is still rather rudimentary and should definitely be improved. However, given that this seemingly simple approach outperformed the extensively tested and task-specific custom model of the baseline, perhaps future research in this topic needs to take a new outlook beyond novel and custom neural network architectures while recognizing the niche task at hand.

5. CONCLUSION

In this work, we present our research towards developing a pipeline to detect sentences where code-switching present and to classify the polarity of these sentences. This pipeline has the potential to assist researchers in further improving the sentiment analysis of code-switched text.
The results show that our approach significantly surpasses the baseline performance (GIRNET) by 11.2% for accuracy and 11.6% for the F1 -score. Furthermore, when comparing the two embedding models utilized within our algorithm, XLM-RBASE outperformed the Universal Sentence Encoder. These results suggest that our algorithm utilizing semantic similarity and multilingual language models for representation can offer a method for detecting the sentiment of code-switched text in a language agnostic manner. Theoretically, the proposed approach may be applied to a number of downstream classification tasks and applications that involve code-switched textual data. However, larger and comprehensive data sets may enable more complex and better performing approaches for language-agnostic sentiment analysis of code-switched text. To be added if accepted.

ACKNOWLEDGEMENTS

This project was supported (in part) by from the Northrop Grumman Corporation Under Award #201603 and The National Security Agency, USA under Award Number H98230-19-P-1674. The authors would also like to acknowledge the initial work and support provided by Mr. Cesa Salaam in the preparation of the manuscript. The content is solely the responsibility of the authors and does not necessarily represent the official views of the Northrop Grumman Corporation, or The National Security Agency.

REFERENCES

[1] S. Poplack, ‘Code-switching (linguistic)’, International encyclopedia of the social and behavioral sciences, τ. 12, σσ. 2062–2065, 2001.
[2] &. M. C. Poplack S. Sankoff D., ‘The social and linguistic processes of borrowing and assimilation’, Linguistics, τ. 26, 1988.
[3] G. Sankoff, ‘Language use in multilingual societies: Some alternative approaches’, J. Pride & J. Homes (ed.). Sociolinguistics, σσ. 33–52, 1971.
[4] J. Devlin, M.-W. Chang, K. Lee, και K. Toutanova, ‘Bert: Pre-training of deep bidirectional transformers for language understanding’, arXiv preprint arXiv:1810. 04805, 2018.

[5] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, και L. Specia, 'Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation', arXiv preprint arXiv:1708. 00055, 2017.
[6] Y. Liu κ.ά., 'Roberta: A robustly optimized bert pretraining approach', arXiv preprint arXiv:1907. 11692, 2019.
[7] C. May, A. Wang, S. Bordia, S. R. Bowman, και R. Rudinger, 'On measuring social biases in sentence encoders', arXiv preprint arXiv:1903. 10561, 2019.
[8] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, και Y. Artzi, 'Bertscore: Evaluating text generation with bert', arXiv preprint arXiv:1904. 09675, 2019.
[9] Y. Qiao, C. Xiong, Z. Liu, και Z. Liu, 'Understanding the Behaviors of BERT in Ranking', arXiv preprint arXiv:1904. 07531, 2019.
[10] N. Reimers και I. Gurevych, 'Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks', CoRR, τ. abs/1908.10084, 2019.
[11] G. I. Winata, S. Cahyawijaya, Z. Liu, Z. Lin, A. Madotto, και P. Fung, 'Are Multilingual Models Effective in Code-Switching?', arXiv preprint arXiv:2103. 13309, 2021.
[12] M. Faruqui και C. Dyer, 'Improving vector space word representations using multilingual correlation', στο Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, 2014, σσ. 462–471.
[13] K. M. Hermann και P. Blunsom, 'Multilingual models for compositional distributed semantics', arXiv preprint arXiv:1404. 4641, 2014.
[14] M.-T. Luong, H. Pham, και C. D. Manning, 'Bilingual word representations with monolingual quality in mind', στο Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, 2015, σσ. 151–159.
[15] A. Joulin, P. Bojanowski, T. Mikolov, H. Jégou, και E. Grave, 'Loss in translation: Learning bilingual word mapping with a retrieval criterion', arXiv preprint arXiv:1804. 07745, 2018.
[16] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, και H. Jégou, 'Word translation without parallel data', arXiv preprint arXiv:1710. 04087, 2017.
[17] R. Kiros κ.ά., 'Skip-thought vectors', στο Advances in neural information processing systems, 2015, σσ. 3294–3302.
[18] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, και A. Bordes, 'Supervised learning of universal sentence representations from natural language inference data', arXiv preprint arXiv:1705. 02364, 2017.
[19] A. Williams, N. Nangia, και S. R. Bowman, 'A broad-coverage challenge corpus for sentence understanding through inference', arXiv preprint arXiv:1704. 05426, 2017.
[20] D. Cer κ.ά., 'Universal sentence encoder', arXiv preprint arXiv:1803. 11175, 2018.
[21] B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, και J. P. McCrae, 'Corpus creation for sentiment analysis in code-mixed Tamil-English text', arXiv preprint arXiv:2006. 00206, 2020.
[22] B. R. Chakravarthi κ.ά., 'Overview of the track on sentiment analysis for dravidian languages in code-mixed text', στο Forum for Information Retrieval Evaluation, 2020, σσ. 21–24.
[23] B. R. Chakravarthi, N. Jose, S. Suryawanshi, E. Sherly, και J. P. McCrae, 'A sentiment analysis dataset for code-mixed Malayalam-English', arXiv preprint arXiv:2006. 00210, 2020.
[24] D. Gupta, T. Chakraborty, και S. Chakrabarti, 'GIRNet: Interleaved Multi-Task Recurrent State Sequence Models', CoRR, τ. abs/1811.11456, 2018.
[25] P. Patwa κ.ά., 'Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets', στο Proceedings of the Fourteenth Workshop on Semantic Evaluation, 2020, σσ. 774–790.
[26] S. M. Jayanthi και A. Gupta, 'Sj_aj@ dravidianlangtech-eacl2021: Task-adaptive pre-training of multilingual bert models for offensive language identification', arXiv preprint arXiv:2102. 01051, 2021.
[27] A. Gupta, S. K. Rallabandi, και A. Black, 'Task-specific pre-training and cross lingual transfer for code-switched data', arXiv preprint arXiv:2102. 12407, 2021.
[28] R. Priyadharshini, B. R. Chakravarthi, M. Vegupatti, και J. P. McCrae, 'Named entity recognition for code-mixed Indian corpus using meta embedding', στο 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, σσ. 68–72.
[29] S. Dowlagar και R. Mamidi, 'Cmsaone@ dravidian-codemix-fire2020: A meta embedding and transformer model for code-mixed sentiment analysis on social media text', arXiv preprint arXiv:2101. 09004, 2021.
[30] Learner English: A teacher's guide to interference and other problems. 2001.

[31] D. L. August, M. Calderón, και M. S. Carlo, 'Transfer of Skills from Spanish to English : A Study of Young Learners REPORT FOR PRACTITIONERS , PARENTS , AND POLICY MAKERS', 2003.
[32] 'sentiment-lexicons-for-81-language'. [Εκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://www.kaggle.com/rtatman/sentiment-lexicons-for-81-language.
[33] R. A. Shirvani, 'Computational Analysis of Bilingual Natural Language', Howard University, 2018.

APPENDIX

All supplemental materials (source code, data set, results) can be found in the link below: https://bit.ly/3fZrMNJ

STREAMING PUNCTUATION FOR LONG-FORM DICTATION WITH TRANSFORMERS

Piyush Behre, Sharman Tan, Padma Varadharajan and Shuangyu Chang

Microsoft Corporation, USA

ABSTRACT

While speech recognition Word Error Rate (WER) has reached human parity for English, long-form dictation scenarios still suffer from segmentation and punctuation problems resulting from irregular pausing patterns or slow speakers. Transformer sequence tagging models are effective at capturing long bi-directional context, which is crucial for automatic punctuation. Automatic Speech Recognition (ASR) production systems, however, are constrained by real-time requirements, making it hard to incorporate the right context when making punctuation decisions. In this paper, we propose a streaming approach for punctuation or re-punctuation of ASR output using dynamic decoding windows and measure its impact on punctuation and segmentation accuracy across scenarios. The new system tackles over-segmentation issues, improving segmentation $F_{0.5}$-score by 13.9%. Streaming punctuation achieves an average BLEU-score improvement of 0.66 for the downstream task of Machine Translation (MT).

KEYWORDS

Automatic punctuation, automatic speech recognition, re-punctuation, speech segmentation.

1. INTRODUCTION

Our hybrid Automatic Speech Recognition (ASR) generates punctuation with two systems working together. First, the Decoder generates text segments and passes them to the Display Post Processor (DPP). The DPP system then applies punctuation to these segments.

This setup works well for single-shot use cases like voice assistant or voice search but fails for long-form dictation. A dictation session typically comprises many spoken-form text segments generated by the decoder. Decoder features such as speaker pause duration determine the segment boundaries. The punctuation model in DPP then punctuates each of those segments. Without cross-segment look-ahead or the ability to correct previously finalized results, the punctuation model functions within the boundaries of each provided text segment. Consequently, punctuation model performance is highly dependent on the quality of text segments generated by the decoder.

Our past investments have focused on both systems independently - (1) improving decoder segmentation using look-ahead-based acoustic-linguistic features and (2) using neural network architectures to punctuate in DPP. As measured by punctuation-F_1 scores, these investments have improved our punctuation quality. Yet, over-segmentation for slow speakers or irregular pauses is still prominent.

In streaming punctuation, we explore a system that discards decoder segmentation, shifting punctuation decision-making towards a powerful long-context Transformer-based punctuation

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 187-197, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122014

model. Instead of segments, this system emits well-formed sentences, which is much more desirable for downstream tasks like Translation of Speech Recognition output. The proposed architecture also satisfies real-time latency constraints for ASR.

Many works have demonstrated that leveraging prosodic features and audio inputs can improve punctuation quality [1, 2, 3]. However, as we show in our experiments, misleading pauses may undermine punctuation quality and encourage overly aggressive punctuation. It is especially true in scenarios like dictation, where users pause often and unintentionally. Our work shows that text-only streaming punctuation is robust to over-segmentation from irregular pauses and slow speakers.

We make the following key contributions: (1) we introduce a novel streaming punctuation approach to punctuate and re-punctuate ASR outputs, as described in section 3, (2) we demonstrate streaming punctuation's robustness to model architecture choices through experiments described in section 5, and (3) we achieve not only gains in punctuation quality but also significant downstream BLEU score gains on Machine Translation for a set of languages, as demonstrated in section 6.

2. RELATED WORK

Approaches to punctuation restoration have evolved to capture the surrounding context better. Early sequence labeling approaches for punctuation restoration used n-grams to capture context [4]. However, this simple approach becomes unscalable as n grows large, limiting the amount of context used in punctuation prediction.

Classical machine learning approaches such as conditional random fields (CRFs) [5, 6, 7], maximum entropy models [8], and hidden Markov models (HMMs) [9] model complex features by leveraging manual feature engineering. This manual process is cumbersome, and the quality of these features limits the effectiveness of these classical approaches.

Neural approaches mostly displaced manual feature engineering, opting instead to learn more complex features through deep neural models. Recurrent neural networks (RNNs), specifically Gated Recurrent Units (GRUs) and bidirectional Long Short-Term Memory (LSTM) networks, have advanced natural language processing (NLP) and punctuation restoration by specifically modeling long-term dependencies in the text [10, 11, 12, 13, 14]. Prior works have also successfully used LSTMs with CRF layers [15, 16]. Most recently, using Transformers [17] and especially pre-trained embeddings from models such as Bidirectional Encoder Representations from Transformers (BERT) [18] has significantly advanced quality across natural language processing (NLP) tasks. By leveraging attention mechanisms and more complex model architectures, Transformers better capture bidirectional long-range text dependencies for punctuation restoration [19, 20, 21, 22, 23, 24].

3. PROPOSED METHOD

3.1. Punctuation Model

We frame punctuation prediction as a sequence tagging problem. Figure 1 illustrates the end-to-end punctuation tagging and application process. We first tokenize the input segment as byte-pair encoding (BPE) tokens and pass this through a transformer encoder. Next, a punctuation token classification head, consisting of a dropout layer and a fully connected layer, generates token-level punctuation tags. Finally, we convert the token-level tags to word-level and generate

punctuated text by appending each specified punctuation symbol to the corresponding word in the input segment.

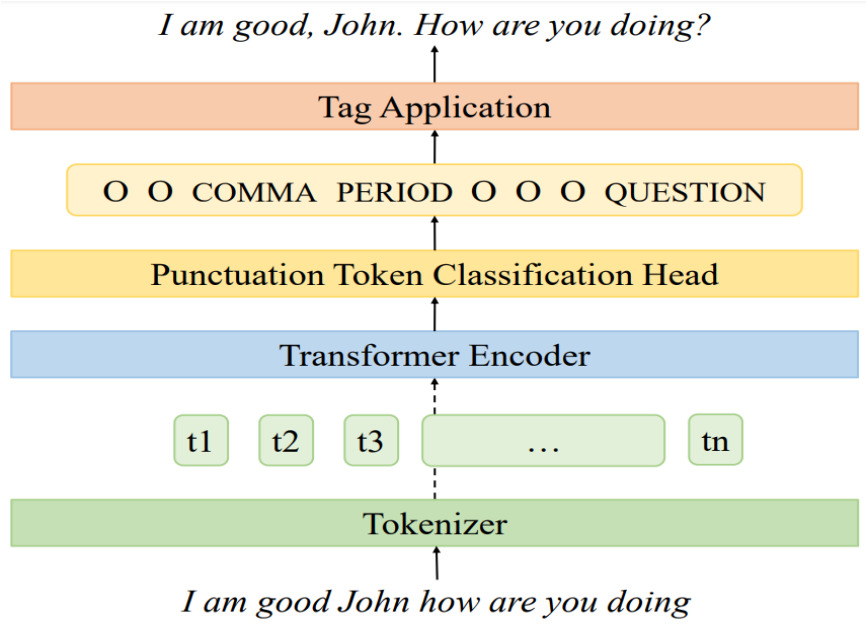


Figure 1. Punctuation tagging model using transformer encoder

3.2. Streaming Decoder for Punctuation

Hybrid ASR systems often define segmentation boundaries using silence thresholds. However, for human2machine scenarios like dictation, pauses do not necessarily indicate ideal segmentation boundaries for the ASR system. In our experience, users pause at unpredictable moments as they stop to think. All A1-4 segments in Table 1 are possible; each is a valid sentence with correct punctuation. Even with a punctuation model, if A4 is the user's intended sentence, all A1-3 would be incorrect. For dictation users, this system would produce a lot of over-segmentation. To solve this issue, we must incorporate the right context across segment boundaries.

Table 1. Examples of possible segments generated by ASR

Id	Segment
A1	It can happen.
A2	It can happen in New York.
A3	It can happen in New York City.
A4	It can happen in New York City, right?

Our solution is a streaming punctuation system. The key is to emit complete sentences only after detecting the beginning of a new sentence. At each step, we punctuate text within a dynamic decoding window. This window consists of a buffer for which the system hasn't yet detected a sentence boundary and the new incoming segment. When at least one sentence boundary is detected within the dynamic decoding window, we emit all complete sentences and reserve any remaining text as the new buffer. This process is illustrated in Figure 2.

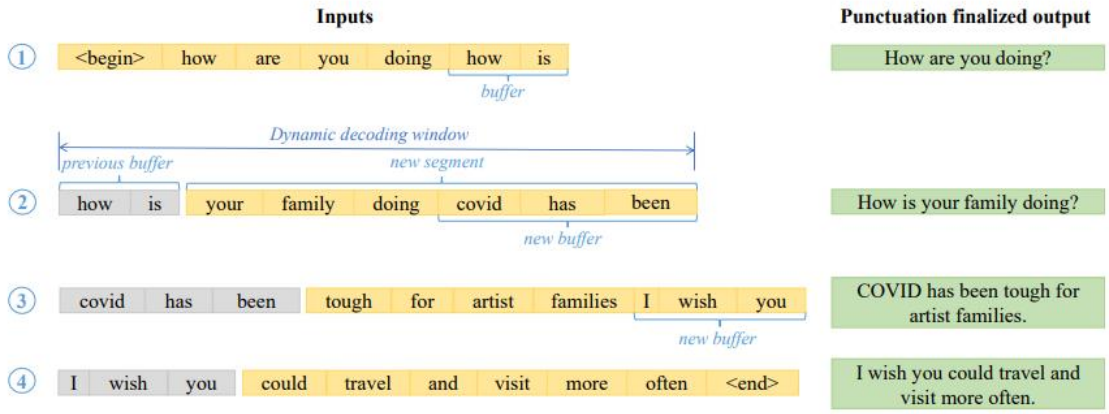


Figure 2. Dynamic decoding window for streaming punctuation

This strategy discards the original decoder boundary and decides the sentence boundary purely on linguistic features. A powerful transformer model that captures the long context well is ideal for this strategy, as dynamic windows ensure that we incorporate enough left and right context before finalizing punctuation. Our approach also meets real-time requirements for ASR without incurring additional user-perceived latency, owing to the continual generation of hypothesis buffers within the same latency constraints. An improvement to this system would be to use a prosody-aware punctuation model that captures both acoustic and linguistic features. That would be a way to re-capture the acoustic cues that we lose by discarding the original segments. However, prosody-aware punctuation models may cause regressions in scenarios such as dictation.

4. DATA PROCESSING PIPELINE

4.1. Datasets

We use public datasets from various domains to ensure a good mix of conversational and written-form data. Table 2 shows the word count distributions by percentage among the sets.

OpenWebText [25]: This dataset consists of web content from URLs shared on Reddit with at least three upvotes.

Stack Exchange: This dataset consists of user-contributed content on the Stack Exchange network.

OpenSubtitles2016 [26]: This dataset consists of movie and TV subtitles.

Multimodal Aligned Earnings Conference (MAEC) [27]: This dataset consists of transcribed earnings calls based on S&P 1500 companies.

National Public Radio (NPR) Podcast: This dataset consists of transcribed NPR Podcast episodes.

Table 2. Data distribution by number of words per dataset

Dataset	Distribution
OpenWebText	52.8%
Stack Exchange	31.5%
OpenSubtitles2016	7.6%
MAEC	6.7%
NPR Podcast	1.4%

4.2. Data Processing

As described in Section 3.1, the transformer sequence tagging model takes spoken-form unpunctuated text as input and outputs a sequence of tags signifying the punctuation to append to the corresponding input word. All datasets consist of punctuated written-form paragraphs, and we process them to generate spoken-form input text and output tag sequences for training.

To preserve the original context, we keep the original paragraph breaks in the datasets and use each paragraph as a training row. We first clean and filter the sets, removing symbols apart from alphanumeric, punctuation, and necessary mid-word symbols such as hyphens. To generate spoken-form unpunctuated data, we strip off all punctuation from the written-form paragraphs and use a Weighted Finite-State Transducers (WFST) based text normalization system to generate spoken-form paragraphs. During text normalization, we preserve alignments between each written-form word and its spoken form. We then use these alignments and the original punctuated display text to generate ground truth punctuation tags corresponding to the spoken-form text.

We set aside 10% or at most 50k paragraphs from each set for validation and use the rest for training.

4.3. Tag Classes

We define four tag categories: comma, period, question mark, and ‘O’ for no punctuation. Each punctuation tag represents the punctuation that appears appended to the corresponding text token. When we convert input word sequences into BPE sequences, we attach the tags only to the last BPE token for each word. The rest of the tokens are tagged with ‘O’.

5. EXPERIMENTS

5.1. Test Sets

We evaluate our punctuation model performance across various scenarios using private and public test sets. Each set contains long-form audio and corresponding written-form transcriptions with number formatting, capitalization, and punctuation. Starting from audio rather than text is critical to highlight the challenges associated with irregular pauses or slow speakers. This prohibits us from using the text-only International Conference on Spoken Language Translation (IWSLT) 2011 TED Talks corpus, typically used for reporting punctuation model performance.

Dictation (Dict-100): This internal set consists of 100 sessions of long-form dictation ASR outputs and corresponding human transcriptions. On average, each session is 180 seconds long.

MAEC: 10 hours of test data taken from the MAEC corpus, containing transcribed earnings calls.

European Parliament (EP-100): This dataset contains 100 English sessions scraped from European Parliament Plenary videos. This dataset already contains English transcriptions, and human annotators provided corresponding translations into seven other languages.

NPR Podcast (NPR-76): 20 hours of test data from transcribed NPR Podcast episodes.

5.2. Experimental Setup

Our baseline system primarily uses Voice Activity Detection (VAD) based segmentation with a silence-based timeout threshold of 500ms. When VAD doesn't trigger, the system applies a segmentation at 40 seconds. The streaming punctuation system receives the input from the baseline system but can delay finalizing punctuation decisions until it detects the beginning of a new sentence.

We hypothesize that streaming punctuation outperforms the baseline system. We test our hypothesis on LSTM and transformer punctuation tagging models. For the LSTM tagging model, we trained a 1-layer LSTM with 512-dimension word embeddings and 1024 hidden units. We used a look-ahead of 4 words, providing limited right context for better punctuation decisions. For the transformer tagging model, we trained a 12-layer transformer with 16 attention heads, 1024-dimension word embeddings, 4096-dimension fully connected layers, and 8-dimension layers projecting from the transformer encoder to the decoder that maps to the tag classes.

Both models are trained with 32k BPE units. We limited training paragraph lengths to 250 BPE tokens and trimmed paragraphs to the last full sentence. All models are trained to convergence.

6. RESULTS AND DISCUSSION

We compare the results of baseline (BL) and streaming (ST) punctuation systems on (1) the LSTM tagging model and (2) the Transformer tagging model. As expected, Transformers outperform LSTMs for this task. Here we test our hypothesis for both model types to establish the effectiveness and robustness of our proposed system. For LSTM tagging models, BL-LSTM refers to the baseline system, and ST-LSTM refers to the streaming punctuation system. Similarly, for Transformer tagging models, BL-Transformer refers to the baseline system, and ST-Transformer refers to the streaming punctuation system.

6.1. Punctuation and Segmentation Accuracy

We measure and report punctuation accuracy with word-level precision (P), recall (R), and F_1-score. Table 3 summarizes punctuation metrics measured and aggregated over three punctuation categories: period, question mark, and comma.

Our customers consistently prefer higher precision (system only acting when confident) over higher recall (system punctuating generously). Punctuation-F_1 does not fully capture this preference. Customers also place higher importance on correctly detecting sentence boundaries over commas. We, therefore, propose segmentation-$F_{0.5}$ as a primary metric for this and future sentence segmentation work. The segmentation metric ignores commas and treats periods and question marks interchangeably, thus only measuring the quality of sentence boundaries. Table 4 summarizes segmentation metrics.

Although our target scenario was long-form dictation (human2machine), we found this technique equally beneficial for conversational (human2human) and broadcast (human2group) scenarios,

establishing its robustness. On average, the ST-Transformer system has a segmentation-$F_{0.5}$ gain of 13.9% and a punctuation-F_1 gain of 4.3% over the BL-Transformer system. Similarly, the ST-LSTM system has a Segmentation-$F_{0.5}$ improvement of 12.2% and a Punctuation-F_1 improvement of 2.1% over the BL-LSTM system. These results support our hypothesis.

6.2. Downstream Task: Machine Translation

We measure the impact of segmentation and punctuation improvements on the downstream task of MT. Higher quality punctuation leads to translation BLEU gains for all seven target languages, as summarized in Table 5. The ST-Transformer system achieves the best results across all seven target languages. On average, the ST-Transformer system has a BLEU score improvement of 0.66 over the BL-Transformer and wins for all target languages. Similarly, the ST-LSTM system has a BLEU score improvement of 0.33 over the BL-LSTM system and wins for 5 out of 7 target languages. These results support our hypothesis.

Table 3. Punctuation results

Test Set	Model	PERIOD			Q-MARK			COMMA			OVERALL			F_1-Gain
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	
Dict-100	BL-LSTM	64	71	67	47	88	61	62	52	57	63	61	61	
	ST-LSTM	77	63	69	67	71	69	60	52	56	68	57	62	0.6%
	BL-Transformer	69	76	72	50	88	64	68	52	59	68	63	65	
	ST-Transformer	81	71	76	82	82	82	69	51	59	74	60	67	2.9%
MAEC	BL-LSTM	68	79	73	46	44	45	63	50	56	65	63	64	
	ST-LSTM	77	70	73	65	45	54	60	51	55	68	60	64	0.0%
	BL-Transformer	71	80	75	50	50	50	65	49	56	67	63	65	
	ST-Transformer	80	78	79	69	46	56	65	48	55	72	62	66	2.4%
EP-100	BL-LSTM	56	71	63	64	62	63	55	47	51	56	58	56	
	ST-LSTM	70	62	66	69	55	61	57	49	53	63	55	59	4.2%
	BL-Transformer	58	76	66	58	70	64	57	49	53	57	61	59	
	ST-Transformer	70	71	71	76	70	73	59	51	55	64	60	62	5.8%
NPR-76	BL-LSTM	72	71	72	71	66	69	65	58	61	69	65	67	
	ST-LSTM	82	71	76	76	69	73	65	59	62	74	66	70	4.0%
	BL-Transformer	76	77	76	76	70	73	68	60	64	72	69	71	
	ST-Transformer	87	79	83	81	75	78	70	61	65	79	71	75	6.0%

Table 4. Segmentation results

Test Set	Model	Segmentation					
		P	R	F_1	F_1-gain	$F_{0.5}$	$F_{0.5}$-gain
Dict-100	BL-LSTM	62	68	65		63	
	ST-LSTM	74	60	66	1.5%	71	12.0%
	BL-Transformer	66	74	70		67	
	ST-Transformer	79	69	73	4.3%	77	13.8%
MAEC	BL-LSTM	66	76	71		68	
	ST-LSTM	76	68	72	1.4%	74	9.5%
	BL-Transformer	69	77	73		70	
	ST-Transformer	79	75	77	5.5%	78	10.9%
EP-100	BL-LSTM	53	67	59		55	
	ST-LSTM	66	58	62	5.1%	64	16.1%
	BL-Transformer	54	72	62		57	
	ST-Transformer	67	68	68	9.7%	67	18.2%
NPR-76	BL-LSTM	71	70	70		71	
	ST-LSTM	81	70	75	7.1%	79	10.9%

	BL-Transformer	74	75	75		74	
	ST-Transformer	85	79	81	8.0%	84	12.5%

6.3. Downstream Task: Machine Translation

We measure the impact of segmentation and punctuation improvements on the downstream task of MT. Higher quality punctuation leads to translation BLEU gains for all 7 target languages, as summarized in Table 5. The ST-Transformer system achieves the best results across all 7 target languages. On average, the ST-Transformer system has a BLEU gain of 0.66 over BL-Transformer and wins for all target languages. Similarly, the ST-LSTM system has a BLEU gain of 0.33 over BL-LSTM system and wins for 5 out of 7 target languages. These results support our hypothesis.

We used Azure Cognitive Services Translator API and compared them with reference translations. For Portuguese (pt) and French (fr), ST-LSTM regresses slightly, while ST-Transformer outperforms BL-Transformer. It is worth noting that ST-Transformer achieves significant gains over BL-Transformer, +1.1 for German (de) and +1.4 for Greek (el). The results suggest that punctuation has a higher impact on translation accuracy for some language pairs. For some language pairs, translation is more robust to punctuation errors.

Table 5. Translation BLEU Results: English audio recognized, punctuated, and translated to 7 languages

Language	Model	BLEU	Gain
de	BL-LSTM	36.0	
	ST-LSTM	36.6	+0.6
	BL-Transformer	36.4	
	ST-Transformer	37.5	+1.1
el	BL-LSTM	39.8	
	ST-LSTM	40.8	+1.0
	BL-Transformer	40.3	
	ST-Transformer	41.7	+1.4
fr	BL-LSTM	41.0	
	ST-LSTM	40.6	-0.4
	BL-Transformer	41.7	
	ST-Transformer	41.8	+0.1
It	BL-LSTM	35.2	
	ST-LSTM	35.5	+0.3
	BL-Transformer	35.4	
	ST-Transformer	35.9	+0.5
pl	BL-LSTM	30.2	
	ST-LSTM	30.9	+0.7
	BL-Transformer	31.1	
	ST-Transformer	31.7	+0.6
pt	BL-LSTM	33.2	
	ST-LSTM	33	-0.2
	BL-Transformer	33.7	
	ST-Transformer	33.9	+0.2
ro	BL-LSTM	39.8	
	ST-LSTM	40.1	+0.3
	BL-Transformer	40.5	
	ST-Transformer	41.2	+0.7

Table 6. BL-Transformer's incorrect punctuation leads to incorrect translations from English to several languages. ST-Transformer correctly punctuates, resulting in correct translations.

Language	**BL-Transformer**	**ST-Transformer**
en	I. Just have to share the view ...	I just have to share the view ...
de	I. Ich muss nur die Ansicht teilen ...	Ich muss nur die Ansicht teilen ...
fr	I. Il suffit de partager le point de ...	Je dois simplement partager le point de ...
it	I. Basti condividere l'opinione ...	Devo solo condividere l'opinione ...

Table 6 presents an example of how incorrect punctuation can lead to downstream consequences in translated outputs. Here BL-Transformer incorrectly punctuates after "I" which results in (1) failure to translate the word, (2) incorrect translations for the subsequent text, and (3) incorrect punctuation in the translations to all languages. ST-Transformer, however, correctly punctuates and thus produces correct translations. This example demonstrates the importance of punctuation quality for downstream tasks such as MT.

7. CONCLUSION

Long pauses and hesitations occur naturally in dictation. We started this work to solve the over-segmentation problem for long-form dictation users. We discovered these elements affect other long-form transcription scenarios like conversations, meeting transcriptions, and broadcasts. Our streaming punctuation approach improves punctuation for a variety of these ASR scenarios. Higher quality punctuation directly leads to higher quality downstream tasks, for instance, improvement in BLEU scores in MT. We also established the effectiveness of streaming punctuation across the transformer and LSTM tagging models, thus establishing the robustness of streaming punctuation to different model architectures.

In this paper, we focused on improving punctuation for hybrid ASR systems. Our preliminary analysis has found that though end-to-end (E2E) ASR systems produce better punctuation out-of-the-box, they still don't fully solve over-segmentation and could benefit from streaming re-punctuation techniques. We plan to present our findings in the future. Streaming punctuation discussed here relies primarily on linguistic features and discards acoustic signals. We plan to extend this work with prosody-aware punctuation models. As we explore our method's effectiveness and potential for other languages, we are also interested in exploring the impact of intonation or accents on streaming punctuation.

REFERENCES

[1] Elizabeth Shriberg, Andreas Stolcke, Dilek HakkaniTur, and Gokhan Tur, "Prosody-based automatic segmentation of speech into sentences and topics," *Speech communication*, vol. 32, no. 1-2, pp. 127–154, 2000.

[2] Madina Hasan, Rama Doddipatla, and Thomas Hain, "Multi-pass sentence-end detection of lecture speech," in Fifteenth Annual Conference of the International Speech Communication Association, 2014.

[3] Piotr Zelasko, Piotr Szyma ˙ nski, Jan Mizgajski, Adrian Szymczak, Yishay Carmiel, and Najim Dehak, "Punctuation prediction model for conversational speech," arXiv preprint arXiv:1807.00543, 2018.

[4] Agustin Gravano, Martin Jansche, and Michiel Bacchiani, "Restoring punctuation and capitalization in transcribed speech," in 2009 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2009, pp. 4741–4744.

[5] Wei Lu and Hwee Tou Ng, "Better punctuation prediction with dynamic conditional random fields," in Proceedings of the 2010 conference on empirical methods in natural language processing, 2010, pp. 177–186.

[6] Xuancong Wang, Hwee Tou Ng, and Khe Chai Sim, "Dynamic conditional random fields for joint sentence boundary and punctuation prediction," in Thirteenth Annual Conference of the International Speech Communication Association, 2012.

[7] Nicola Ueffing, Maximilian Bisani, and Paul Vozila, "Improved models for automatic punctuation prediction for spoken and written text.," in Interspeech, 2013, pp. 3097–3101.

[8] Jing Huang and Geoffrey Zweig, "Maximum entropy model for punctuation annotation from speech.," in Interspeech, 2002.

[9] Yang Liu, Elizabeth Shriberg, Andreas Stolcke, Dustin Hillard, Mari Ostendorf, and Mary Harper, "Enriching speech recognition with automatic detection of sentence boundaries and disfluencies," IEEE Transactions on audio, speech, and language processing, vol. 14, no. 5, pp. 1526–1540, 2006.

[10] Ronan Collobert, Jason Weston, Leon Bottou, Michael, Karlen, Koray Kavukcuoglu, and Pavel Kuksa, "Natural language processing (almost) from scratch," Journal of machine learning research, vol. 12, no. ARTICLE, pp. 2493–2537, 2011.

[11] Xiaoyin Che, Cheng Wang, Haojin Yang, and Christoph Meinel, "Punctuation prediction for unsegmented transcript based on word vector," in Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), 2016, pp. 654–658.

[12] William Gale and Sarangarajan Parthasarathy, "Experiments in character-level neural network models for punctuation.," in INTERSPEECH, 2017, pp. 2794–2798.

[13] Vasile Pais¸ and Dan Tufis¸, "Capitalization and punctuation restoration: a survey," Artificial Intelligence Review, vol. 55, no. 3, pp. 1681–1722, 2022.

[14] Kaituo Xu, Lei Xie, and Kaisheng Yao, "Investigating lstm for punctuation prediction," in 2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP), 2016, pp. 1–5.

[15] Xuezhe Ma and Eduard Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," arXiv preprint arXiv:1603.01354, 2016.

[16] Jiangyan Yi, Jianhua Tao, Zhengqi Wen, Ya Li, et al., "Distilling knowledge from an ensemble of models for punctuation prediction.," in Interspeech, 2017, pp. 2779–2783.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[19] Jiangyan Yi, Jianhua Tao, Zhengkun Tian, Ye Bai, and Cunhang Fan, "Focal loss for punctuation prediction.," in INTERSPEECH, 2020, pp. 721–725.

[20] Yangjun Wu, Kebin Fang, and Yao Zhao, "A contextaware feature fusion framework for punctuation restoration," arXiv preprint arXiv:2203.12487, 2022.

[21] Maury Courtland, Adam Faulkner, and Gayle McElvain, "Efficient automatic punctuation restoration using bidirectional transformers with robust inference," in Proceedings of the 17th International Conference on Spoken Language Translation, 2020, pp. 272–279.

[22] Tanvirul Alam, Akib Khan, and Firoj Alam, "Punctuation restoration using transformer models for high-and low-resource languages," in Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020), 2020, pp. 132–142.
[23] Raghavendra Pappagari, Piotr Zelasko, Agnieszka Mikołajczyk, Piotr Pezik, and Najim Dehak, "Joint prediction of truecasing and punctuation for conversational speech in low-resource scenarios," in 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2021, pp. 1185–1191.
[24] Attila Nagy, Bence Bial, and Judit Acs, "Automatic punctuation restoration with bert models," arXiv preprint arXiv:2101.07343, 2021.
[25] Aaron Gokaslan and Vanya Cohen, "Openwebtext corpus," http://Skylion007.github.io/OpenWebTextCorpus, 2019.
[26] Pierre Lison and Jorg Tiedemann, "Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles," 2016.
[27] Jiazheng Li, Linyi Yang, Barry Smyth, and Ruihai Dong, "Maec: A multimodal aligned earnings conference call dataset for financial risk prediction," in Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 3063–3070.

THE ECONOMIC PRODUCTIVITY OF WATER IN AGRICULTURE BASED ON ORDERED WEIGHTED AVERAGE OPERATORS

José Manuel Brotons Martínez

Economic and Financial Department,
Miguel Hernández University, Elche, Spain

ABSTRACT

Water productivity is one of the main indicators used in agriculture. Price of water change from some regions where the price is free to other with a very high price. When water productivity is measured in Euros, to make comparable the results of the regions where the price is free, we need to obtain a correct measurement, which will require setting a market price for water in areas where no price has yet been set. Therefore, the aim of this paper is to propose new productivity indicators based on fuzzy logic, whereby experts' opinions about the possible price of the use of water as well as the annual variability of agricultural prices can be added. Therefore, the fuzzy willingness to pay (FWTP) and fuzzy willingness to accept (FWTA) methodology will be applied to create an artificial water market. The use of fuzzy logic will allow the uncertainty inherent in the experts' answers to be collected. Ordered Weighted Averaging (OWA) operators and their different extensions will allow different aggregations based on the sentiment or interests reflected by the experts. These same aggregators, applied to the prices of the products at origin, will make it possible to create new indicators of the economic productivity of water. Finally, through an empirical application for a pepper crop in south-eastern Spain we can visualize the importance of the different indicators and their influence on the final results.

KEYWORDS

Water Economic Productivity, OWA, Fuzzy Willingness to Pay, Fuzzy Willingness to Accept.

1. INTRODUCTION

Water is an essential resource for agriculture. However, there are regions where this resource is scarce or practically non-existent, such as the Spanish southeast. Here, the water used comes from other basins or even from desalination plants, which entails a significant increase in costs. This situation contrasts with other regions where water exists in abundance and farmers do not pay for it, or they pay minimal costs for the maintenance of traditional channels. For this reason, it is essential to analyse the economic results of water exploitation [1].

Various indicators measure water productivity. Among these are production per cubic meter consumed or income per cubic meter [2]; but the most representative indicator is the income obtained per euro of water consumption by a farmer. However, it is evident that this information is not always available, so it will be necessary to resort to the use of surveys where the high subjectivity inherent in these estimates must be considered [3].

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 199-207, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122015

The valuation of environmental assets has been carried out using the contingent valuation method, travel costs methodology, or hedonic prices [4-5]. However, many of these studies do not include the subjectivity contained in the answers of the respondents, since they often have an interest in distorting prices based on their interests. For this reason, on the one hand, by using fuzzy logic the uncertainty of this process can be included; and on the other, by using different aggregators such as Ordered Weighting Aggregators (OWA) [6] and their different extensions such as heavy OWA (HOWA) [7], induced OWA (IOWA) [8], probability OWA (POWA) [9] or induced POWA (IPOWA) [10], the interests of the respondents can be adequately treated.

From these aggregators, new indicators of the economic productivity of water will be created. Firstly, this allows us to collect information on water prices in areas where there is no market; and secondly, it allows for an adequate aggregation of the prices of agricultural products in order to reduce their annual variability [11].

Therefore, the aim of this paper is to propose a new methodology for estimating the economic productivity of water in agricultural companies. To do this, first, the price of water will be estimated using the willingness to pay or willingness to accept methodology by means of fuzzy logic and different aggregators. Then, a reference price of production will be obtained using these aggregators, which allows the economic productivity of water to be obtained. Finally, an application of the economic productivity of water in a greenhouse pepper plantation will be carried out.

2. MATERIAL AND METHODS

Definition 1. An OWA operator [6] of dimension n is a mapping of $OWA: R^n \to R$ with an associated weight vector W of dimension *n* such that $\sum_{j=1}^{n} w_j = 1$ and $w_j \in [0,1]$ according to the following expression in which b_j is the *jth* largest element of the collection a_i

$$OWA(a_1, a_2, ..., a_n) = \sum_{j=1}^{n} w_j b_j \quad (1)$$

2.1. Water Demand Function

For a region where the price of water for agriculture is null or merely nominal, such as paying for certain minimum infrastructure maintenance costs without energy consumption and with minimum personnel cost, the application of a cost can lead to saving water. For this reason, a group of experts (farmers or managers of agricultural companies) are asked to assess their willingness to pay a series of prices. To do this, a series of prices will be provided $P = \{P_1, P_2, ..., P_P\}$ in ascending order, with $P_i < P_{i'}, i < i'$. These prices must be valued according to the following linguistic scale in which the membership function is indicated: “totally disagree” (0.00), “strongly disagree” (0.20), “disagree” (0.40), “agree” (0.60), “strongly agree” (0.80) and “totally agree” (1.00).

The expert will be presented with the first price, P_1. If their assessment is “totally disagree”, the price they would be willing to pay would be \$0 m^{-3}. For any other feedback, price P_2 would be offered, and the process would proceed in a similar way. If the answer is “totally disagree”, the process stops, but if any other evaluation is given, then it continues with the following price, and so on.

From this information, it is possible to obtain the price that the expert would be willing to pay to have the water necessary for irrigation. It is important to bear in mind that this is about determining the price they would be willing to pay for their current water needs. To do this, the following algorithm is used.

Step 1. Determination of willingness to pay from expert *j* (WTP_j) can be obtained as the sum of the products of the price increases of each phase in relation to the previous one ($\Delta P_i = P_i - P_{i-1}$), and the membership function μ_{ij} stated by expert j for phase i. It is assumed that $\Delta P_1 = P_1$.

(2)

As a result, the prices that each of the J experts will be willing to pay are obtained $WTP^1 = \{WTP_1^1, WTP_2^1, ..., WTP_J^1\}$.

Step 2. Obtaining the willingness to pay weights. They are assigned according to the different aggregator operators. Each aggregator operator will generate a willingness to pay (*WTP*) price.
Step 3. Water demand function. The demand curve is obtained by combining the sum of the experts' weights, for which we have calculated a *WTP* equal to or higher than the price, P_i, for each price presented to the experts (P_i), abscissa axis.

$$\mu^1(P_i) = \sum_{j=1}^{J} \omega_j \,/\, WTP_j \geq P_i \tag{3}$$

2.2. Water Supply Function

The water supply function is obtained in a similar way to the water demand function. In this case, the current owners of the water will be asked the price they require in exchange for giving up their rights. The evaluation will start at higher prices, and they will be asked about their willingness to accept each price in exchange for permitting the use of their water. In this case, obtaining willingness to accept (WTA) will be obtained from the maximum price and subtracting the membership function of each lower price by the price reduction that it implies in relation to the previous one.

2.3. Equilibrium Price

One group of experts will have expressed their willingness to pay and the other their willingness to accept a price. Membership functions of willingness to pay decrease with P and willingness to accept increases with P. Therefore, the objective is to find a price for which both membership functions are equal. To do this, the following algorithm is presented.

Step 1. Starting from the price series of the demand function with their corresponding membership functions $(P_1, \mu_1^P), (P_2, \mu_2^P), \ldots, (P_n, \mu_n^P)$ and from the supply prices $(P_1, \mu_1^A), (P_2, \mu_2^A), \ldots, (P_n, \mu_n^A)$.
Step 2. Search for the point of intersection, placed between two consecutive prices P_s y P_{s+1} such that $\mu_s^P > \mu_s^A$ and $\mu_{s+1}^P < \mu_{s+1}^A$.
Step 3. Obtaining the intersection point by interpolation between the points indicated in step 2.

$$\left.\begin{aligned}\frac{x-\mu_s^P}{y-P_s}&=\frac{\mu_{s+1}^P-\mu_s^P}{P_{s+1}-P_s}\\ \frac{x-\mu_s^A}{y-P_s}&=\frac{\mu_{s+1}^A-\mu_s^A}{P_{s+1}-P_s}\end{aligned}\right\} \tag{4}$$

It is evident that on many occasions, the experts' opinions are biased out of the fear that they may imply a higher cost in the future (water users) or a lower income (water providers). The use of OWA equilibrium prices will allow these circumstances to be properly incorporated.

2.4. Water Productivity

Water productivity is measured in different ways, such as productivity in Euros per cubic meter of water consumed or income per cubic meter [12]. From an economic point of view, the income from the cost of water is of more interest; that is to say, the quotient between a farm's income (total production, V, multiplied by the average sale price P^V) and the cost of water (cubic meters consumed, W, multiplied by its price P^W.

$$EPW=\frac{V\times P^V}{W\times P^W} \tag{5}$$

The average sales price of each campaign is different, so to obtain a reference price, it is necessary for an aggregation system to be established.

2.5. OWA Operators in Water Economic Productivity

Aggregation only with arithmetic means or weighted means prevents other aspects from being considered. For example, subjective considerations on whether it is convenient for the respondents to obtain high or low results regarding the price of water, which could favour their interests. With respect to product sale price, the problem is that only historical prices are available, so it is necessary to incorporate relevant information not included in said prices [13]. We propose the following operators:

Definition 2. An OWA-EPW operator of dimension $n+m$ is a mapping of $OWA-EPW:R^{n+m}\to R$ with an associated weight vector W of dimension $n+m$ such that $\sum_{j=1}^{n}w_j=1$, $\sum_{j=n+1}^{n+m}w_j=1$ and $w_j\in[0,1]$ according to the expression:

$$OWA-EPW\left(p_V^1,...,p_V^n,p_W^1,...,p_W^m\right)=\frac{OWA\left(p_V^1,...,p_V^n\right)}{OWA\left(p_W^1,...,p_W^m\right)} \tag{6}$$

Definition 3. A HOWA-EPW operator is a mapping of $OWA-EPW:R^{n+m}\to R$ associated with a weighting vector W of dimension $n+m$, such that $w_j\in[0,1]$, $1\le\sum_{j=1}^{n}w_j\le n$ and $1\le\sum_{j=n+1}^{n+m}w_j\le m$ such that:

$$HOWA-EPW\left(p_V^1,...,p_V^n,p_W^1,...,p_W^m\right)=\frac{HOWA\left(p_V^1,...,p_V^n\right)}{HOWA\left(p_W^1,...,p_W^m\right)} \quad (7)$$

Definition 4. An IOWA-EPW operator of dimension $n+m$ is a mapping of $IOWA-EPW:R^{n+m}\times R^{n+m}\to R$ that has an associated weighting vector, W of dimension $n+m$ where $\sum_{j=1}^{n}w_j=1$, $\sum_{j=n+1}^{n+m}w_j=1$ and $w_j\in[0,1]$ such that:

$$IOWA-EPW\left(\left\langle u_{p_V}^1,p_V^1\right\rangle,...,\left\langle u_{p_V}^n,p_V^n\right\rangle,\left\langle u_{p_W}^1,p_W^1\right\rangle,...,\left\langle u_{p_W}^m,p_W^m\right\rangle\right)=\frac{IOWA\left(\left\langle u_{p_V}^1,p_V^1\right\rangle,...,\left\langle u_{p_V}^n,p_V^n\right\rangle\right)}{IOWA\left(\left\langle u_{p_W}^1,p_W^1\right\rangle,...,\left\langle u_{p_W}^m,p_W^m\right\rangle\right)} \quad (8)$$

Definition 5. A POWA-EPW operator of dimension $n+m$ is a mapping of $POWA-EPW:R^{n+m}\to R$ having an associated weighting vector P, where $p_j\in[0,1]$, $\sum_{j=1}^{n}p_j=1$ and $\sum_{j=n+1}^{n+m}p_j=1$ such that:

$$OWA-EPW\left(p_V^1,p_V^2,...,p_V^n,p_W^1,p_W^2,...,p_W^m\right)=\frac{POWA\left(p_V^1,p_V^2,...,p_V^n\right)}{POWA\left(p_W^1,p_W^2,...,p_W^m\right)} \quad (9)$$

Definition 6. An IPOWA-EPW operator of dimension $n+m$ is a mapping of $IPOWA:R^{n+m}\times R^{n+m}\to R$ that has an associated weight vector W of dimension $n+m$, where $\sum_{j=1}^{n}w_j=1$, $\sum_{j=n+1}^{n+m}w_j=1$ and $w_j\in[0,1]$ so that:

$$IPOWA-EPW\left(\left\langle u_{p_V}^1,p_V^1\right\rangle,...,\left\langle u_{p_V}^n,p_V^n\right\rangle,\left\langle u_{p_W}^1,p_W^1\right\rangle,...,\left\langle u_{p_W}^m,p_W^m\right\rangle\right)=\frac{IPOWA\left(\left\langle u_{p_V}^1,p_V^1\right\rangle,...,\left\langle u_{p_V}^n,p_V^n\right\rangle\right)}{IPOWA\left(\left\langle u_{p_W}^1,p_W^1\right\rangle,...,\left\langle u_{p_W}^m,p_W^m\right\rangle\right)} \quad (10)$$

Expressions (6) to (10) can have different combinations. In particular, the proposed definitions can be used or simplified by assuming that the OWA and its extensions are only applied to water prices or agricultural sales prices.

3. EMPIRICAL APPLICATION

The aim was to analyse the water productivity of one-hectare plantation of Lamuyo peppers under greenhouse in 2021. Average production was 35,530 kg ha^{-1} and water consumption was estimated at 8,200 m^{-3}. Since the plantation had unlimited access to water at practically zero cost, we wanted to know the economic productivity of the water. To do this, experts were asked for their willingness to pay and to accept.

Table I shows the weighting vectors for the OWAs and HOWAs, the induced variable (number of cultivated hectares), expert number, and the probability assigned to each of the eight that act as requestors and each of the six that act as bidders. The induced variable for bidders was their responsibility in the firm they worked for, ranging from 1 to 10. In both cases, $\beta=0.4$ has been considered.

Table I. Weighting vectors, experts (Exp), and probabilities (p).

Demand					Supply				
OWA	**HOWA**	**Induced**	**Exp**	**p**	**OWA**	**HOWA**	**Induced**	**Exp**	**p**
0.25	0.30	15	1	0.02	0.26	10	9	1	0.15
0.20	0.25	10	2	0.25	0.22	8	7	2	0.20
0.18	0.20	8	3	0.10	0.17	6	4	3	0.12
0.15	0.15	7	4	0.20	0.15	4	3	4	0.21
0.08	0.10	4	5	0.18	0.12	3	6	5	0.13
0.07	0.10	6	6	0.12	0.08	2	10	6	0.19
0.05	0.05	5	7	0.08					
0.02	0.05	3	8	0.05					

Table II shows the opinions about each price. Expert 1 (consumer) was not willing to pay any amount, but expert 2 totally agreed up to the price of €0.10 m^{-3}, and strongly agreed to the price of €0.15 m^{-3}, etc. For the experts who acted as bidders, they started at higher prices, so they first agreed to accept a price of up to €0.15 m^{-3}, but for €0.10 m^{-3}, they only totally agreed, and for €0.05 m^{-3}, they agreed.

Table II. Willingness to Pay (WTP) and Willingness to Accept (WTA)

Demand	**Price**							
Expert	**0.05**	**0.1**	**0.15**	**0.2**	**0.25**	**0.3**	**0.35**	**WTP**
1	0	0	0	0	0	0	0	0.00
2	1	1	0.8	0.6	0.2	0	0	0.18
3	1	0.8	0.8	0.2	0.2	0	0	0.15
4	1	1	1	1	1	1	0.4	0.32
5	1	1	0.8	0.6	0.4	0.2	0	0.20
6	0.8	0.4	0.2	0	0	0	0	0.07
7	1	1	0.4	0	0	0	0	0.12
8	1	1	1	0.8	0.6	0.6	0.2	0.26
Supply								**WTA**
1	0	0.8	1	1	1	1	1	0.06
2	0	0	0	0	0.2	0.8	1	0.25
3	0	0	0	0	0.2	0.6	0.8	0.27
4	0.4	1	1	1	1	1	1	0.03
5	0	0	0	0	0	0	0.4	0.33
6	0.2	0.4	0.4	0.8	1	1	1	0.11

The demand and supply curves were obtained from these opinions. As can be seen, the equilibrium points (which are analytically shown in Figure 1) differ greatly depending on the aggregators used. The membership function corresponding to the demand aggregator IOWA is systematically lower than the rest, being the highest those corresponding to the OWA and HOWA aggregators. On the contrary, the supply function of the aggregators IPOWA and IOWA present the highest values, being the lowest the HOWA operator. In addition to those described, multiple combinations could be made between an aggregator for demand and another for supply. In general, they range from €0.14 to €0.27 m^{-3}.

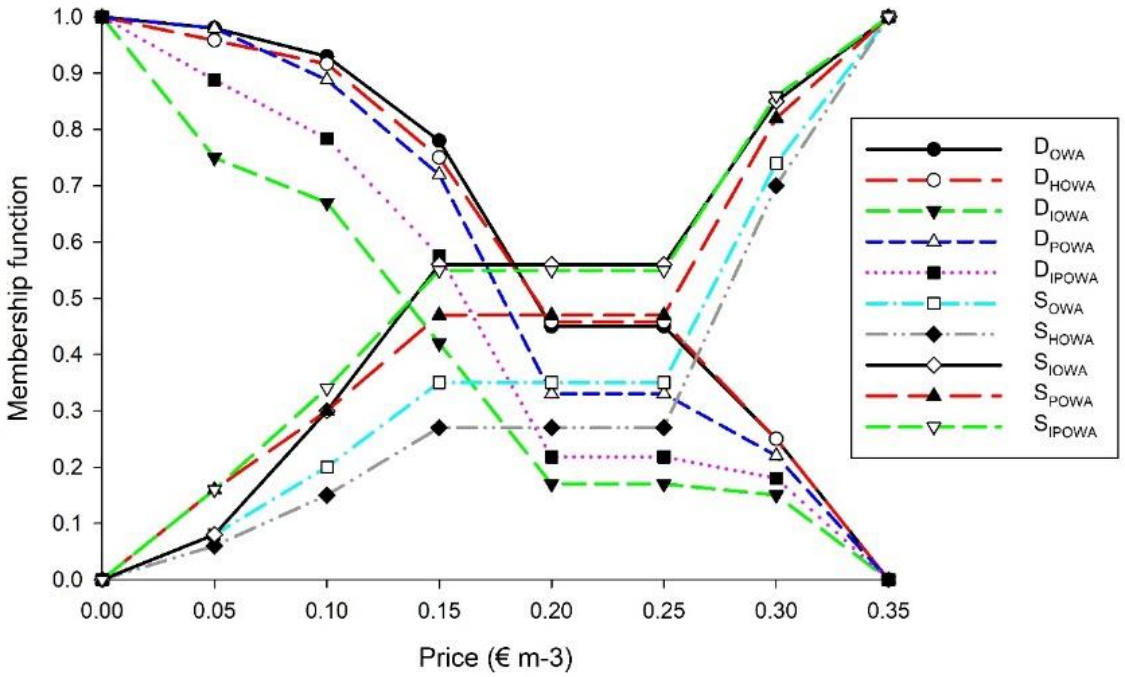


Figure 1. Water demand (D) and supply (S) functions for Ordered Weighting Average (OWA), Heavy OWA (HOWA), Induced OWA (IOWA), Probability OWA (POWA), and Induced POWA (IPOWA).

The historical prices of peppers are a known variable, and they are shown in Table III [14]. Their variability makes the inference of future prices complex. In fact, prices have ranged from €0.56 kg^{-1} in 2014 to €1.00 kg^{-1} in 2021. Table III shows the weighting vectors for each aggregator and the probabilities assigned to each expert. $\beta = 0.4$ has been considered.

Table III. Annual Prices of Pepper and Weighting Vectors and Probabilities for Each Expert

Year	Prices	OWA	HOWA	Induced	Expert	Probability
2021	1.00	0.17	20	0.86	1	0.06
2020	0.86	0.15	15	0.87	2	0.14
2019	0.87	0.14	13	0.73	3	0.11
2018	0.73	0.12	12	0.69	4	0.07
2017	0.69	0.11	8	0.89	5	0.08
2016	0.89	0.10	6	0.82	6	0.16
2015	0.82	0.08	5	0.56	7	0.17
2014	0.56	0.06	4	0.78	8	0.08
2013	0.78	0.04	3	0.54	9	0.09
2012	0.54	0.03	3	0.68	10	0.04

Finally, the equilibrium prices of water, the average prices of peppers, and the economic productivity of water are given (Table IV). The results show a wide dispersion, which demonstrates the importance of this study since the value of productivity is highly influenced by the type of aggregator used.

Table IV. Water and Pepper Prices, and Productivity by Aggregators

Operator	Water price (€ m^{-3})	Pepper price (€ kg^{-1})	Productivity
OWA-EPW	0.26	0.77	12.97
HOWA-EPW	0.26	0.84	13.67
IOWA-EPW	0.14	0.79	25.25
POWA-EPW	0.18	0.79	18.77
IPOWA-EPW	0.15	0.81	23.06

The use of different aggregators allows obtaining several values for water and pepper price. In particular, water price ranges from €0.26 m^{-3} for OWA-EPW and HOWA-EPW and €0.15 m^{-3}, and pepper price from €0.77 to €0.84 kg^{-1}. As a result, and according to pepper production and water consumption, productivity ranges from 12.97 to 23.06. The option for one or another aggregator will depend, among others, on the optimism and pessimism degree. Anyway, this methodology generalizes the productivity ratio, even for the case in which the water is free.

4. CONCLUSIONS

The aim of this paper is to introduce a new formulation in the traditional calculation of the economic productivity of water, using OWA-EPW, HOWA-EPW, IOWA-EPW, POWA-EPW, and IPOWA-EPW. The advantage of these operators is that they provide new ways of aggregating prices and expert opinions. This makes it possible to visualise the potential risks of certain crops not becoming profitable because it takes many years at exiguous prices. This form of aggregation allows for simplifications if circumstances change. The results or the empirical application show how productivity can almost double depending on the aggregator used. Results of the empirical application show a range of productivity values from 12.97 to 23.06. The choice for one or another will depend on the optimism or pessimism degree.

ACKNOWLEDGEMENTS

This work was financed by Ayudas a la Investigación Vicerrectorado de Investigación, Universidad Miguel Hernández, 2022.

REFERENCES

[1] J.M. Cámara-Zapata, J.M. Brotons-Martínez, S. Simón-Grao, J.J. Martinez-Nicolás & F. García-Sánchez, "Cost-benefit analysis of tomato in soilless culture systems with saline water under greenhouse conditions," J. Sci. Food Agr, vol. 99(13), pp. 5842-5851, 2019

[2] J. López-Marín, M. Romero, A. Gálvez, F. M. Del Amor & M.C. Piñero & J. M. Brotons-Martínez, "The Use of Hydromulching as an Alternative to Plastic Films in an Artichoke (Cynara cardunculus cv. Symphony) Crop: A Study of the Economic Viability," Sustainability, vol. 13(9), pp. 5313, 2021.

[3] J. López-Marín, A. Gálvez, F.M. del Amor, M.C. Piñero & J.M. Brotons-Martínez, "The cost-benefits and risks of use raffia made of biodegradable polymer: the case of pepper and tomato crops under green-houses," Horticulturae, vol. 8(2), pp. 133, 2022.

[4] J.M. Brotons, R. Chávez & G. Ruiz-Sevilla, "OWA aggregation to determine the water equilibrium price in the tancítaro area," IT in Industry, vol. 9(3), pp. 1-10, 2021

[5] R. Nallathiga & R. Paravasthu, "Economic value of conserving river water quality: Results from a contingent valuation survey in Yamuna river basin, India,". Water Policy, vol 12(2), pp. 2449, 2010.

[6] R.R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decision making," IEEE Trans. Syst. Man Cybern, vol. 18(1), pp. 183–190, 1988.

[7] R.R. Yager, "Heavy OWA operators," Fuzzy Optim. Decis. Mak, vol. 1(4), pp. 379–397, 2002.

[8] R.R. Yager, D.P. Filev, "Induced ordered weighted averaging operators," IEEE Trans. Syst. Man Cybern., Part B, Cybern, vol. 29(2), pp. 141–150, 1999.

[9] J.M. Merigó, "Probabilities in the OWA operator," Expert Syst. Appl, vol. 39(13), pp. 11456–11467, 2012.
[10] J.M. Merigó, "Decision-making under risk and uncertainty and its application in strategic management," J. Bus. Econ. Manag, vol. 16(1), pp. 93–116, 2015
[11] E. León-Castro, L.F. Espinoza-Audelo, J.M. Merigó, E. Herrera-Viedma & F. Herrera, "Measuring volatility based on ordered weighted average operators: The case of agricultural product prices" Fuzzy Sets and Systems, vol. 422, pp. 161–176, 2021.
[12] R. Salvador, A. Martínez-Cob, J. Cavero & E. Playán, "Seasonal On-Farm Irrigation Performance in the Ebro Basin (Spain): Crops and Irrigation Systems," Agric. Water Manag, vol. 98, pp. 577–587, 2011.
[13] M. Olazabal-Lugo, E. Leon-Castro, L.F. Espinoza-Audelo, J.M. Merigo & A.M. Gil Lafuente, "Forgotten effects and heavy moving averages in exchange rate forecasting," Econ. Comput. Econ. Cybern. Stud. Res, vol. 53(4), 2019.
[14] CARM, 2021. https://caamext.carm.es/esamweb/faces/vista/seleccionPrecios.jsp

AUTHOR

José Manuel Brotons is Economist (1991, Alicante University) and PhD in Business Administration (2003). From 1999 he is a member of the Economic and Financial Studies Department of Miguel Hernández University in Elche (Spain). He has an 7 h-index (Web of Science), with 29 documents listed in Web of Science (10 in Q1 and 11 in Q2) and 168 citations (accessed on 18/6/2022). In the period 2012-2022, J.M. Brotons has contributed in 47 scientific publications (32 of them published in indexed journals in the Journal Citations Report), 29 national and international congress communications, 11 books; he has participated in 25 projects and 2 contracts; 6 end-of-degree projects. He has obtained two recognised six-year research periods by CNAI and four five-years teaching periods. He is expert in financial valuation and economic analysis applied to the agriculture. In 2003 he defended the thesis "Financing of the Wastewater Treatment Company Sector. An Operational Approach". One of his interest research is the economic analysis of the agriculture production. Another interest research is the company financial valuation as a result of the use of new methodologies such as the innovations in fuzzy logic and the analysis of the quality system in the enterprises.

MULTI-SINK CONVERGECAST PROTOCOL FOR LARGE SCALE WIRELESS SENSOR NETWORKS

Gokou Hervé Fabrice Diédié[1], Koigny Fabrice Kouassi[2]
and Tchimou N'Takpé[2]

[1]Laboratory of Mathematics and Computer Science,
Université Peleforo Gon Coulibaly, Korhogo, Côte d'Ivoire
[2]Laboratory of Mathematics and Computer Science,
Université Nangui Abrogoua, Abidjan, Côte d'Ivoire

ABSTRACT

Wireless sensor nodes are designed to collect information about their immediate environment. Once gathered, such data are forwarded via a multi-hop communication pattern to a remote gateway, also known as the sink. This process referred to as the convergecast may often require several sinks in order to improve network efficiency and resilience. Provided that load among the latter nodes are well balanced and packet losses are mitigated. This paper aims to design such a protocol by combining clustering, path-vector routing and sinks' duty cycle scheduling schemes to help balance load and minimize message overhead. Simulation results proved that this solution outperforms DMS-RP (Dynamic Multi-Sink Routing Protocol), a recent state-of-the-art contribution, in terms of delay minimization, packet delivery and network lifetime enhancement.

KEYWORDS

Wireless Sensor Networks, Clustering, Convergecast, Multi-sink, Protocol, Scheduling.

1. INTRODUCTION

Wireless Sensor Networks (WSNs) comprise small detection devices called sensor nodes and a central equipment referred to as the sink. The latter acts like a gateway to a third-party transportation network. WSNs' applications are encountered in domains like ecology, security, transportation, to industry, health etc. [1-3].

One of the most critical operations in such networks is convergecast, namely, the data gathering and forwarding process [4]. Typical WSNs architectures involve usage of a single sink. However, many applications including those deployed in harsh environments require a large number of sensors [5]. Therefore, the latter have to face long routing paths and congestions particularly in sink's neighbourhood [6]. The resulting energy wastes and packet losses quickly become detrimental to network lifetime and efficiency. A common technique to address these issues is to add extra sinks to the network [7]. Nevertheless, to really benefit this strategy, it is mandatory to distribute network load among sinks and to always find the optimal gateway for each sensor.

Unfortunately, most solutions found in the literature leverage assumptions and schemes that are restrictive for real-world applications; e.g. fixed size clusters, interference-free links etc.

David C. Wyld et al. (Eds): SIGEM, MLTEC, SEAPP, ITCON, NATL, FUZZY, CSEA - 2022
pp. 209-225, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122016

In this paper, we propose to seamlessly combine clustering, path-vector routing and scheduling schemes to assign the proper sink to each sensor node while balancing network load. The resulting protocol helps minimize delay, mitigate packet losses and prolong network lifetime.

Our main contributions are as follows:

- a hop limitation clustering strategy that mitigates message overhead and network flooding;
- a route discovery processes that simultaneously consider links' asymmetry, interferences, and nodes' congestion level;
- a fully-distributed loop-free path-vector routing scheme that minimizes delays between sensor nodes and sinks;
- a scheduling strategy that helps balance loads for both sensor nodes and sinks.

The rest of the paper is organized as follows: Section 2 surveys the related contributions; then, the proposed solution is detailed in Section 3; the performance evaluation process, the results, and discussions are presented in Sections 4 and 5 followed by conclusion in Section 6.

2. RELATED WORK

Convergecast schemes commonly use routing techniques where shortest paths from sensor nodes to sinks are constructed relying on different metrics (Euclidean distance, number of hops, link quality, nodes' energy level, number of possible retransmissions etc.) [8]. The resulting solutions can be reactive or proactive when the route discovery process is launched respectively on-demand or in advance. From a topological point of view, these protocols are generally classified as flat and hierarchical ones [9]. In the first category, solutions leverage well-known tree construction schemes such as the Shortest Path Tree (SPT), Minimum Spanning Tree (MST), and random tree (RDCT) to find relevant routes between sensor nodes and sinks. However, in large scale wireless sensor networks, where scalability is at stake, hierarchical techniques like clustering are indispensable. This scheme consists of grouping nodes around a leader referred to as the Cluster Head (CH) [10]. The latter may be chosen randomly or not; but often according to different criteria (energy level, degree, location, etc.) [11-15]. LEACH by Heinzelman *et al.* [16] is one of the oldest protocols in this category. This protocol has inspired a huge number of contributions in the past two decades [17]. Unfortunately, these solutions are generally single-sink oriented. Relatively few multi-sink solutions have been recently proposed.

Masdari and Naghiloo[18] suggested a distributed fuzzy logic-based sink selection scheme to cope with congestions. Each sink declares its load to their neighbours. The nearest uncongested sink is then selected. Nevertheless, this solution is dedicated to only one-hop networks. Rajput and Kumaravelu [19] used a similar approach. They applied in contrast, a fuzzy c-means algorithm to balance the size of clusters and to optimize the number of sinks and their locations in the region of interest. However, this solution is limited to applications with a deterministic sink deployment and is not scalable; since the latter deployment and the clustering process are centralized to the distant base station.

Singh and Nagaraju [20-21] proposed to create routes by constructing a Wiener minimum spanning tree based on an Artificial Bee Colony optimization scheme. Regrettably, this strategy is not really scalable and requires knowing sinks' positions.

As for Wang and Su [22], they designed an algorithm based on breadth-first search strategy to create 2-disjoint path between each sensor node and the sinks using an edge colouring scheme. Nevertheless, this solution requires estimating Euclidean distance from a node to the sink. Calculation of this distance is costly.

Mukherjee *et al.* [23] used sensor nodes with 3-sector antennas to locate sinks. Based on the Euclidean distance from the sink, the deployment area is divided into 3 regions where nodes can respectively send data directly to the nearest sink (direct routing), via a one-hop (two-hop routing) or using multi-hop routing. Regrettably, this scheme does not cope with load balancing and needs specific sensor nodes.

With protocol GeoM, Leão and Felea [24] considered using a geographical-based solution that linearly combines Euclidean distance and energy level metrics to determine the next hop. Data are sent to available sinks using a two-step multicast communication scheme. The first and the second step consist of selecting respectively target sinks and candidate forwarders. After calculating the weighted metrics of all couples (candidates and sinks), intersecting decisions are detected and eliminated to avoid packets duplications. However, this solution does not scale since it is not really distributed. Additionally, it requires that every node is aware of both its own geographic position and that of all the sinks. Gathering such information is very costly.

Yildiz [25] proposed a Mixed Integer Programming (MIP) model aimed to maximize network lifetime for underwater WSNs. Unfortunately, this solution is not scalable since it is centralized and the proposed model leverages parameters difficult and costly to collect, such as total number of flows generated at a node and transferred over a specific link during the network, otal number of flows collected at a sink etc.

Fu *et al.* [26] advised a field-driven paradigm to make routing decisions. In this scheme network is abstracted into an electrostatic or magnetic field etc. where packets behave like objects that can be"attracted" by the sinks. This solution is the first field-driven protocol that uses multi-path (multi-hop) routing and considers impact of external environment. The path selection leverages different metrics referred to as the potential fields, namely the depth, residual energy and the environmental information of sensor nodes. These metrics help respectively estimate the Euclidean distance while preventing messages to traverse dangerous areas and nodes with poor energy level. This strategy introduces fault-tolerance in the network but struggles to balance sinks' load.

Liu *et al.* [27] developed a solution to schedule traffic and select optimal paths by considering delays and load balancing. Each sensor node tries to find the shortest path to each sink then records the results and the transmitted traffics obtained. The latter are finally scheduled to balance the load of sinks. This solution does not scale since during initial step, it requires each sensor node to construct a path to each sinks.

Onwuegbuzie *et al.* [28] suggested a three-step strategy. Firstly, the amount of traffic or network load/task to assign to each sink is estimated, and then whenever a task is to be executed, the real-time load of each sink is computed. Finally, the real-time load to total weight ratio of each participating sink is calculated. The current task is thus assigned to the sink having the least load to weight ratio. Regrettably, this scheme is centralized hence, not scalable.

Hassani *et al.* [29] presented a RPL-based [30] solution where hierarchical paths to select are evaluated via two linearly combined metrics namely, number of hops and the RSSI (Received Signal Strength Indicator). Each node selects a possible parent then computes its own rank from that of this neighbour using the combined metrics. The proper parent is chosen after comparing the expected transmission count for this neighbour metric to the obtained rank. However, in harsh environments, RSSI-based metrics are often a misleading.

Daas *et al.* [31] designed a distributed multi-hop cluster-based routing protocol where paths are selected using both hop count and link state via its SNR (Signal-to-Noise-Ratio). This protocol

formally uses the *Sink-As-Cluster-Head* strategy to help balance the load of both sensor nodes and sinks. However, this solution requires fixed size clusters; thus, can hardly be applied to randomly deployed networks, e.g. to monitor phenomena in harsh environments.

3. PROPOSED SOLUTION

This section aims at presenting our scheme. We discuss our motivations, objectives and assumptions, then detail our solution referred to as MSCP (Multi-Sink Convergecast Protocol).

3.1. Motivations and Objectives

As discussed in previous sections, large scale wireless sensor networks often require using both clustering and deployment of several sinks. Surprisingly, few state-of-the-art cluster-based multi-sink convergecast protocols have been proposed so far. *Sink-As-Cluster-Head* is a technique often used in these solutions to enhance energy waste mitigation particularly in sinks' neighbourhood [32]. In this category, DMS-RP by Daas *et al.* [31] is one of the recent contributions that consider both link sate and asymmetry. However, this protocol ignores sinks load balancing, does not cope with network flooding and is only dedicated to deterministically deployed networks.

We believe that to further increase throughput, network lifetime, and its pervasiveness, a convergecast protocol, besides scalability, must consider interferences and link asymmetry, schedule duty cycles of both sensor nodes and sinks, be loop-free, while minimizing message overhead. This work is aimed to address these issues.

3.2. Assumptions

We assume that:
- nodes are equipped with an omni-directional radio;
- each node has a unique identifier (ID);
- nodes are uniformly and randomly deployed in the area of interest;
- nodes' connection is modelled as an UDG (Unit Disk Graph);
- each node can assess distances through the received signal strength or a specific localization protocol;
- the number of sensor nodes is higher than the number of sinks.

3.3. Description

Let V denote the set of nodes in the deployment zone and E the set links between them. Formally, $E=\{(u,v)\in V\times V|\ d(u,v) < (r_u + r_v)\ \}$ where $d(u,v)$ is the Euclidean distance between nodes u and v ; while r_u and r_v respectively denote the communication ranges of u and v. MSCP consists of two stages, namely the clustering phase and the convergecast tree construction one. This protocol is a distributed asynchronous based scheme that uses message passing communication model.

Clustering-based convergecast protocols require a head node to be elected inside each cluster. This issue referred to as the leader election problem is a well-known topic in distributed systems design. In addition, clustering is a proved NP-complete problem [33].

3.3.1. Cluster Formation

This is a two-step cyclic phase starting with neighbour discovery and possibly followed by the creation of a new cluster. Each node *u* must to give a unique integer number to each neighbour. This number is randomly chosen in interval $\left[1;\left|N(u)\right|\right]$; where $N(u)$ is the set of its neighbours.

We use a scheme similar to the one proposed for the clustering process inside the CONSTRUCT protocol [34]. Note that each cluster is identified by its Cluster Head ID (CHID); hence, each node knows the cluster it belongs to. Besides, we use a *Sink-As-CH* strategy, i.e. each sink creates a cluster in its *k*-hop neighbourhood; *k* is given as a parameter such as *k = max_hop_count.*

CHs have a limited service time of which duration is fixed as a parameter. Therefore, after its *mandate* a CH must abandon its status and launch a new election process in its *k*-hop neighbourhood. Moreover, when two CHs move next to each other, the one with less cluster mates will eventually lose its role and become a new cluster mate; ties are randomly broken (see [34] for details). Figure 1 depicts the different statuses applicable to each node in a section of the network after the cluster formation phase.

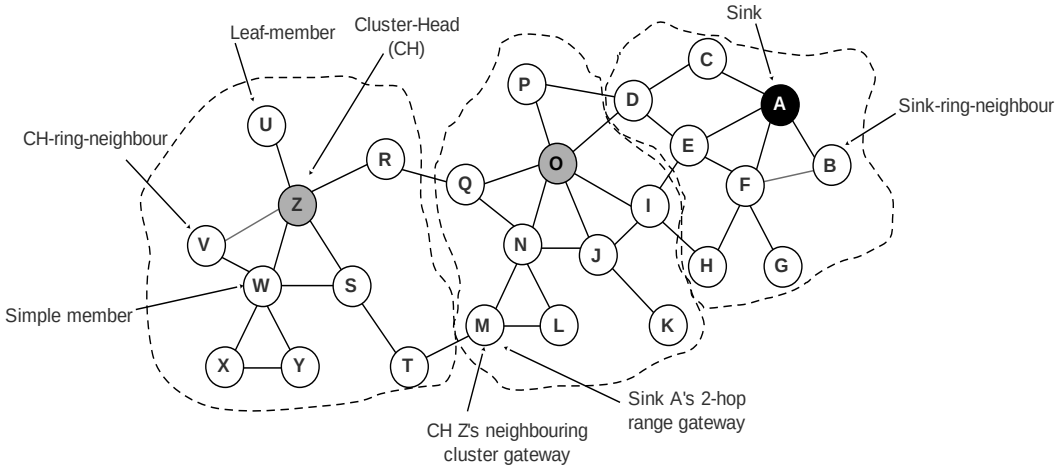


Figure 1. Statuses of nodes used by MSCP's clustering scheme

3.3.2. Intra-cluster convergecast sub-tree construction

After the creation of a new cluster, the CH has to construct the intra-cluster data forwarding sub-tree and interconnect it to the inter-cluster infrastructure. To do so, CH *u* broadcasts in its *k*-hop neighbourhood a TREE-REQ message containing its ID then triggers a timer (*TREE-timer*) and waits for any response during $max\_hop\_count \times t_{wait}(u)$ seconds. This duration is calculated using Equation (1); where rtt(.) denotes the round-trip-time (in seconds) experienced with node *v* during neighbour discovery. *N(u)* is node *u*'s neighbourhood.

$$t_{wait}(u) = \max\left\{\text{rtt}(v); v \in N(u)\right\} \quad (1)$$

When receiving a TREE-REQ message from a neighbour u, a node v must increment the value of the *hop-count* field to 1 then estimates the transmission delay $\delta_{u,v}$ using Equation (2); where $C_{u,v}$, $L_{u,v}$ and S_u respectively denote the capacity (i.e. maximum transmission rate) of link (u,v),

the received packet length on this link and the queuing delay (sojourn time) of this message inside neighbour u .

Note that S_u is provided by the latter neighbour via a dedicated field in the message.

$$\delta_{u,v} = \frac{L_{u,v}}{C_{u,v}} + S_u \tag{2}$$

$$C_{u,v} = W_{u,v} \times \log_2(1 + SINR_{u,v}) \tag{3}$$

As for, $C_{u,v}$ it is estimated using Equation (3); where $W_{u,v}$ and $SINR_{u,v}$ respectively denote the bandwidth of link (u,v) and the Signal Interference plus Noise Ratio experienced on this link.

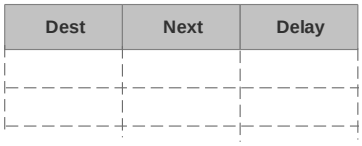


Figure 2. Structure of convergecast table

Node v then increments the *total-delay* field with the estimated delay $\delta_{u,v}$. Node v must also insert its *ID* into the *first-forwarder* field if the latter node is a neighbour of the TREE-REQ message's sender.

A node must respond to a TREE-REQ message by broadcasting a TREE-ACK message if this node is a neighbouring gateway (i.e. is affiliated to another CH but has at least one neighbour inside the sender's cluster), is a leaf-mate (i.e. a cluster member with only one neighbour) or is a CH-ring-neighbour (i.e. has exactly two neighbours of which one is the latter CH) see Figure 1. TREE-ACK message contains its *I*D, its CH*'s* ID, the value of the *first-forwarded* field provided by the received TREE-REQ message, a list of IDs of its neighbouring sinks, the delay to each of these sinks, and the estimated *total delay* of the TREE-REQ message.

Note that TREE-REQ messages are forwarded by a node if the sender is a cluster mate (CHID =ID), this message is received for the first time, and its *hop_count* field's value is below *max_hop_count* +1. Moreover, to mitigate the protocol overhead, TREE-REQ messages are never sent back to a forwarder.

When forwarding a TREE-REQ message to its neighbours, a node sends the list of pairs (unique number, neighbour's ID). The recipient neighbour then concatenates the given unique number to content of the message's *cast_vector* field.

TREE-ACK messages are forwarded following the same rules applied to TREE-REQ except that the CH's ID is not compared to the forwarder ID.

After receiving a TREE-ACK message sent by a neighbouring cluster gateway, the CH node u must update its convergecast table (as shown in Figure 2) by calculating the delay to each destination (i.e. sink) via Equation (4); where $\delta_{u,x}$, $\delta_{u,s}$ and $\delta_{s,x}$ respectively denote delays from CH u to sink x, from u to the sender s and from the sender s to sink x .The *first forwarder*

provided by the TREE-REQ message is the next-hop. For the same destination only the lowest delay so far experienced is kept in this table.

$$\delta_{u,x} = \delta_{u,s} + \delta_{s,x} \quad (4)$$

TREE-ACK messages sent by other eligible nodes help CH to determine total delay from the senders.

After *TREE-timer* expiration, the CH must reply to each received TREE-ACK message, via the *first forwarder* node, by sending a CAST message where the field *type* is respectively set to 1 if the receiver (i.e. sender of the TREE-ACK message) is a neighbouring cluster gateway and 0 otherwise.

Each CAST message also contains the list of sinks discovered by the CH and the delays to them. Before, sending this message, CH copies into the *cast_vector* field, content of the corresponding field of the received TREE-ACK.

When receiving a CAST message with *type* field equals to 1, a node u must update its convergecast table by calculating its transmission delay $\delta_{u,x}$ to each sink x as expressed by Equation (5); where $\delta_{u,x}$, $\delta_{s,x}$ and $\delta_{s,u}$ respectively denote delays from node u to sink x, from the sender s to sink x and from the sender s to node u.

The latter analyzes content of the *cast_vector* field, extracts the unique number of which position equals value of the *hop_count* field. The neighbour that was assigned the latter unique number is chosen as the next hop.

$$\delta_{u,x} = \delta_{s,x} - \delta_{s,u} \quad (5)$$

However, when a leaf-member or a CH-ring-neighbour (see Figure 1 for illustration) receives a CAST message, it must respond by sending a CAST-ACK. Except that a CH-ring-neighbour must send a CAST-ACK only after receiving a CAST message forwarded by its non-CH neighbour.

Note that CAST and CAST-ACK messages are forwarded following the same rules used for TREE-REQ and TREE-ACK ones.

Each CAST-ACK message contains the list of sinks and delays to reach them with its *cast_vector* field embedding the forwarders' unique number. All these information were extracted from the CAST message previously sent by the CH.

After receiving a CAST-ACK message a node u must use Equation (4) to update its convergecast table by calculating the delay $\delta_{u,x}$ to reach each sink x via the CH s; then extracts from the *cast_vector* field, the unique number of which position equals value of the *hop_count* field; so as to choose as next hop, the neighbour that was assigned the latter unique number.

3.3.3. Inter-cluster convergecast sub-tree construction

Note that only sinks are allowed to periodically (i.e. at a beginning of a new duty-cycle) broadcast a HELLO message to clusters located at most *max_cluster_hop_count* hops.

After sending such a HELLO message a sink *u* must trigger a timer (*HELLO-timer*) and wait for any response from next-hop gateways during $max\_cluster\_hop\_count \times t_{wait}(u)$ seconds (see Equation (1)). HELLO messages are forwarded like TREE-ACK messages.

Once a node receives a new HELLO message, if the forwarder's CHID is different from its own CHID, the latter node must increment the *cluster_hop_count* field value to 1 and forward this message only if the value of this field is lower than *max_cluster_hop_count*. This node must also insert its ID into the *first-forwarder* field if it is a neighbour of the message's sender (i.e. the sink). Only a range gateway (i.e. last cluster hop gateway), a leaf-mate (i.e. member with only one neighbour), sink–ring-neighbour (i.e. a node that has exactly two neighbours of which one is the sender sink), must respond by broadcasting a HELLO-ACK. (See Figure 1 for illustration). Note that a HELLO-ACK message is forwarded only by nodes that have previously received a HELLO message from the same sink.

After receiving a new HELLO-ACK from a neighbour u, a node v estimates the transmission delay $\delta_{u,v}$ using Equation (2); then increments the *total-delay* field with the latter estimated delay $\delta_{u,v}$ and finally forwards the message.

After *HELLO-timer* expiration, the sink must reply to each received HELLO-ACK message, via the *first forwarder* node, by sending a CAST message where the field *type* is set to 0. This message also contains the total delay value extracted from the HELLO-ACK. Such CAST messages are forwarded using the same rules applied to the HELLO ones.

When a node receives a CAST message, it must reply by sending a CAST-ACK. Except that sink-ring-neighbours must send a CAST-ACK message only after receiving a CAST message forwarded by its non-sink neighbour. CAST-ACK messages must also embed the total delay value extracted from the CAST ones. Besides, the *cast_vector* field of each CAST-ACK message contains the values provided by its corresponding CAST message.

Therefore, when receiving a CAST-ACK message sent by node s, a node u can update its convergecast table by calculating its transmission delay $\delta_{u,x}$ to the sink x using Equation (5) where $\delta_{u,x}$, $\delta_{s,x}$ and $\delta_{s,u}$ respectively denote delays from node u to sink x, from the sender s to sink x and from the sender s to node u. Then the latter must analyzes content of the *cast_vector* field, in order to extract the unique number of which position equals value of the *hop_count* field. The neighbour that was assigned the latter unique number is chosen as the next hop.

To help balance the load, every sink must shift to sleep mode after t_{wake} seconds, i.e. if its energy consumption ratio ξ reaches a threshold ρ defined as a parameter. ξ is calculated using Equation (6) where E_i and E_f respectively denote sink's energy at the beginning and at the end of its current duty-cycle. Before being inactive, sinks must alert their cluster mates by broadcasting a SLEEP message. The latter will be forwarded like HELLO messages.

$$\xi = \frac{E_i - E_f}{E_i} \tag{6}$$

This idle state will last t_{sleep} seconds. The latter duration is determined using Equation (7) where α denotes the number of times a sink has been active; $\beta > 0$ is the Weibull distribution shape parameter; R is uniformly and randomly chosen in the interval $]0;1[$.

$$t_{sleep} = \frac{1}{\alpha} \times \ln(\frac{1}{R})^{\frac{1}{\beta}} \quad (7)$$

It is also noteworthy to mention that a CH must also trigger a timer (*BUILD-timer*) to help reconstruct the intra-cluster part of the convergecast tree after t_{build} seconds. This duration is determined using Equation (8); where $\overline{t_i}$ denotes the mean time between wake-ups of the i^{th} neighbouring sink while λ is given as a parameter.

$$t_{build} = \frac{\min(\overline{t_i}, \overline{t_{i+1}}, ..., \overline{t_n})}{\lambda}, \quad \lambda \geq 2 \quad (8)$$

Note that $\overline{t_i}$ is calculated then included in HELLO messages sent by sink *i*. Sink's range gateways also spread this information along with the list of active sinks when sending TREE-ACK messages.

3.3.4. Data sending process

After the convergecast tree is constructed, each cluster mate can send its sensed data to the nearest available sink. To do so, the CH or the sink may send a dedicated schedule via CAST messages.

Note that, to mitigate energy waste, data often needs to be aggregated before being sent out of the cluster. In such a case, CH could information about this aggregation tree along with the possible scheduling scheme. The design of such a process is beyond the scope of this work. Nevertheless, any of the solutions that exist in the literature could be used.

Figure 2 depicts algorithm used by MSCP.

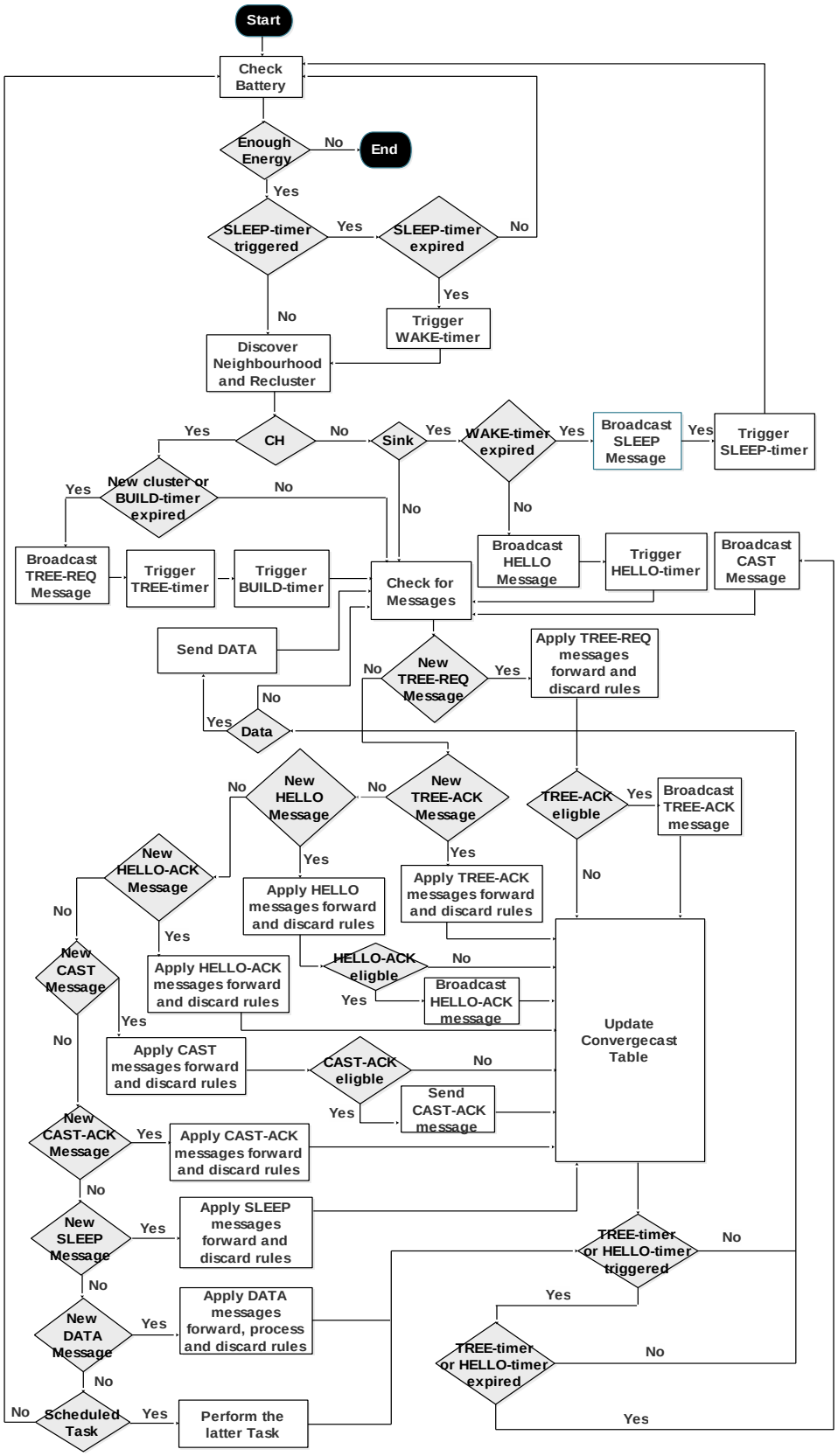


Figure 2. Flowchart of MSCP

4. EXPERIMENTAL SET-UP

In this section, we detail the extensive simulation campaign we carried out. Experiments were conducted using OMNeT++ 6.0 simulator [35]. We used the energy consumption model proposed by Heinzelman *et al.* [16]; the other parameters are summarized in Tables 1 - 4. The results were compared to those we obtained with DMS-RP (Dynamic Multi-Sink Routing Protocol) Multi-parent version + SNR (Signal Noise Ratio) proposed by Daas *et al.* [31].

Table 3 presents parameters we used to randomly and uniformly vary link quality; namely, PRR (Packet Reception Ratio), SNR (Signal to Noise Ratio), SINR (Signal Interference plus Noise Ratio) and LQI (Link Quality Indicator) [36]. Tables 2 and 4 are inspired by the specifications of the IEEE 802.15.4 standard [37].

We evaluated the ability of MSCP and DMS-RP to efficiently transfer data the sinks through three metrics, namely, the *average end-to-end delay*, the *packet delivery ratio*, and the *network lifetime* [31]. To this end, we randomly and uniformly deployed sensors and sinks varying their population as described in Table 1, respectively using, a 100 and a 2 steps scale so that the sink-to-sensor ratio is 0.02. We specifically investigated how the latter populations influenced these metrics. To vary link quality, we used the uniform distribution to randomly change parameters described in Table 3. Each 2.5s, 30% to 50% of nodes were randomly chosen to send data.

This experiment was replicated 50 times for each variation of the number of nodes. Results were averaged with a 95% confidence interval. The experiment started after all the nodes were deployed and was ended according to the network lifetime definition i.e., when a sensor or a sink depleted is energy.

Table 1. Simulation general parameters

Parameter	**Value**
deployment zone	1000 m X 1000 m
number of sensors	100 - 1000
number of sinks	2 - 20
sensors' transmission ranges	127 m
sinks' transmission ranges	250 m
max_hop_count	2
max_cluster_hop_count	2
sensors' initial energy	0.2 J
sinks' initial energy	20 J
self-discharge per second	0.1 μJ
E_{elec}	50 nJ/bit
e_{fs}	10 nJ/bit/m^2
e_{amp}	0.0013 nJ/bit/m^4
d_0	87 m
length of data	2000 bits
β Weibull distribution shape	3
$W_{u,v}$ bandwith of a link (u,v)	2.4 GHz
ρ Threshold of energy consumption ratio	0.01

Table 2. Transition state delay

	R_x (µs)	T_x (µs)	Sleep (µs)	Idle
R_x	-	1	194	-
T_x	1	-	194	-
Sleep	5	5	-	-
Idle	-	-	-	-

Table 3. Link quality parameters

	PRR	SNR (dBm)	SINR(dBm)	LQI
Excellent	1	]40; 60]	]30; 40]	]106; 255]
Good	]0.75; 1[	]25; 40]	]15; 30]	]102; 106]
Medium	]0.35; 0.75]	]15; 25]	]5; 15]	]80; 102]
Poor	[0; 0.35]	[0; 15]	[0; 5]	[0; 80]

Table 4. Transition state energy consumption

	R_x (mW)	T_x (mW)	Sleep (mW)	Idle
R_x	-	62	62	-
T_x	62	-	62	-
Sleep	1.4	1.4	-	1.4
Idle	-	-	1.4	-

5. RESULTS AND DISCUSSIONS

This section is aimed to analyse and explain results we have obtained from experiments we have described in the previous sections.

5.1. Average end-to-end delay

Figure 3 depicts the effect of number of nodes on packets delivery delay. Indeed, delays increase according to network size regardless of the evaluated protocol, for networks with less than 500 sensors then decrease.

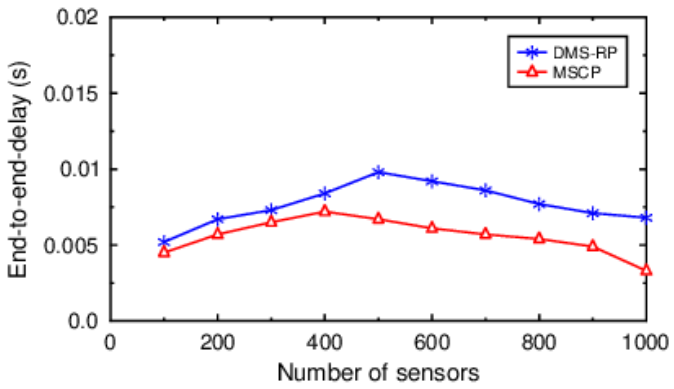


Figure 3. Number of sensors vs. End-to-end delay

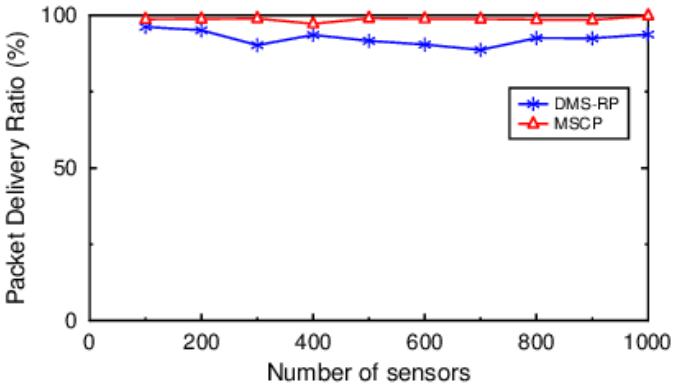


Figure 4. Number of sensors vs. PDR

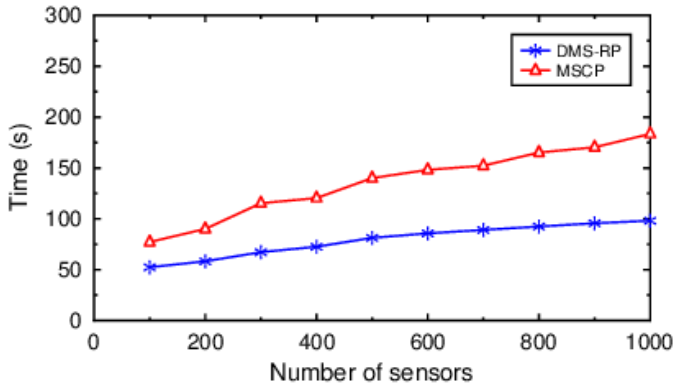
However, one can notice that MSCP provides the best results. This is due to the delay-based route selection process of MSCP. Indeed, sensors always choose the nearest sink considering lowest transmission and queuing delays. Unlike, DMS-RP, to make decisions MSCP leverages the interference (via the SINR metric) and the level of congestion of the intermediary nodes, especially in dense networks (number of sensors > 500) with high levels of interference.

5.2. Packet Delivery Ratio

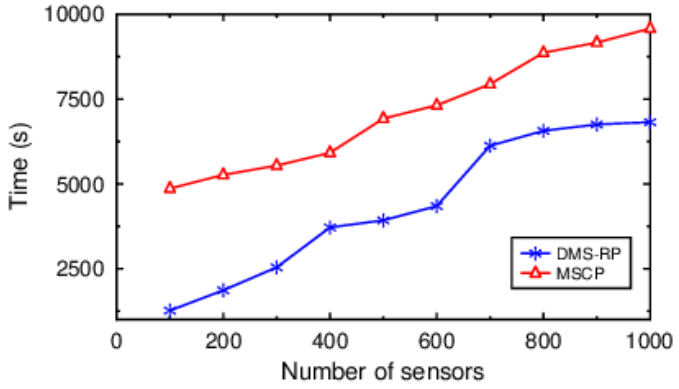
Figure 4 suggests that both protocols yield ratios higher than 90%. However, the values obtained by MSCP are around 98% despite the number of sensors. This is also due to its path update policy and to its SINR + congestion level-based link quality estimation scheme. Indeed, during the route selection phase, MSCP strives to keep only the links with the lowest transmission and queuing delays in both intra and inter-cluster topologies. This results in the minimization of packets losses.

5.3. Network lifetime

Figures 5 show that irrespective of the protocol used, network lifetime increases with its size; since high node degrees help to provide more alternative routes. However, MSCP yields the best results. This is mainly due to packet losses hence the retransmissions reduction scheme used during link selection phase (as discussed above); then to CHs' re-elections and sinks' duty-cycle scheduling. Furthermore, unlike DMS-RP, MSCP significantly mitigates message overhead hence energy waste, by reducing the range of its signaling messages and their number thanks to the forwarding rules applied. Additionally, the path-vectors piggybacking scheme used for CAST-ACK and TREE-REQ messages, helps avoid energy-consuming broadcasts.



a)



b)

Figure 5. Number of sensors vs. Network lifetime:
a) Until a node dies b) Until a sink dies

Using MSCP, only TREE-REQ and HELLO messages are respectively used to update the intra-cluster and inter-cluster routes. DMS-RP instead, uses messages that can unnecessary flood the network and are unable to prevent loops.

6. CONCLUSIONS

In this paper we presented MSCP, a convergecast protocol for large-scale multi-sink WSNs. We combined clustering technique to path vector-based routing and both sensor and sink scheduling schemes. The *Sink-As-CH* scheme is applied to save sinks' neighbours from being involved clustering elections. The resulting fully-distributed and proactive strategy helped balance the load between sinks, and mitigate message complexity. Unlike many state-of-the-art contributions, during route selection phase, transmission and queuing delays are estimated, considering interferences through a SINR (Signal-to-Interference-plus-Noise-Ratio)-based metric and nodes' level of congestion. Simulations show that MSCP provides high packet delivery ratios, lowest latencies and enhances network lifetime.

As a future work, we will formally focus on the impact of mobility of both sensors and sinks on the performance of MSCP. We also plan to provide this solution with security, data aggregation tree construction and traffic scheduling schemes.

REFERENCES

[1] Kayhan Erciyes, (2019) Case study: Environment monitoring by a wireless sensor network, Computer Communications and Networks,.Springer International Publishing.

[2] Dionisis Kandris, Christos Nakas, Dimitrios Vomvas, and Grigorios Koulouras, (2020) "Applications of wireless sensor networks: An up-to-date survey", Applied System Innovation, Vol. 3, No.1, pp.14.

[3] Kamal Gulati, Raja Sarath Kumar Boddu, Dhiraj Kapila, Sunil L. Bangare, Neeraj Chandnani, and G. Saravanan, (2022) "A review paper on wireless sensor network techniques in internet of things (IoT)", Materials Today: Proceedings, Vol. 51, pp. 161–165.

[4] Deepak Sharma, Amritesh Ojha, and Amol P. Bhondekar. "Heterogeneity consideration in wireless sensor networks routing algorithms: a review".,The Journal of Supercomputing, Vol.75, No.5, pp.:2341–2394.

[5] Essam H. Houssein, Mohammed R. Saad, Kashif Hussain, William Zhu, Hassan Shaban, and M. Hassaballah, (2020) "Optimal sink node placement in large scale wireless sensor networks based on harris hawk optimization algorithm". IEEE Access, Vol.8 pp. 19381–19397.

[6] Jobish John, Gaurav S. Kasbekar, and Maryam Shojaei Baghini (202), Maximum lifetime convergecast tree in wireless sensor networks. Ad Hoc Networks, Vol.120, No.102564.

[7] Aysegul Tuysuz Erman, Thijs Mutter, Lodewijk van Hoesel, and Paul Havinga, (2009), A cross-layered communication protocol for load balancing in large scale multi-sink wireless sensor networks, 2009 International Symposium on Autonomous Decentralized Systems, IEEE.

[8] Tarunpreet Kaur and Dilip Kumar, (2019) "Computational intelligence-based energy efficient routing protocols with QoS assurance for wireless sensor networks: a survey", International Journal of Wireless and Mobile Computing, Vol.16, No.2 172.

[9] Louie Chan, Karina Gomez Chavez, Heiko Rudolph, and Akram Hourani, (2020) "Hierarchical routing protocols for wireless sensor network: a compressive survey". Wireless Networks, Vol. 26, No.5, pp.:3291–3314.

[10] Piyush Rawat and Siddhartha Chauhan, (2021), "A survey on clustering protocols in wireless sensor network: taxonomy, comparison, and future scope", Journal of Ambient Intelligence and Humanized Computing, Vol. 12, No. 7, pp. 181-190.

[11] Kowsalya. R and B. Rosiline Jeetha, (2019), "A survey on distributed clustering techniques for wireless sensor networks", International Journal of Computer Sciences and Engineering, Vol.7, No. 3, pp. 404–409.

[12] Muhammad Noman Riaz, (2018), "Clustering algorithms of wireless sensor networks: A survey", International Journal of Wireless and Microwave Technologies, Vol. 8, No. 4, pp.40–53.

[13] Anagha Rajput and Vinoth Babu Kumaravelu (2020), "A cluster leader selection algorithm to enhance the lifetime of scalable wireless sensor networks", .Journal of Circuits, Systems and Computers, Vol.30, No. 03, 2150056.

[14] Amin Shahraki, Amir Taherkordi, Oystein Haugen, and Frank Eliassen, (2020) "Clustering objectives in wireless sensor networks: A survey and research direction analysis", Computer Networks, Vol. 180, No. 107376.

[15] Jain Amutha, Sandeep Sharma, and Sanjay Kumar Sharma, (2021) "Strategies based on various aspects of clustering in wireless sensor networks using classical, optimization and machine learning techniques: Review, taxonomy, research findings, challenges and future directions". Computer Science Review,Vol. 40, No. 100376, pp 1-43.

[16] Wendi Beth Rabiner Heinzelman, Anathan.P. Chandrakasan, and Hari Balakrishnan (2002) "An application-specific protocol architecture for wireless microsensor networks", IEEE Transactions on Wireless Communications, Vol. 1, No. 4, pp. 660–670.

[17] Ikram Daanoune, Baghdad Abdennaceur, and Abdelhakim Ballouk. (2021) "A comprehensive survey on LEACH-based clustering routing protocols in wireless sensor networks". Ad Hoc Networks, Vol. 114 No.102409.

[18] Mohammad Masdari and Farah Naghiloo. (2017) "Fuzzy logic-based sink selection and load balancing in multi-sink wireless sensor networks", Wireless Personal Communications, Vol. 97, No.2, pp.2713–2739.

[19] Anagha Rajput and Vinoth Babu Kumaravelu. (2020) "Fuzzy-based clustering scheme with sink selection algorithm for monitoring applications of wireless sensor networks", Arabian Journal for Science and Engineering, Vol.45, No.8, pp. 6601–6623.

[20] Amit Singh and Aitha Nagaraju.(2018). Energy efficient optimal path based coded transmission for multi-sink and multi-hop WSN, Second International Conference on Green Computing and Internet of Things (ICGCIoT).IEEE.
[21] Amit Singh and A. Nagaraju. (2020), Low latency and energy efficient routingaware network coding-based data transmission in multi-hop and multi-sink WSN. Ad Hoc Networks, Vol. 107, No. 102182.
[22] Fu Wang and Qinggang Su. (2018) A nonintersecting multipath routing protocol for wireless sensor networks, 5th International Conference on Systems and Informatics (ICSAI), IEEE.
[23] Sankar Mukherjee, Ruhul Amin, and G. P. Biswas , (2019) " Design of routing protocol for multisink based wireless sensor networks", Wireless Networks, Vol. 25, No. 7, pp. 4331–4347.
[24] Lucas Leão and Violeta Felea. (2019) Latency and network lifetime trade-off in geographic multicast routing for multi-sink wireless sensor networks. Mobile, Secure, and Programmable Networking, Springer International Publishing.
[25] Huseyin Ugur Yildiz (2019) Utilization of multi-sink architectures for lifetime maximization in underwater sensor networks, 2nd IEEE Middle East and North Africa COMMunications Conference (MENACOMM), IEEE.
[26] Xiuwen Fu, Yongsheng Yang, and Octavian Postolache, (2021) "Sustainable multipath routing protocol for multi-sink wireless sensor networks in harsh environments", .IEEE Transactions on Sustainable Computing, Vol. 6, No.1, pp. 168–181.
[27] Yang Liu, Lei Liu, Zhongmin Yan, and Jia Hu, (2021) The algorithm of multisource to multi-sink traffic scheduling. 17th International Conference on Mobility, Sensing and Networking (MSN). IEEE.
[28] Innocent Uzougbo Onwuegbuzie, Shukor Abd Razak, and Arafat Al-Dhaqm, (2021) Multi-sink load-balancing mechanism for wireless sensor networks, IEEE International Conference on Computing (ICOCO). IEEE.
[29] Abdelhadi Eloudrhiri Hassani, Aicha Sahel, and Abdelmajid Badri. (2021).Amalgamation of novel objective function and multi-sink solution for a reliable RPL in high traffic monitoring applications, Digital Technologies and Applications, Springer International Publishing.
[30] Muhammad Omer Farooq, Cormac J. Sreenan, Kenneth N. Brown, and Thomas Kunz, (2017) "Design and analysis of RPL objective functions for multigateway ad-hoc low-power and lossy networks", Ad Hoc Networks, Vol. 65, pp.78–90.
[31] Mohamed Skander Daas, Salim Chikhi, and El-Bay Bourennane, (2021) "A dynamic multi-sink routing protocol for static and mobile self-organizing wireless networks: A routing protocol for internet of things". Ad Hoc Networks, Vol.117, No.102495.
[32] Aarti Jain and B.V.R. Reddy. (2014) Sink as cluster head: An energy efficient clustering method for wireless sensor networks, International Conference on Data Mining and Intelligent Computing (ICDMIC), IEEE.
[33] Michael Randolph Garey, David Stifler Johnson, (1979) Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman,CA, USA.
[34] Gokou Hervé Fabrice Diédié, Boko Aka, and Michel Babri, (2019) "Self-stabilising hybrid connectivity control protocol for WSNs", IET Wireless Sensor Systems, Vol. 9, No. 1, pp. 6–24.
[35] Varga, A.: 2022. Available at 'https://omnetpp.org', accessed on August 2022.
[36] Carlo Alberto Boano, Marco Antonio Zuniga, Thiemo Voigt, Andreas, Willig, and Kay Romer, (2010) The triangle metric: Fast link quality estimation for mobile wireless sensor networks, Proceedings of 19th International Conference on Computer Communications and Networks, IEEE.
[37] Bruno Bougard, Francky Catthoor, Denis Clarke Daly, Anantha Chandrakasan, and Wim Dehaene, (2005) Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: Modeling and improvement perspectives, Design, Automation and Test in Europe, IEEE.

AUTHORS

Gokou Hervé Fabrice Diédié received his Ph.D. in Computer Science at Université Nangui Abrogoua (UNA) in Abidjan, Côte d'Ivoire in September 2018. He received his MS in Computer Science from the same university in 2012. His research interests include distributed algorithms, self-organization in ad hoc networks, intelligent transportation systems. He is currently a Professor and Researcher in Computer Science and a member of Laboratory of Mathematics and Computer Science at Université Peleforo Gon Coulibaly, Korhogo Côte d'Ivoire.

Koigny Fabrice Kouassi received his MS in Computer Science at Université Nangui Abrogoua (UNA) in Abidjan, Côte d'Ivoire in November 2021. He received his BS in Computer Science from the same university in 2018. His research interests include distributed algorithms and hoc networks. He is member of and Researcher in Computer Science and a member of Laboratory of Mathematics and Computer Science at UNA.

Tchimou N'Takpé received his Ph.D. in Computer Science at Nancy University, France, in January 2009. He received his MS in engineering from Ecole Nationale d'Electricité et de Mécanique at Nancy, France, in 2005. His research interests include distributed and parallel algorithms and Networking. He is currently a Professor and Researcher in Computer Science and a member of Laboratory of Mathematics and Computer Science at Université Niangui Abrogoua in Abidjan, Côte d'Ivoire.

AUTHOR INDEX