Computer Science & Information Technology 162

Artificial Intelligence, Soft Computing and Applications

David C. Wyld,
Dhinaharan Nagamalai (Eds)

Computer Science & Information Technology

- 6th International Conference on Artificial Intelligence, Soft Computing and Applications (AISCA 2022), January 29~30, 2022, Copenhagen, Denmark
- 6th International Conference on Networks and Communications (NET 2022)
- 3rd International Conference on Data Mining and NLP (DNLP 2022)
- 3rd International Conference on Big Data & Health Informatics (BDHI 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: 978-1-925953-60-2
DOI: 10.5121/csit.2022.120201- 10.5121/csit.2022.120207

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

Preface

6th International Conference on Artificial Intelligence, Soft Computing and Applications (AISCA 2022), January 29~30, 2022, Copenhagen, Denmark, 6th International Conference on Networks and Communications (NET 2022), 3rd International Conference on Data Mining and NLP (DNLP 2022), 3rd International Conference on Big Data & Health Informatics (BDHI 2022) was collocated with 6th International Conference on Artificial Intelligence, Soft Computing and Applications (AISCA 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 AISCA 2022, NET 2022, DNLP 2022 and BDHI 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, AISCA 2022, NET 2022, DNLP 2022 and BDHI 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 AISCA 2022, NET 2022, DNLP 2022 and BDHI 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)

Program Committee Members

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Artificial Intelligence Community (AIC)

Soft Computing Community (SCC)

Digital Signal & Image Processing Community (DSIPC)

6th International Conference on Artificial Intelligence, Soft Computing and Applications (AISCA 2022)

6th International Conference on Networks and Communications (NET 2022)

3rd International Conference on Data Mining and NLP (DNLP 2022)

3rd International Conference on Big Data & Health Informatics (BDHI 2022)

STRIDE RANDOM ERASING AUGMENTATION

Teerath Kumar, Rob Brennan and Malika Bendechache

CRT AI and ADAPT, School of Computing, Dublin City University, Ireland

ABSTRACT

This paper presents a new method for data augmentation called Stride Random Erasing Augmentation (SREA) to improve classification performance. In SREA, probability based strides of one image are pasted onto another image and also labels of both images are mixed with the same probability as the image mixing, to generate a new augmented image and augmented label. Stride augmentation overcomes limitations of the popular random erasing data augmentation method, where a random portion of an image is erased with 0 or 255 or the mean of a dataset without considering the location of the important feature(s) within the image. A variety of experiments have been performed using different network flavours and the popular datasets including fashion-MNIST, CIFAR10, CIFAR100 and STL10. The experiments showed that SREA is more generalized than both the baseline and random erasing method. Furthermore, the effect of stride size in SREA was investigated by performing experiments with different stride sizes. Random stride size showed better performance. SREA outperforms the baseline and random erasing especially on the fashion-MNIST dataset. To enable the reuse, reproduction and extension of SREA, the source code is provided in a public git repository https://github.com/kmr2017/stride-aug.

KEYWORDS

Data Augmentation, Image Classification, Erasing Augmentation.

1. INTRODUCTION

Since the advent of deep learning, it has improved classification performance in a wide variety of domains including image classification [1, 2, 3], audio classification [4,5,6] and text classification [8,9,10]. The performance of deep learning algorithms is evaluated by model generalization. To prevent overfitting, two popular techniques of model generalization are used: model regularization i.e. batch normalization [11], dropout [12] and data augmentation [14, 15, 16]. There are many state-of-the-art techniques for data augmentation and random erasing data augmentation [14] is one of them. In random erasing, a randomly sized patch in a random position in an image is erased with 0 or 255 or the mean of the dataset. Though it is effective, there is a high probability that significant features of the image can be erased which deteriorates model performance. The effect of this deterioration is shown in Figure 1, where a random part of the image is erased, consequently the augmented image has lost many significant features of the original input. Thus, this augmented image when used as training data leads to bad model generalization rather than improving the performance. To overcome this issue, this paper proposes a new data augmentation named Stride Random Erasing Augmentation (SREA), where random size strides (or slices) of one image are pasted onto another image with a random probability. We investigate if SREA provides the benefits of random erasing augmentation while preserving the good features. In this work, we use the terms model and network interchangeably. Our work has the following contributions:

David C. Wyld et al. (Eds): AISCA, NET, DNLP - 2022
pp. 01-09, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120201

- We propose a novel augmentation approach, named Stride Random Erasing Augmentation (SREA), it does not only provide random erasing (as images are mixed in random stride way) but also preserves the significant features
- Unlike conventional augmentation techniques, features are not lost as in random erasing
- We perform a series of image classification experiments on standard datasets using our proposed approach and it outperforms both baseline and random erasing-based classification.
- We investigate the effect of different stride sizes (small, random and large) and the effect of different augmentation probability values.
- We provide full source code for SREA in an open repository: https://github.com/kmr2017/stride-aug

The rest of the paper is structured as follows: Section 2 describes the closely related work, Section 3 describes the algorithm of proposed SREA method, Section 4 explains the experimental setup and results, and finally, Section 5 provides conclusions and ideas for future work.

2. Related Work

The objective of model generalization is to prevent the model from overfitting. The two main techniques used for model generalization are: regularization [11, 12, 21, 22, 23] and data augmentation [13, 15, 14, 16, 17, 18, 20].

2.1. Regularization

Dropout [12] is a regularization technique, where hidden and visible neural network neuron probabilities are randomly set to zero and are dropped. In Ba, J. [21] an adaptive dropout is proposed where the probability of a hidden neuron, that is to be discarded, is calculated using a binary belief network. DropConnect [22] randomly selects the subsets of weights and sets them to zero instead of disconnecting the neurons. In the stochastic pooling [23], parameter free activations are selected during training from a multinomial distribution and used with state-of-the-art regularization techniques.

2.2. Data Augmentation

Data augmentation is one of the prominent techniques used for regularization [14]. Data augmentation is used to increase training dataset size and thereby increase classification test accuracy with less original data. There are many techniques for data augmentations, i.e., translation, rotation and addition of salt-and-pepper noise, etc. Among them, the three most popular and close to the proposed approach are flipping [15], random cropping [13] and random erasing [14]. Flipping is simply a manipulation where the object is flipped horizontally or vertically or both. Random cropping selects a random patch from an image and resizes it to the original image size. In random erasing [14], a random part of an image is erased during the training. In random image cropping and patching [16], patches from four images are extracted and mixed to create a new image and the labels are mixed correspondingly. This work [17] analyzes traditional data augmentation techniques i.e., rotating, cropping, zooming, histogram based methods and others. Recently a new perspective of data augmentation named mathematical framework was proposed in [18]. It explains data augmentation benefits and the authors proved that data augmentation is equivalent to performing the average operation on a certain group that does not vary in data distribution. The proposed SREA does not only provide random erasing (as images are mixed in random stride way) but also preserves the significant features (features are not lost as in random erasing [14]). So it is useful for models to learn these features, resulting in a good regularization effect.

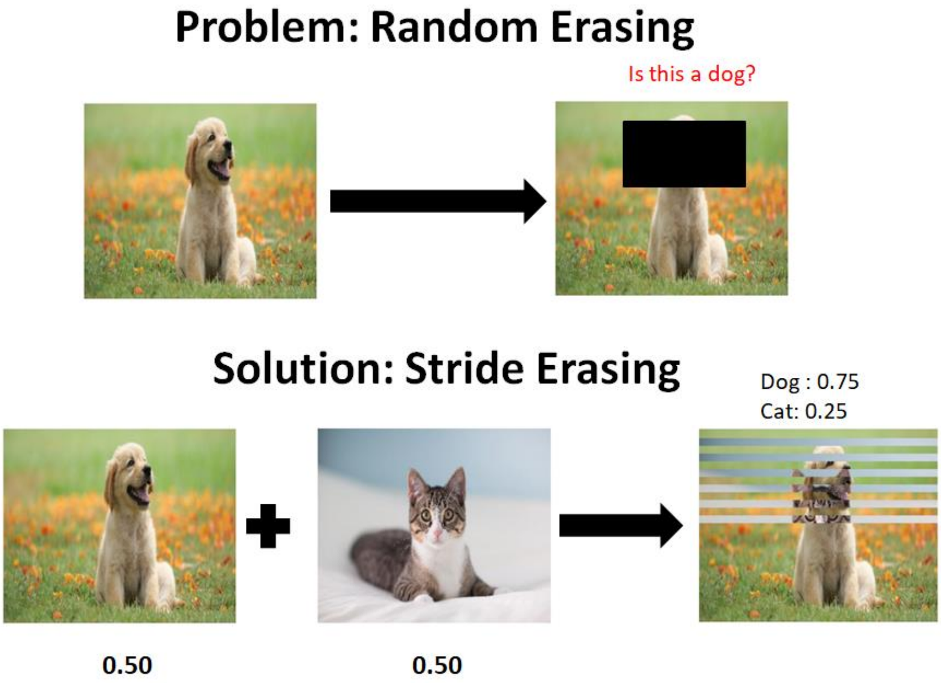


Fig. 1. The first row highlights the problem of important features removal with random erasing, the second row represents the proposed solution

3. PROPOSED METHOD

In this section, we explain our proposed approach stride random erasing data augmentation (SREA) method. During training, there is a probability P of performing SREA. In SREA, W and $P_s/2$ represent the width of image and the striding probability, respectively. There are n strides, calculated by $\lfloor W \times P_s/2 \rfloor$ and with random stride size S, of image X_1 and X_2 are pasted alternatively to generate a new augmented image X_a. As for images X_1 and X_2, the stride probability is $P_s/2$ and $1 - P_s/2$, respectively, so, with the same probability, L_1 and L_2 are labels of image X_1 and X_2, respectively, are mixed to generate an augmented label L_a. The newly augmented image X_a and augmented label L_a are used for training the model. The reason for halving the Ps is, in an augmented image, strides of images X_1 and X_2 are pasted alternatively i.e. one stride of X_1, then stride of X_2, process continues till n strides are done, consequently half strides of X_1 are pasted and place of half strides taken by strides of X_2, logically the probability of X_1 is also halved. For further clarification, for example, although the dog and cat have a initially mixing probability of 0.5 each before mixing, but in the augmented image, half strides of the cat are taken by strides of the dog, so the cat contributes half (0.25) of the original probability (0.5) as it is shown in Figure 1. We define the SREA mathematical combination operation as below:

$$X_a = X_2 \oplus [X_1 \otimes n * S] \quad \textit{Eq. 1}$$

In the above equation, $\oplus$ and $\otimes$ represent pasting and striding operations, respectively. $n * S$ represents n strides of size S each. In the same way, labels are also mixed as follows:

$$L_a = L_1 . P_s/2 + L_2 . (1 - P_s/2) \quad \textit{Eq. 2}$$

The labels are mixed in the same ratio as images are mixed, consequently this provides a strong regularization effect and makes the model more generalized.

The proposed algorithm for this approach is defined in algorithm 1. The source code is available in a git repository.

Algorithm 1: Stride Augmentation(X_1,X_2,L_1, L_2)

```
Input: $X_1$ : Image 1
       $X_2$ : Image 2
       $L_1$: label of $X_1$,
       $L_2$: label of $X_2$,
Output: Augmented image, Augmented label
Probability = random() // random probability of mixing x2
strideSize= random(2,10) // random stride size in range of 2 to 10
width = width($X_1$) // getting width of image
totalStrides= int(width/strideSize) // Total number of strides
mixingStride = int(totalStrides*Probability) // number of strides use for
    mixing images
startPosition = random(1,int(width/2)) // startion position for mixing
    images
$X_a$ = copy($X_1$) // Augmented image to be stored
for Choose $i \in range(0, mixingStrides, 2)$ do
    $X_a$[startPosition+i*strideSize:startPosition+(i+1)*strideSize,:] =
        $X_2$[startPosition+i*strideSize:startPosition+(i+1)*strideSize,:]
        // Mixing images with strides
Probability = Probability/2.0 // In loop, after each second interval
    stride of x2 is mixed, so probability is halved
$L_a$=$L_1$*(1-Probability)+$L_2$*Probability
return $X_a$, $L_a$
```

4. Experiment

In this section, we define the datasets used, the training set up and the classification results obtained for this initial evaluation of our SREA method, the random erasing method and a baseline with no data augmentation.

4.1. Datasets

We used four datasets for our experiments including Fashion-MNIST [24], CIFAR10 [25], CIFAR100 [25] and STL10 [26].

Fashion-MNIST

It consists of 70000 images including 60000 training and 10000 test images. Each image is gray scale and of size 28 × 28. There are 10 classes of clothing items e.g. t-shirt, shoe and dress. Before training, we normalized these images between 0 and 1.

CIFAR10 and CIFAR100

It consists of 60000 images, including 50000 training and 10000 test images. Each image is RGB color and of dimension 32 ×32 × 3. There are 10 classes in this dataset. These data images were normalized using the mean and standard deviation of the dataset. Similar to CIFAR10, CIFAR100 has the same number of images, same dimensions and everything except the number of classes are 100.

STL10

This dataset has a total of 8500 images including 500 training images and 8000 test images. Each image is RGB color and of dimension 96 x 96 x 3. There are 10 classes in this dataset. These images are acquired from the biggest imagenet dataset.

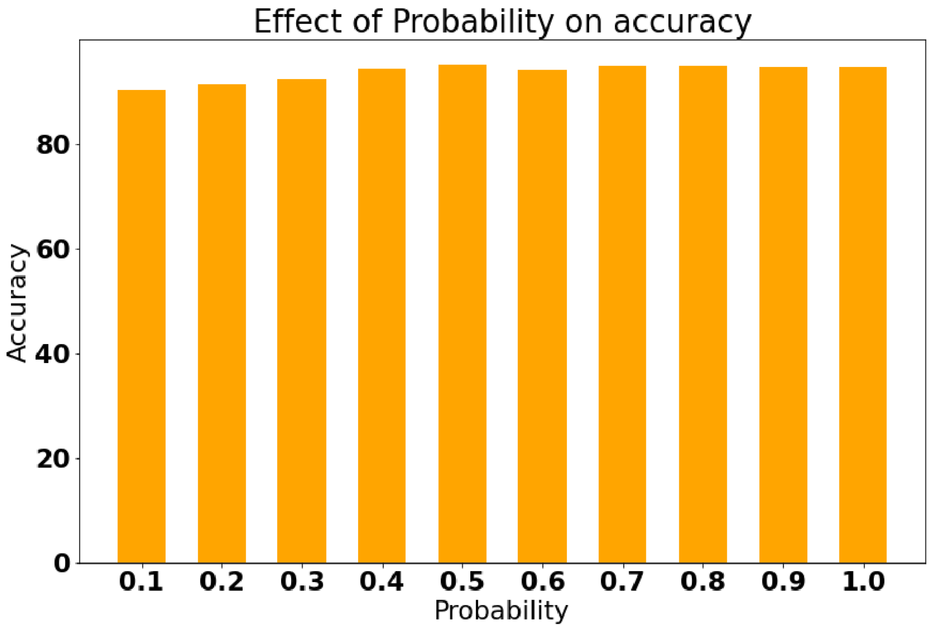


Fig. 2. On Fashion-MNIST dataset using Resnet20 network

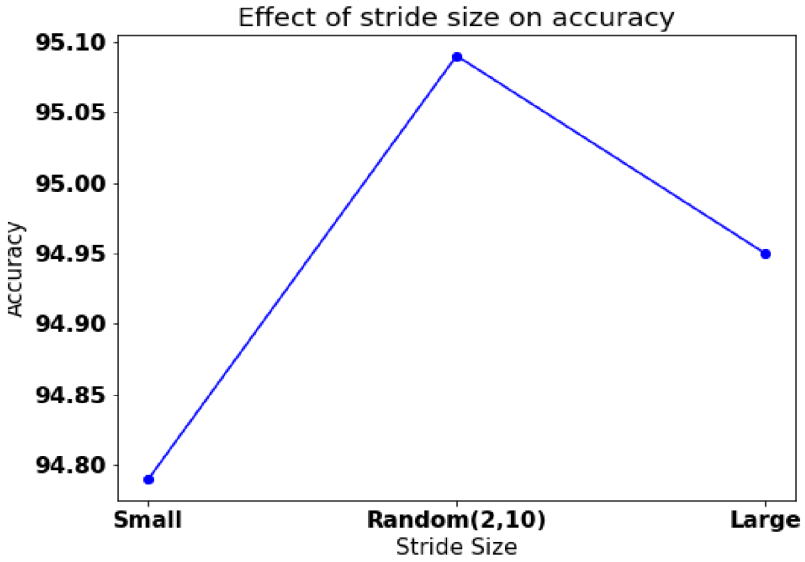


Fig. 3. On Fashion-MNIST dataset using Resnet20 network

4.2. Training setup

For training setup, we use multiple flavours of resnet [27] : resnet20, resnet32, resnet44, resnet56 , resnet100 and flavours of VGG [28] model i.e. VGG11, VGG13, VGG16 and VGG19. For the fair comparison with random erasing, the overall parametric settings are employed with the same setting as in [14]. We used 300 epochs for training, the learning rate was initially set to 0.1 and reduced by 10 times at epoch 100, 150, 175 and 190. The probability of performing SREA is set to 0.5 for the main experiments. This is because we initially investigated 10 different SREA probability settings with an interval of 0.1 starting from 0.1 on FashionMNIST using resnet20 model. In this test 0.5, SREA probability showed the best result, as shown in Figure 2. We re-performed all the Zhong et al.'s experiments for fashion-MNIST, because the original experiments performed in the random erasing paper [29] were on an old fashioned dataset, in which there was overlapping between test and training images (this issue is discussed in the Github repository of random erasing [14]). Each experiment is repeated three times and the mean error with standard deviation is reported in Table 1. Note that, boldface number shows the best performance.

4.3. Results

In this section, the results achieved with SREA are compared with the baseline and the standard random erasing augmentation method. Firstly, we investigated the effect of stride size. For this purpose, we used a fixed small stride size of 2, a fixed large stride size of 10 and a randomly generated stride size between 2 and 10 on the Fashion-MNIST dataset using resnet20. Out of all three sizes, the randomly generated stride size has shown better performance for this dataset as shown in Figure 3. Furthermore, with classification tasks, SREA also outperformed both baseline and random erasing in all flavours of the resnet model by showing better results in all categories, albeit sometimes within the margin of error. While in the case of CIFAR10 and CIFAR100, this initial implementation of SREA has shown competitive results with random erasing. In some resnet flavour cases it narrowly outperformed random erasing (again within the margin of error) and it showed impressive performance over baseline in all resnet flavours. For further evaluating the effectiveness of SREA, we use multiple flavours of VGG, it shows superior performance as compared to baseline and competitive performance with random erasing. For STL10 data, SREA outperformed both baseline and random erasing except the VGG19 network.

Table 1. Error rate performance comparison of the proposed SREA method with a baseline and random erasing.

Models	**Baselines**	**Random Erasing**	**SREA**
Fashion-MNIST			
ResNet20	6.21± 0.11	5.04 ± 0.10	**4.91 ± 0.12**
Resnet32	6.04 ± 0.13	4.84 ± 0.12	**4.81 ± 0.17**
Resnet44	6.08 ± 0.16	4.87 ± 0.1	**4.07 ± 0.14**
Resnet56	6.78 ± 0.16	5.02 ± 0.11	**5.00 ± 0.19**
CIFAR10			
Resnet20	7.21 ± 0.17	**6.73 ± 0.09**	7.18 ± 0.13
Resnet32	6.41 ± 0.06	**5.66 ± 0.10**	6.31 ± 0.14
Resnet44	5.53 ± 0.0	5.13 ± 0.09	**5.09 ± 0.10**
Resnet56	5.31 ± 0.07	**4.89 ± 0.0**	5.02 ± 0.11
VGG11	7.88±0.76	7.82±0.65	**7.80±0.65**
VGG13	6.33±0.23	6.22±0.63	**6.18±0.54**
VGG16	6.42±0.34	6.21±0.76	**6.20±0.34**
VGG19	6.88±0.65	6.85±0.65	**6.75±0.55**
CIFAR100			
Resnet20	30.84 ± 0.19	**29.97 ± 0.11**	30.18 ± 0.27
Resnet32	28.50 ± 0.37	27.18 ± 0.32	**27.08 ± 0.34**
Resnet44	25.27 ± 0.21	**24.29 ± 0.16**	24.49 ± 0.23
Resnet56	24.82 ± 0.27	23.69 ± 0.33	**23.35 ± 0.26**
VGG11	28.97±0.76	28.73±0.67	**28.26±0.75**
VGG13	25.73±0.67	**25.71±0.54**	25.71±0.56
VGG16	26.64±0.56	26.63±0.75	**26.61±0.65**
VGG19	**28.65±0.23**	28.69±0.76	28.75±0.76
STL10			
VGG11	22.29±0.13	22.27±0.21	**20.68±0.23**
VGG13	20.64±0.26	20.18±0.23	**19.91±0.92**
VGG16	20.62±0.34	20.12±0.65	**20.09±0.23**
VGG19	**19.15±0.32**	19.22±0.45	19.35±0.11

5. CONCLUSION

This paper addressed the issues of random erasing, where good features are lost due to randomly erasing a random size of patch, which deteriorates the model performance. To cope up with this issue, we proposed a new data augmentation method named Stride Random Erasing data augmentation, that not only provides random erasing but also preserves significant features. We investigated the effect of different probability values and stride sizes parameters on our approach. Furthermore, our approach outperformed baseline and random erasing on a wide variety of datasets using different flavour of resnet and vgg. In future, we will extend our work by including column-wise strides, both row-wise and column-wise strides and test SREA on audio datasets. Nonetheless this first implementation of the approach shows promise for building a new family of stride-based data augmentation techniques.

ACKNOWLEDGEMENTS

This publication has emanated from research [conducted with the financial support of/supported in part by a grant from] Science Foundation Ireland under Grant number 18/CRT/6223 and is supported by the ADAPT Centre for Digital Content Technology which is funded under the SFI Research Centres Programme (Grant 13/RC/2106/_P2), Lero SFI Centre for Software (Grant 13/RC/2094/_P2) and is co-funded under the European Regional Development Fund. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

REFERENCES

[1] Kumar, J., Bedi, P., Goyal, S. B., Shrivastava, A., & Kumar, S. (2021, March). Novel Algorithm for Image Classification Using Cross Deep Learning Technique. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1099, No. 1, p. 012033). IOP Publishing.

[2] Liu, J. E., & An, F. P. (2020). Image classification algorithm based on deep learning-kernel function. *Scientific programming, 2020.*

[3] Wang, H., & Meng, F. (2019). Research on power equipment recognition method based on image processing. *EURASIP Journal on Image and Video Processing*, *2019*(1), 1-11.

[4] Nanni, L., Maguolo, G., Brahnam, S., & Paci, M. (2021). An ensemble of convolutional neural networks for audio classification. *Applied Sciences*, *11*(13), 5796.

[5] Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., ... &Wilson, K. (2017, March). CNN architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 131-135). IEEE.

[6] Rong, F. Audio classification method based on machine learning. 2016 International Conference On Intelligent Transportation, Big Data & Smart City (ICITBS) pp.81-84 (2016)

[7] Aiman, A., Shen, Y., Bendechache, M., Inayat, I. & Kumar, T. AUDD: Audio Urdu Digits Dataset for Automatic Audio Urdu Digit Recognition. Applied Sciences. 11, 8842 (2021)

[8] Kolluri, J., Razia, D. S., & Nayak, S. R. (2019, June). Text classification using Machine Learning and Deep Learning Models. In *International Conference on Artificial Intelligence in Manufacturing & Renewable Energy (ICAIMRE)*.

[9] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep Learning--based Text Classification: A Comprehensive Review. *ACM Computing Surveys (CSUR)*, *54*(3), 1-40.

[10] Nguyen, T. H., & Shirai, K. (2013, June). Text classification of technical papers based on text segmentation. In *International Conference on Application of Natural Language to Information Systems* (pp. 278-284). Springer, Berlin, Heidelberg.

[11] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456). PMLR.

[12] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929-1958.

[13] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*, 1097-1105.

[14] Zhong, Z., Zheng, ., Kang, G., Li, S., & Yang, Y. (2020, April). Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 07, pp. 13001-13008).

[15] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[16] Takahashi, R., Matsubara, T., & Uehara, K. (2019). Data augmentation using random image cropping and patching for deep CNNs. *IEEE Transactions on Circuits and Systems for Video Technology*, *30*(9), 2917-2931.

[17] Mikołajczyk, A., & Grochowski, M. (2018, May). Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)* (pp. 117-122). IEEE.

[18] Chen, S., Dobriban, E., & Lee, J. H. (2020). A group-theoretic framework for data augmentation. *Journal of Machine Learning Research*, *21*(245), 1-71.

[19] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(1), 1-48

[20] Wei, J., & Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

[21] Ba, J., & Frey, B. (2013). Adaptive dropout for training deep neural networks. *Advances in neural information processing systems*, *26*, 3084-3092.

[22] Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., & Fergus, R. (2013, May). Regularization of neural networks using dropconnect. In *International conference on machine learning* (pp. 1058-1066). PMLR.

[23] Zeiler, M. D., & Fergus, R. (2013). Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*.

[24] Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

[25] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.

[26] A. Coates, A. Ng, and H. Lee, “An analysis of single-layer networks in unsupervised feature learning,” in Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2011, pp. 215–223.

[27] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[28] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[29] https://github.com/zhunzhong07/Random-Erasing/issues/9

AUTHORS

Teerath kumar received his Bachelor’s degree in Computer Science with distinction from National University of Computer and Emerging Science (NUCES), Islamabad, Pakistan, in 2018. Currently, he is pursuing PhD from Dublin City University, Ireland. His research interests include advanced data augmentation, deep learning for medical imaging, generative adversarial networks and semi-supervised learning.

R. Brennan is an Assistant Professor in the School of Computing, Dublin City University, Chair of the DCU MA in Data Protection and Privacy Law and a Funded investigator in the Science Foundation Ireland ADAPT Centre for Digital Content Technology which is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund, His main research interests are data protection, data value, data quality, data privacy, data/AI governance and semantics.

M. Bendechache is an Assistant Professor in the School of Computing at Dublin City University, Ireland. She obtained her Ph.D. degree from University College Dublin, Ireland in 2018. Malika’s research interests span the areas of Big data Analytics, Machine Learning, Data Governance, Cloud Computing, Blockchain, Security, and Privacy. She is an academic member and a Funded Investigator of ADAPT and Lero research centres.

Fixed-Point Code Synthesis for Neural Networks*

Hanane Benmaghnia[1], Matthieu Martel[1,2], and Yassamine Seladji[3]

[1] University of Perpignan Via Domitia, Perpignan, France
[2] Numalis, Cap Omega, Rond-point Benjamin Franklin 34960 Montpellier, France
[3] University of Tlemcen Aboubekr Belkaid, Tlemcen, Algeria
[1]{first.last}@univ-perp.fr,[3]yassamine.seladji@univ-tlemcen.dz

Abstract. Over the last few years, neural networks have started penetrating safety critical systems to take decisions in robots, rockets, autonomous driving car, etc. A problem is that these critical systems often have limited computing resources. Often, they use the fixed-point arithmetic for its many advantages (rapidity, compatibility with small memory devices.) In this article, a new technique is introduced to tune the formats (precision) of already trained neural networks using fixed-point arithmetic, which can be implemented using integer operations only. The new optimized neural network computes the output with fixed-point numbers without modifying the accuracy up to a threshold fixed by the user. A fixed-point code is synthesized for the new optimized neural network ensuring the respect of the threshold for any input vector belonging the range $[x_{min}, x_{max}]$ determined during the analysis. From a technical point of view, we do a preliminary analysis of our floating neural network to determine the worst cases, then we generate a system of linear constraints among integer variables that we can solve by linear programming. The solution of this system is the new fixed-point format of each neuron. The experimental results obtained show the efficiency of our method which can ensure that the new fixed-point neural network has the same behavior as the initial floating-point neural network.

Keywords: Computer Arithmetic, Code Synthesis, Formal Methods, Linear Programming, Numerical Accuracy, Static Analysis.

1 Introduction

Nowadays, neural networks have become increasingly popular. They have started penetrating safety critical domains and embedded systems, in which they are often taking important decisions such as autonomous driving cars, rockets, robots, etc. These neural networks become larger and larger while embedded systems still have limited resources (memory, CPU, etc.) As a consequence, using and running deep neural networks [26] on embedded systems with limited resources introduces several new challenges [7,9,11,12,15,16,18,19]. The fixed-point arithmetic is more adapted for these embedded systems which often have a working processor with integers only. The approach developed in this article concerns the fixed-point and integer arithmetic applied to trained neural networks (NNs). NNs are trained on computers with a powerful computing unit using most of the time the IEEE754 floating-point arithmetic [13,21]. Exporting NNs using fixed-point arithmetic can perturb or change the answer of the NNs which are in general sensible to the computer arithmetic. A new approach is required to adapt NN computations to the simpler CPUs of embedded systems. This method consists in using fixed-point arithmetic because it is faster and lighter to manipulate for a CPU while it is more complicated to handle for the developer. We consider the problem of tuning the formats (precision) of an already trained floating-point NN, in such a way that, after tuning, the synthesized fixed-point NN behaves almost like the original performing computations. More precisely, if the NN is an interpolator, i.e. NNs computing mathematical functions, the original NN (floating-point) and the new NN (fixed-point) must behave identically if they calculate a given function f, such that, the

* This work is supported by La Region Occitanie under Grant GRAINE - SYFI. https://www.laregion.fr

David C. Wyld et al. (Eds): AISCA, NET, DNLP - 2022
pp. 11-30, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120202

absolute error (Equation (1)) between the numerical results computed by both of them is equal to or less than a threshold set by the user. If the NN is a classifier, the new NN have to classify correctly the outputs in the right category comparing to the original NN. This method is developed in order to synthesize NNs fixed-point codes using integers only. This article contains nine sections and an introductory example in Section 3, where we present our method in a simplified and intuitive way. Some notations are introduced in Section 2. In Section 4, we present the fixed-point arithmetic, where we show how to represent a fixed-point number and the elementary operations. The errors of computations and conversions inside a NN are introduced in Section 5. Section 6 deals with the generation of constraints to compute the optimal format for each neuron using linear programming [30]. Our tool and its features are presented in Section 7. Finally, we demonstrate the experimental results in Section 8 in terms of accuracy and bits saved. Section 9 presents the related work then Section 10 concludes and gives an overview of our future work.

2 Notations

In the following sections, we will use these notations:
●$\aleph$: Set of fixed-point numbers. ●$\mathbb{N}$: Set of natural integers. ●$\mathbb{Z}$: Set of relative integers. ●$\mathbb{R}$: Set of real numbers. ●$\mathbb{F}$: Set of IEEE754 floating-point numbers [13]. ●**NN**: Neural Network. ● $<\boldsymbol{M^{\hat{x}}}, \boldsymbol{L^{\hat{x}}}>$: Format of the fixed-point number $\hat{x}$ where $\boldsymbol{M^{\hat{x}}}$ represents the Most significant bit (integer part) and $\boldsymbol{L^{\hat{x}}}$ the Least significant bit (fractional part). ●$\boldsymbol{\epsilon_{\hat{x}}}$: Error on the fixed-point number $\hat{x}$. ●$\boldsymbol{b}$: Bias. ●$\boldsymbol{W}$: Matrix of weights. ●$\boldsymbol{m}$: Number of layers of a neural network. ●$\boldsymbol{n}$: Number of neurons by layer. ●$\boldsymbol{k}$: Index of layer. ●$\boldsymbol{i}$: Index of neuron. ●**ufp**: Unit in the first place [13,21]. ● **ReLU**: Rectified Linear Unit [10, 24,29]. ●$\boldsymbol{T}$: size of data types (8, 16, 32 bits.) ●$\oplus$: Fixed-point addition. ●$\otimes$: Fixed-point multiplication.

3 An Introductory Example

In this section, we present a short example of a fully connected neural network [3] containing three layers ($m = 3$) and two neurons by layer ($n = 2$) as shown in Figure 1. The objective is to give an intuition of our approach.

Our main goal is to synthesize a fixed-point code for an input NN with an error threshold between 0 and 1 defined by the user, and respecting the initial NN which uses the floating-point arithmetic [13, 21]. The error threshold is the maximal absolute error accepted between the original floating-point NN and the synthesized fixed-point NN in all the outputs of the output layer (max norm). This absolute error is computed by substracting the fixed-point value to the floating-point value (IEEE754 with single precision [13,21]) as defined in Equation (1). To compute this error, we convert the fixed-point value into a floating-point value.

$$Absolute\,error = |FloatingPoint\,Result - FixedPoint\,Result| \tag{1}$$

In this example, the threshold is 0.02 and the data type $T = 32$ bits. In other words, the resulting error of all neurons in the output layer (u_{30}, u_{31} of Figure 1) must be equal to or less than 0.02 using integers in 32 bits.

Hereafter, we consider the feature layer X_0, which corresponds to the input vector. The biases are b_0, b_1 and b_2. The matrices of weights are W_0, W_1 and W_2, such that each bias (respectively matrix of weights) corresponds to one layer.

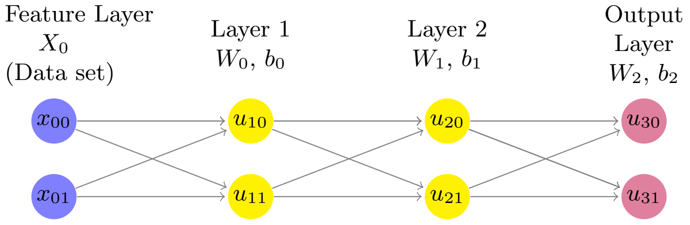


Fig. 1. A fully connected NN with 3 layers and 2 neurons per layer.

The affine function for the k^{th} layer is defined in a standard way as

$$\begin{aligned} f_{k,i}: \quad & \mathbb{F}^n \longrightarrow \mathbb{F} \\ & X_{k-1} \longmapsto u_{k,i} = f_{k,i}(X_{k-1}) = \sum_{j=0}^{n-1}(w_{k-1,i,j} \times x_{k-1,j}) + b_{k-1,i}, \end{aligned} \tag{2}$$

$\forall 1 \leq k \leq m$, $\forall 0 \leq i < n$, where X_{k-1}= $(x_{k-1,0}, ..., x_{k-1,n-1})^t$ is the input vector (the input of the layer k is the output of the layer $k-1$), $b_{k-1} = (b_{k-1,0}, ..., b_{k-1,n-1})^t \in \mathbb{F}^n$ and $W_{k-1} \in \mathbb{F}^{n\times n}$ ($w_{k-1,i,j}$ is the coefficient in the line i and column j in W_{k-1}.)
Informally, a fixed-point number is represented by an $integer\,value$ and a $format$ $< M, L >$ which gives us the information about the number M of significant bits before the binary point and L the number of significant bits after the binary point required for each coefficient in W_{k-1}, b_{k-1}, X_0, and the output of each neuron $u_{k,i}$. We notice that, at the beginning, we convert the input vector X_0 in the size of the data type required T. In fixed-point arithmetic, before computing the affine function defined in Equation (2), we need to know the optimal formats $< M, L >$ of all the coefficients. To compute these formats, we generate automatically linear constraints according to the given threshold. These linear constraints formally defined in Section 6 are solved by linear programming [30], and they give us the optimal value of the number of significant bits after the binary point L for each neuron. We show in Equation (3) some constraints generated for the neuron u_{31} of the NN of Figure 1.

$$\begin{cases} L^u_{31} \geq 6 \\ L^u_{31} + M^u_{31} \leq 31 \\ L^u_{31} \leq L^x_{20}, \\ L^u_{31} \leq L^x_{21}, \\ \dots \end{cases} \tag{3}$$

We notice that L^x_{20} (respectively L^x_{21}) is the length of the fractional part of u_{20} (respectively u_{21}.) The first constraint gives a lower bound for L^u_{31}, so the output u_{31} in the output layer has to fullfil the threshold fixed by the user and the error done must be equal to or less than this one. In other words, the number of significant bits of each neuron in the output layer must be equal at least to 6 (if it is greater than 6, this means that we are more accurate.) The value 6 is obtained by computing the unit in the first place [13, 21] of the threshold defined as

$$\forall x \in \mathbb{F}, \quad \text{ufp}(x) = \min\{i \in \mathbb{N} : 2^{i+1} > x\} = \lfloor \log_2(x) \rfloor \tag{4}$$

The second constraint avoids overflow and ensures compliance to the data type chosen by the user (integers on 8, 16 or 32 bits.) The third and fourth constraints ensure that the length of the fractional part L^u_{31} computed is less than or equal to the length of the fractional parts of all its inputs (L^x_{20} and L^x_{21}.) Using the formats resulting from the solver, firstly, we convert all the coefficients of weights matrices W_{k-1}, biases b_{k-1} and inputs X_0 from floating-point values to fixed-point values.

Let b_{k-1}, W_{k-1} and X_0 used in this example be

$$b_0 = \begin{pmatrix} -2 \\ 4.5 \end{pmatrix}, b_1 = \begin{pmatrix} 1.2 \\ 0.5 \end{pmatrix}, b_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, W_0 = \begin{pmatrix} 3.5 & 0.25 \\ -1.06 & 4.1 \end{pmatrix}, W_1 = \begin{pmatrix} -0.75 & 4.85 \\ 2.1 & 0.48 \end{pmatrix},$$
$$W_2 = \begin{pmatrix} -5 & 12.4 \\ 0.2 & -2 \end{pmatrix}, X_0 = \begin{pmatrix} 2 \\ 0.5 \end{pmatrix}.$$

Table 1 presents the output results for each neuron. The floating-point results are shown in the second column and the fixed-point results in the third one. The last column contains the absolute error defined in Equation (1) for the output layer (u_{30}, u_{31}) only.

Table 1. Comparison between the floating-point and the fixed-point results corresponding to the NN of Figure 1.

Neuron	Floating-Point Result	Fixed-Point Result	Absolute Error
$\boldsymbol{u_{10}}$	5.125	2624<3,9>= 5.125	/
$\boldsymbol{u_{11}}$	4.43	4535<2,10>= 4.4287	/
$\boldsymbol{u_{20}}$	18.8417	9643<5,9>= 18.8339	/
$\boldsymbol{u_{21}}$	13.3889	6854<5,9>= 13.3867	/
$\boldsymbol{u_{30}}$	74.8136	76620<9,10>= 74.8247	$1.06 \times 10^{-2} \boldsymbol{\approx 2^{-7}}$
$\boldsymbol{u_{31}}$	-22.0094	-22536 <7,10>= -22.0078	$1.63 \times 10^{-3} \boldsymbol{\approx 2^{-10}}$

The error threshold fixed by the user at the beginning was 0.02 (6 significant bits after the binary point.) As we can see, the absolute error of the output layer in the Table 1 is under the value of the threshold required. This threshold is fulfilled with our method using fixed-point arithmetic. Now, we can synthesize a fixed-point code for this NN respecting the user's threshold, the data type T, and ensuring the same behavior and quality as the initial floating-point NN.

Figure 2 shows some lines of code synthesized by our tool for the neurons u_{10} and u_{11} using Equation (2) and the fixed-point arithmetic. The running code gives the results shown in the third column of the Table 1. For example, the line 5 represents the input $x_{00} = 2$ in the fixed-point representation. This value is shifted on the right through 6 bits (line 7) in order to be aligned and used in the multiplication (line 8) by $w_{000} = 3.5$ represented by 112 in the fixed-point arithmetic. The fixed-point output u_{10} (2624) in the Table 1 is returned by the line 16.

4 Fixed-Point Arithmetic

In this section, we briefly describe the fixed-point arithmetic as implemented in most digital computers [4,5,32]. Since fixed-point operations rely on integer operations, computing with fixed-point numbers is highly efficient. We start by defining the representation of a fixed-point number in Subsection 4.1, then we present briefly the operations needed (addition, multiplication and activation functions) in this article in Subsection 4.2.

4.1 Representation of a Fixed-Point Number

A fixed-point number is represented by an **integer** value and a **format** $< M, L >$ where $M \in \mathbb{Z}$ is the number of significant bits before the binary point and $L \in \mathbb{N}$ is the number of significant bits after the binary point. We write the fixed-point number $\hat{a} = value_{<M^{\hat{a}}, L^{\hat{a}}>}$ and define it in Definition 1.

Definition 1. *Let us consider* $\hat{a} \in \aleph$, $A_{\hat{a}} \in \mathbb{N}$ *such that* $\hat{a} = (-1)^{s_{\hat{a}}}.A_{\hat{a}}.\beta^{-L_{\hat{a}}}$ *and* $P_{\hat{a}} = M_{\hat{a}} + L_{\hat{a}} + 1$, *where* β *is the basis of representation,*

```
 int main()
{ /* That NN has 3 layers and 2 neurons per layer */
  int mul, u[3][2], x[4][2];

 x[0][0]=1073741824;          // <1,29>
 x[0][1]=1073741824;          // <-1,31>
 x[0][0]=x[0][0]>>6;          // <1,23>
 mul=112*x[0][0];             // <3,9>=<1,5>*<1,23>
 mul=mul>>19;
 u[1][0]=mul;                 // <3,9>
 x[0][1]=x[0][1]>>7;          // <-1,24>
 mul=16*x[0][1];              // <-2,14>=<-2,9>*<-1,24>
 mul=mul>>19;
 mul=mul>>5;
 u[1][0]=u[1][0]+mul;         //<3,9>=<3,9>+<-2,9>
 u[1][0]=u[1][0]+-1024;       //<3,9>=<3,9>+<1,9>
 u[1][0]=max(0,u[1][0]);      // ReLU(u[1][0])
 x[1][0]=u[1][0];
 x[0][0]=x[0][0]>>3;          // <1,20>
 mul=-543*x[0][0];            // <2,10>=<0,9>*<1,20>
 mul=mul>>19;
 u[1][1]=mul;                 // <2,10>
 x[0][1]=x[0][1]>>5;          // <-1,19>
 mul=262*x[0][1];             // <2,10>=<2,10>*<-1,19>
 mul=mul>>19;
 u[1][1]=u[1][1]+mul;         //<2,10>=<2,10>+<2,10>
 u[1][1]=u[1][1]+4608;        //<2,10>=<2,10>+<2,10>
 u[1][1]=max(0,u[1][1]);      // ReLU(u[1][1])
 ...
 return 0; }

```

Fig. 2. Fixed-point code synthesized for the neurons u_{10} and u_{11} of Figure 1 on 32 bits.

$\hat{a}$ is the fixed-point number with implicit scale factor $\beta^{-L_{\hat{a}}}$ (Figure 3), $A_{\hat{a}}$ is the integer representation of $\hat{a}$ in the basis β, $P_{\hat{a}} \in \mathbb{N}$, $P_{\hat{a}} = M_{\hat{a}} + L_{\hat{a}} + 1$ is the length of $\hat{a}$ and $s_{\hat{a}} \in \{0, 1\}$ is its sign.

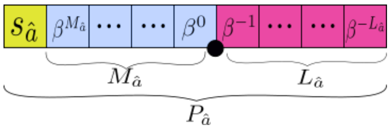


Fig. 3. Fixed-point representation of $\hat{a}$ in a format $< M^{\hat{a}}, L^{\hat{a}} >$.

The difficulty of the fixed-point representation is managing the position of the binary point manually against the floating-point representation which manages it automatically.

Example 1. : The fixed-point value $3 < 1, 1 >$ corresponds to 1.5 in the floating-point representation. We have first to write 3 in binary then we put the binary point at the right place (given by the format) and finally we convert it again into the decimal representation: $3_{10} < 1, 1 >= 11_2 < 1, 1 >= 1.1_2 = 1.5_{10}$.

4.2 Elementary Operations

This subsection defines the elementary operations needed in this article like addition and multiplication which are used later in Equation (11). We also define $Re\hat{L}U$ (respectively $Lin\hat{e}ar$) in fixed-point arithmetic which corresponds to the activation function in some NNs [10, 24, 29].

1. **Fixed-Point Addition**
   Let us consider the two fixed-point numbers $\hat{a}$, $\hat{b} \in \aleph$ and their formats $< M^{\hat{a}}, L^{\hat{a}} >$, $< M^{\hat{b}}, L^{\hat{b}} >$ respectively. Let $\oplus$ be the fixed-point addition given by $\hat{c} \in \aleph$, $\hat{c} = \hat{a} \oplus \hat{b}$.

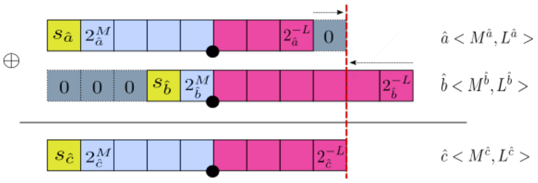


Fig. 4. Addition of two fixed-point numbers without a carry.

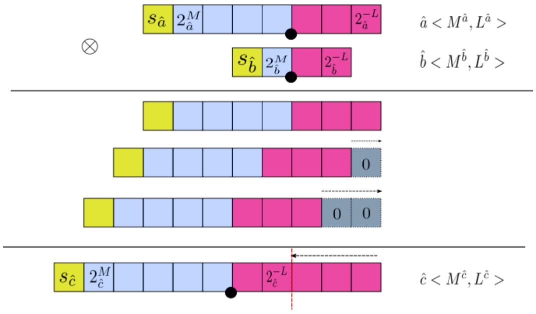


Fig. 5. Multiplication of two fixed-point numbers.

   Figure 4 shows the fixed-point addition between $\hat{a}$ and $\hat{b}$. The fixed-point format required is $< M^{\hat{c}}, L^{\hat{c}} >$. The objective is to have a result in this format, this is why we start by aligning the length of the fractional parts $L^{\hat{a}}$ and $L^{\hat{b}}$ according to $L^{\hat{c}}$. If $L^{\hat{a}} > L^{\hat{c}}$, we truncate with $L^{\hat{a}} - L^{\hat{c}}$ bits, otherwise, we add $L^{\hat{c}} - L^{\hat{a}}$ zeros on the right hand of $\hat{a}$, then we do the same for $\hat{b}$. The length of the integer part $M^{\hat{c}}$ must be the maximum value between $M^{\hat{a}}$ and $M^{\hat{b}}$. If there is a carry, we add +1 to the number of bits of the integer part, otherwise the result is wrong. The algorithm of the fixed-point addition is given in [20, 23].
2. **Fixed-Point Multiplication**
   Let us consider the two fixed-point numbers $\hat{a}$, $\hat{b} \in \aleph$ and their formats $< M^{\hat{a}}, L^{\hat{a}} >$, $< M^{\hat{b}}, L^{\hat{b}} >$ respectively. Let $\otimes$ be the fixed-point multiplication given by $\hat{c} \in \aleph$, $\hat{c} = \hat{a} \otimes \hat{b}$.
   Figure 5 shows the fixed-point multiplication between $\hat{a}$ and $\hat{b}$. The fixed-point format required is $< M^{\hat{c}}, L^{\hat{c}} >$. The objective is to have a result in this format, this is why we start by doing a standard multiplication which is composed by shifts and additions. If $(L^{\hat{a}} + L^{\hat{b}}) > L^{\hat{c}}$, we truncate with $(L^{\hat{a}} + L^{\hat{b}}) - L^{\hat{c}}$ bits, otherwise, we add $L^{\hat{c}} - (L^{\hat{a}} + L^{\hat{b}})$ zeros on the right hand of $\hat{c}$. The length of the integer part $M^{\hat{c}}$ must be the sum of $M^{\hat{a}}$ and $M^{\hat{b}}$, otherwise the result is wrong. The algorithm of the fixed-point multiplication is given in [20, 23].
3. **Fixed-Point $Re\hat{L}U$**
   Definition 2 defines the fixed-point $Re\hat{L}U$ which is a non-linear activation function computing the positive values.

   Definition 2. *Let us consider the fixed-point number $\hat{a} = V_{\hat{a}} < M^{\hat{a}}, L^{\hat{a}} > \in \aleph$ and the fixed-point zero written $\hat{0} = 0 < 0, 0 > \in \aleph$. Let $\hat{c} = V_{\hat{c}} < M^{\hat{c}}, L^{\hat{c}} > \in \aleph$ be the result of the fixed-point $Re\hat{L}U$ given by*

$$\hat{c} = Re\hat{L}U(\hat{a}) = m\hat{a}x(\hat{0}, \hat{a}), \tag{5}$$

$$where\ V_{\hat{c}} = max(0,\, V_{\hat{a}})\ and < \ M^{\hat{c}},\ L^{\hat{c}} >= \begin{cases} < M^{\hat{a}}, L^{\hat{a}} > & if \quad V_{\hat{c}} = V_{\hat{a}}, \\ < 0,\, 0 > & otherwise. \end{cases}$$

4. **Fixed-Point $\hat{Linear}$**
   Definition 3 defines the fixed-point $\hat{Linear}$ which is an activation function returning the identity value.

   Definition 3. *Let us consider the fixed-point number $\hat{a} \in \aleph$ with the format $< M^{\hat{a}},\ L^{\hat{a}} >$. Let $\hat{c} \in \aleph$ be the result of the fixed-point $\hat{Linear}$ activation function given by*

$$\hat{c} = \hat{Linear}(\hat{a}) = \hat{a}. \tag{6}$$

5 Error Modelling

In this section, we introduce some theoretical results concerning the fixed-point arithmetic errors in Subsection 5.1 and we show the numerical errors done inside a NN in Subsection 5.2. The error on the output of the fixed-point affine transformation function can be decomposed into two parts: the propagation of the input error and the computational error. Hereafter, $\hat{x} \in \aleph$ is used for the fixed-point representation with the format $< M^{\hat{x}},\ L^{\hat{x}} >$ and $x \in \mathbb{F}$ for the floating-point representation. $\bar{X} \in \mathbb{F}^n$ is a vector of n floating-point numbers and $\hat{X} \in \aleph^n$ a vector of n fixed-point numbers.

5.1 Fixed-Point Arithmetic Error

This subsection defines two important properties about errors made in fixed-point addition and multiplication which are used to compute affine transformations in a NN (substraction and division are useless in our context.) We start by introducing the propositions and then the proofs. Proposition 1 defines the error of the fixed-point addition when we add two fixed-point numbers and Proposition 2 defines the error due to the multiplication of two fixed-point numbers.

Proposition 1. *Let $\hat{x}$, $\hat{y}$, $\hat{z} \in \aleph$ with a format $< M^{\hat{x}}, L^{\hat{x}} >$ (respectively $< M^{\hat{y}}, L^{\hat{y}} >$, $< M^{\hat{z}}, L^{\hat{z}} >$.) Let x, y, $z \in \mathbb{F}$ be the floating-point representation of $\hat{x}$, $\hat{y}$, $\hat{z}$. Let $\epsilon_{\oplus} \in \mathbb{R}$ be the error between the fixed-point addition $\hat{z} = \hat{x} \oplus \hat{y}$ and the floating-point addition $z = x + y$. We have that*

$$\epsilon_{\oplus} \leq 2^{-L^{\hat{x}}} + 2^{-L^{\hat{y}}} + 2^{-L^{\hat{z}}}. \tag{7}$$

Proof. Let us consider $\epsilon_{\hat{x}}$, $\epsilon_{\hat{y}}$, $\epsilon_{\hat{z}} \in \mathbb{R}$ errors of truncation in the fixed-point representation of $\hat{x}$, $\hat{y}$ and $\hat{z}$ respectively. These ones are bounded by $2^{-L^{\hat{x}}}$ ($2^{-L^{\hat{y}}}$, $2^{-L^{\hat{z}}}$ respectively) because $L^{\hat{x}}$ ($L^{\hat{y}}$, $L^{\hat{z}}$ respectively) is the last correct bit in the fixed-point representation of $\hat{x}$ (respectively $\hat{y}$, $\hat{z}$.)
We have that $z = x+y, \hat{z} = \hat{x} \oplus \hat{y}$ and $\epsilon_{\oplus} \leq \epsilon_{\hat{x}} + \epsilon_{\hat{y}} + \epsilon_{\hat{z}}$. Then we obtain $\epsilon_{\hat{z}} \leq 2^{-L^{\hat{x}}} + 2^{-L^{\hat{y}}} + 2^{-L^{\hat{z}}}$. ■

Proposition 2. *Let $\hat{x}$, $\hat{y}$, $\hat{z} \in \aleph$ with a format $< M^{\hat{x}}, L^{\hat{x}} >$ (respectively $< M^{\hat{y}}, L^{\hat{y}} >, < M^{\hat{z}}, L^{\hat{z}} >$.) Let x, y, $z \in \mathbb{F}$ be the floating-point representation of $\hat{x}$, $\hat{y}$, $\hat{z}$ in such a way $x = \hat{x} + \epsilon_{\hat{x}}$ (respectively $y = \hat{y} + \epsilon_{\hat{y}}$, $z = \hat{z} + \epsilon_{\hat{z}}$.)*
Let $\epsilon_{\otimes} \in \mathbb{R}$ be the resulting error between the fixed-point multiplication $\hat{z} = \hat{x} \otimes \hat{y}$ and the floating-point multiplication $z = x \times y$. We have that

$$\epsilon_{\otimes} \leq \hat{y} \times 2^{-L^{\hat{x}}} + \hat{x} \times 2^{-L^{\hat{y}}} + 2^{-L^{\hat{z}}}. \tag{8}$$

Proof. Let us consider $\epsilon_{\hat{x}}$, $\epsilon_{\hat{y}}$, $\epsilon_{\hat{z}} \in \mathbb{R}$ errors of truncation of $\hat{x}$, $\hat{y}$ and $\hat{z}$ respectively. These ones are bounded by $2^{-L^{\hat{x}}}$ ($2^{-L^{\hat{y}}}$, $2^{-L^{\hat{z}}}$ respectively) because $L^{\hat{x}}$ ($L^{\hat{y}}$, $L^{\hat{z}}$ respectively) is the last correct bit in the fixed-point representation of $\hat{x}$ (respectively $\hat{y}$, $\hat{z}$.)
We have that $z = x \times y$, $\hat{z} = \hat{x} \otimes \hat{y}$. We compute $(\hat{x} + \epsilon_{\hat{x}}) \times (\hat{y} + \epsilon_{\hat{y}})$ and then we obtain

$\epsilon_{\otimes} \leq \hat{y} \times \epsilon_{\hat{x}} + \hat{x} \times \epsilon_{\hat{y}} + \epsilon_{\hat{x}} \times \epsilon_{\hat{y}} + \epsilon_{\hat{z}}$. We get rid of the second order error $\epsilon_{\hat{x}} \times \epsilon_{\hat{y}}$ which is negligible in practice because our method needs to know only the most significant bit of the error which will be used in Equation (28) in Section 6. Now, the error becomes

$$\epsilon_{\otimes} \leq \hat{y} \times \epsilon_{\hat{x}} + \hat{x} \times \epsilon_{\hat{y}} + \epsilon_{\hat{z}}. \tag{9}$$

Finally, we obtain $\epsilon_{\otimes} \leq \hat{y} \times 2^{-L^{\hat{x}}} + \hat{x} \times 2^{-L^{\hat{y}}} + 2^{-L^{\hat{z}}}$. ■

5.2 Neural Network Error

Theoretical results about numerical errors inside a fully connected NN using fixed-point arithmetic are shown in this subsection. There are two types of errors: round off errors due to the computation of the affine function in Equation (11) and the propagation of the error of the input vector.

In a NN with fully connected layers [3],$\forall\, \bar{b} \in \mathbb{F}^n$, $\forall\, W \in \mathbb{F}^{n\times m}$, an output vector $\bar{u} \in \mathbb{F}^n$ is defined as

$$\begin{aligned} f: \quad & \mathbb{F}^m \longrightarrow \mathbb{F}^n \\ & \bar{X} \longmapsto \bar{u} = f(\bar{X}) = W.\bar{X} + \bar{b} \end{aligned} \tag{10}$$

Proposition 3 shows how to bound the numerical errors of Equation (10) using fixed-point arithmetic.

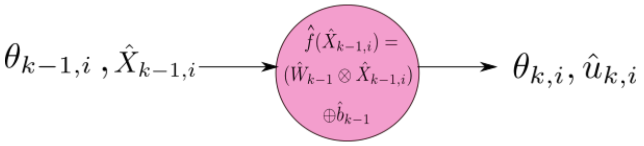

$\theta_{k-1,i}$, $\hat{X}_{k-1,i}$
$\hat{f}(\hat{X}_{k-1,i}) =$
$(\hat{W}_{k-1} \otimes \hat{X}_{k-1,i})$
$\oplus \hat{b}_{k-1}$
$\theta_{k,i}$, $\hat{u}_{k,i}$

Fig. 6. Representation of $\hat{u}_{k,i}$ the i^{th} neuron of the k^{th} layer.

Proposition 3. *Let us consider the affine transformation as defined in Equation (11). It represents the fixed-point version of Equation* (10)*. This transformation corresponds to what is computed inside a neuron (see Figure 6.) Let $\hat{u}_{k,i} \in \aleph$ be the fixed-point representation of $u_{k,i}$ and $\theta_{k,i} \in \mathbb{R}$ the error due to computations and conversions of the floating-point coefficients to fixed-point coefficients such that*

$$\begin{aligned} \hat{f}: \quad & \aleph^n \longrightarrow \aleph \\ & \hat{X}_{k-1} \longmapsto \hat{u}_{k,i} = \hat{f}(\hat{X}_{k-1}). \end{aligned} \tag{11}$$

where $\hat{f}(\hat{X}_{k-1}) = \sum_{j=0}^{n-1} (\hat{w}_{k-1,i,j} \otimes \hat{x}_{k-1,j}) \oplus \hat{b}_{k-1,i}$ *and* $\hat{X}_{k-1} = (\hat{x}_{k-1,0}, ..., \hat{x}_{k-1,n-1})^t$, $1 \leq k \leq m$, $0 \leq i < n$. *Then the resulting error $\theta_{k,i}$ for each neuron $\hat{u}_{k,i}$ of each layer is given by*

$$1 \leq k < m+1, \quad 0 \leq i < n, \quad \theta_{k,i} \leq \sum_{j=0}^{n-1} \left(2^{M^{\hat{x}}_{k-1,j} - L^{\hat{w}}_{k-1,i,j}} + 2^{M^{\hat{w}}_{k-1,i,j} - L^{\hat{x}}_{k-1,j}}\right) + n \times 2^{-L^{\hat{u}}_{k,i}} + 2^{-L^{\hat{u}}_{k,i}+1}. \tag{12}$$

Proof. The objective is to bound the resulting error for each neuron (Figure 6) of each layer due to affine transformations by bounding the error of the formula of Equation (11). We compute first the error of multiplication $\hat{w}_{k-1,i,j} \otimes \ \hat{x}_{k-1,j}$ using Proposition 2. Then we bound the fixed-point sum of multiplications $\sum_{j=0}^{n-1} (\hat{w}_{k-1,i,j} \otimes \hat{x}_{k-1,j})$ and finally we use

Proposition 1 to bound the error of addition of $\sum_{j=0}^{n-1}(\hat{w}_{k-1,i,j} \otimes \hat{x}_{k-1,j}) \oplus \hat{b}_{k-1,i}$.

Let $\epsilon_\alpha \in \mathbb{R}$ be the error of $\hat{w}_{k-1,i,j} \otimes \hat{x}_{k-1,j}$ and $2^{-L^{\hat{u}}_{k,i}}$ the truncation error of the output neuron uˆ . Using Proposition 2, we obtain

$$\epsilon_\alpha \leq 2^{M^{\hat{x}}_{k-1,j}-L^{\hat{w}}_{k-1,i,j}} + 2^{M^{\hat{w}}_{k-1,i,j}-L^{\hat{x}}_{k-1,j}} + 2^{-L^{\hat{u}}_{k,i}}. \tag{13}$$

Now, let us consider $\epsilon_\beta \in \mathbb{R}$ as the error of $\sum_{j=0}^{n-1} (\hat{w}_{k-1,i,j} \otimes \hat{x}_{k-1,j})$. This error is computed by using the result of Equation (13) such that

$$\epsilon_\beta \leq \sum_{j=0}^{n-1}(2^{-L^{\hat{w}}_{k-1,i,j}} \times 2^{M^{\hat{x}}_{k-1,j}} + 2^{-L^{\hat{x}}_{k-1,j}} \times 2^{M^{\hat{w}}_{k-1,i,j}}) + \sum_{j=0}^{n-2} 2^{-L^{\hat{u}}_{k,i}} + 2^{-L^{\hat{u}}_{k,i}}. \tag{14}$$

Consequently,

$$\epsilon_\beta \leq \sum_{j=0}^{n-1}(2^{M^{\hat{x}}_{k-1,j}-L^{\hat{w}}_{k-1,i,j}} + 2^{M^{\hat{w}}_{k-1,i,j}-L^{\hat{x}}_{k-1,j}}) + n \times 2^{-L^{\hat{u}}_{k,i}}. \tag{15}$$

Finally, let $\epsilon_\gamma \in \mathbb{R}$ be the error of $\sum_{j=0}^{n-1} (\hat{w}_{k-1,i,j} \otimes \hat{x}_{k-1,j}) \oplus \hat{b}_{k-1,i}$. Using Equation (15) and Proposition 1 we obtain $\epsilon_\gamma \leq \epsilon_\beta + 2^{-L^{\hat{u}}_{k,i}}$. Finally,

$$\epsilon_\gamma \leq \sum_{j=0}^{n-1}(2^{M^{\hat{x}}_{k-1,j}-L^{\hat{w}}_{k-1,i,j}} + 2^{M^{\hat{w}}_{k-1,i,j}-L^{\hat{x}}_{k-1,j}}) + n \times 2^{-L^{\hat{u}}_{k,i}} + 2^{-L^{\hat{u}}_{k,i}+1}. \tag{16}$$

If we combine Equations (12) and (16), we obtain

$$\theta_{k,i} = \epsilon_\gamma \leq \sum_{j=0}^{n-1}(2^{M^{\hat{x}}_{k-1,j}-L^{\hat{w}}_{k-1,i,j}} + 2^{M^{\hat{w}}_{k-1,i,j}-L^{\hat{x}}_{k-1,j}}) + n \times 2^{-L^{\hat{u}}_{k,i}} + 2^{-L^{\hat{u}}_{k,i}+1}. \quad \blacksquare \tag{17}$$

In this section, we have bounded the affine transformation error $\theta_{k,i}$ for each neuron $\hat{u}_{k,i}$ of each layer k of the NN in Equation (17), respecting the equivalent floating-point computations. This resulting error $\theta_{k,i}$ is used in Section 6 to compute the optimal format $< M^{\hat{u}}_{k,i}, L^{\hat{u}}_{k,i} >$ for each neuron $\hat{u}_{k,i}$.

6 Constraints Generation

In this section, we demonstrate how to generate the linear constraints automatically for a given NN, in order to optimize the number of significant bits after the binary point $L^{\hat{u}}_{k,i}$ of the format $< M^{\hat{u}}_{k,i}, L^{\hat{u}}_{k,i} >$ corresponding to the output $\hat{u}_{k,i}$. Let us remember that we have a floating-point NN with m layers and n neurons per layer working at some precision, and we want to compute a fixed-point NN with the same behavior than the initial floating-point NN for a given input vector. This new fixed-point NN must respect the threshold error and the data type $T \in \{8,\ 16,\ 32\}$ bits for the C synthesized code. The variables of the system of constraints are $L^{\hat{u}}_{k,i}$ and $L^{\hat{w}}_{k-1,i,j}$. They correspond respectively to the length of the fractional part of the output $\hat{u}_{k,i}$ and $\hat{w}_{k-1,i,j}$. We have $M^{\hat{u}}_{k,i}$, $M^{\hat{x}}_{k-1,i}$, $M^{\hat{w}}_{k-1,i,j} \in \mathbb{Z}$, and $L^{\hat{u}}_{k,i}$, $L^{\hat{x}}_{k-1,i}$, $L^{\hat{w}}_{k-1,i,j} \in \mathbb{N}$, for $1 \leq k < m+1$, and $0 < i,\ j < n$, such that $M^{\hat{u}}_{k,i}$ (respectively $M^{\hat{x}}_{k-1,i}$, $M^{\hat{w}}_{k-1,i,j}$) can be negative when the value of the floating-point number is between -1 and 1. We have also, $M^{\hat{w}}_{k-1,i,j}$ (respectively $M^{\hat{x}}_{0,i}$ the number of bits before the binary point of the feature layer) which is obtained by computing the ufp defined in

Equation (4) of the corresponding floating-point coefficient. Finally, the value of $M^{\hat{u}}_{k,i}$ is obtained through the fixed-point arithmetic (addition and multiplication) in Section 4. In Equation (18) of Figure 7 (respectively (19) and (20)), the length $M^{\hat{x}}_{k,i} + L^{\hat{x}}_{k,i}$ (respectively $M^{\hat{u}}_{k,i} + L^{\hat{u}}_{k,i}$ and $M^{\hat{w}}_{k,i,j} + L^{\hat{w}}_{k,i,j}$) of the fixed-point number $\hat{x}$ (respectively $\hat{u}$ and $\hat{w}$) must be less than or equal to $T-1$ to ensure the data type required. We use $T-1$ in these three constraints because we keep one bit for the sign. Equation (21) is about the multiplication. It asserts that the total number of bits of $\hat{x}$ and $\hat{w}$ is not exceeding the data type $T-1$. Equations (22), (23) and (24) assert that the number of significant bits of the fractional parts cannot be negative. The boundary condition for the neurons of the output layer is represented in Equation (25). It gives a lower bound for $L^{\hat{u}}_{m,i}$ and then ensures that the error threshold is satisfied for all the neurons of the output layer.

$$M^{\hat{x}}_{k,i} + L^{\hat{x}}_{k,i} \leq T-1, \qquad 0 \leq k \leq m,\ 0 \leq i < n \tag{18}$$

$$M^{\hat{u}}_{k,i} + L^{\hat{u}}_{k,i} \leq T-1, \qquad 1 \leq k < m+1,\ 0 \leq i < n \tag{19}$$

$$M^{\hat{w}}_{k,i,j} + L^{\hat{w}}_{k,i,j} \leq T-1, \qquad 0 \leq k < m,\ 0 \leq i,j < n \tag{20}$$

$$M^{\hat{w}}_{k,i,j} + L^{\hat{w}}_{k,i,j} + M^{\hat{x}}_{k,j} + L^{\hat{x}}_{k,j} \leq T-1, \qquad 0 \leq k < m,\ 0 \leq i,j < n \tag{21}$$

$$L^{\hat{x}}_{k,i} \geq 0, \qquad 0 \leq k \leq m,\ 0 \leq i < n \tag{22}$$

$$L^{\hat{u}}_{k,i} \geq 0, \qquad 1 \leq k < m+1,\ 0 \leq i < n \tag{23}$$

$$L^{\hat{w}}_{k,i,j} \geq 0, \qquad 0 \leq k < m,\ 0 \leq i,j < n \tag{24}$$

$$L^{\hat{u}}_{m,i} \geq |\mathrm{ufp}(|Threshold|)|, \quad 0 \leq i < n,\ m : \text{last layer of NN} \tag{25}$$

$$\forall j\ :\ L^{\hat{u}}_{k,i} \leq L^{\hat{x}}_{k-1,j}, \quad 1 \leq k < m+1,\ 0 \leq i,j < n \tag{26}$$

$$L^{\hat{x}}_{k,i} \leq L^{\hat{u}}_{k,i}, \quad 1 \leq k < m+1,\ 0 \leq i < n \tag{27}$$

$$L^{\hat{u}}_{k,i} \times (\mathrm{ufp}(n)+1) + \sum_{j=0}^{n-1}(L^{\hat{x}}_{k-1,j} + L^{\hat{w}}_{k-1,i,j}) \geq \sum_{j=0}^{n-1}(M^{\hat{x}}_{k-1,j} + M^{\hat{w}}_{k-1,i,j}) - \mathrm{ufp}(|Threshold|) - 1, \quad 1 \leq k < m+1,\ 0 \leq i,j < n \tag{28}$$

Fig. 7. Constraints generated for the formats optimization of each neuron of the NN.

In Figure 7, Equation (26) represents the constraint where the propagation is done in a forward way, and Equation (27) represents the constraint where the propagation is done in a backward way. These constraints bound the length of the fractional parts in the worst case. The constraint of Equation (26) aims at giving an upper bound of $L^{\hat{u}}_{k,i}$. It ensures that $L^{\hat{u}}_{k,i}$ of the output of the neuron i of the layer k is less than (or equal to) all its inputs $L^{\hat{x}}_{k-1,j}$, $0 \leq j < n$. The constraint of Equation (27) gives an upper bound for $L^{\hat{x}}_{k,i}$ of the input $\hat{x}$. This constraint ensures that the number of significant bits after the binary point $L^{\hat{x}}_{k,i}$ of the input of the neuron i of the layer $k+1$ is equal to (or less than) $L^{\hat{u}}_{k,i}$ of the neuron i of the previous layer k. The constraint of Equation (28) in Figure 7 aims at bounding $L^{\hat{u}}_{k,i}$ of the output of the neuron i for the layer k and $L^{\hat{w}}_{k-1,i,j}$ of the coefficients of matrix $\hat{W}_{k-1}$. This constraint corresponds to the error done during the computation of the affine transformation in Equation (11). The Equation (28) is obtained by the linearization of Equation (12) of Proposition 3, in other words, we have to compute the ufp of the error. The ufp of the error, written ufp($\theta_{k,i}$), is computed as follow

Using Equation (17) of Proposition 3, we have

$$\text{ufp}(\theta_{k,i}) \leq \text{ufp}(\sum_{j=0}^{n-1}(2^{M^{\hat{x}}_{k-1,j}-L^{\hat{w}}_{k-1,i,j}} + 2^{M^{\hat{w}}_{k-1,i,j}-L^{\hat{x}}_{k-1,j}}) + n \times 2^{-L^{\hat{u}}_{k,i}} + 2^{-L^{\hat{u}}_{k,i}+1}),$$

then we obtain

$$\text{ufp}(\theta_{k,i}) \leq \sum_{j=0}^{n-1}(M^{\hat{x}}_{k-1,j} - L^{\hat{w}}_{k-1,i,j} + M^{\hat{w}}_{k-1,i,j} - L^{\hat{x}}_{k-1,j}) - L^{\hat{u}}_{k,i} \times (\text{ufp}(n) + 1) + 1.$$

We notice that $\text{ufp}(\theta_{k,i}) \leq \text{ufp}(|Threshold|) \leq 0$ because the error is between 0 and 1.

Finally, $\text{L}^{\hat{u}}_{k,i}\times(\text{ufp}(n) + 1) + \sum_{j=0}^{n-1}(L^{\hat{x}}_{k-1,j} + L^{\hat{w}}_{k-1,i,j}) \geq \sum_{j=0}^{n-1}(M^{\hat{x}}_{k-1,j} + M^{\hat{w}}_{k-1,i,j}) - \text{ufp}(|Threshold|) - 1.$■

All the constraints defined in Figure 7 are linear with integer variables. The optimal solution is found by solving them by linear programming. This solution gives the minimal number of bits for the fractional part required for each neuron $\hat{u}_{k,i}$ of each layer taking into account the data type T and the error threshold tolerated by the user in one hand, and on the other hand the minimal number of bits of the fractional part required for each coefficient $\hat{w}_{k-1,i,j}$.

7 Implementation

In this section, we present our tool. Our approach which is computing the optimal formats $< M^{\hat{u}}_{k,i}, L^{\hat{u}}_{k,i} >$ for each neuron $\hat{u}_{k,i}$ of each layer for a given NN, satisfying an error threshold between 0 and 1 and a data type T given by the user is evaluated through this tool.

Our tool is a fixed-point code synthesis tool. It synthesizes a C code for a given NN. This code contains arithmetic operations and activation functions, which use the fixed-point arithmetic (integer arithmetic) only. In this article, we present only the $\widehat{ReLU}$ and $\widehat{Linear}$ activation functions (defined in Equation (5) and (6) respectively) but we can also deal with $\widehat{Sigmoid}$ and $\widehat{Tanh}$ activation functions in our current implementation. They are not shown but they are available in our framework. We have chosen to approximate them through piecewise linear approximation [6] using fixed-point arithmetic. We compute the corresponding error like in $\widehat{ReLU}$ and $\widehat{Linear}$, then we generate the corresponding constraints.

A description of our tool is given in Figure 8. It takes a floating-point NN working at some precision, input vectors and a threshold error chosen by the user. The user also has the possibility to choose the data type $T \in \{8, 16, 32\}$ bits wanted for the code synthesis. First, we do a preliminary analysis of the NN through many input vectors in order to determine the range of the outputs of the neurons for each layer in the worst case. We compute also the most significant bit of each neuron of each layer in the worst case which gives us the range of the inputs of the NN $[x_{min}, x_{max}]$ for which we certify that our synthesized code is valid for any input belong this range respecting the required data type and the threshold. Our tool generates automatically the constraints mentioned in Figure 7 in Section 6 and solves them using the linprog function of scipy library [30] in Python.

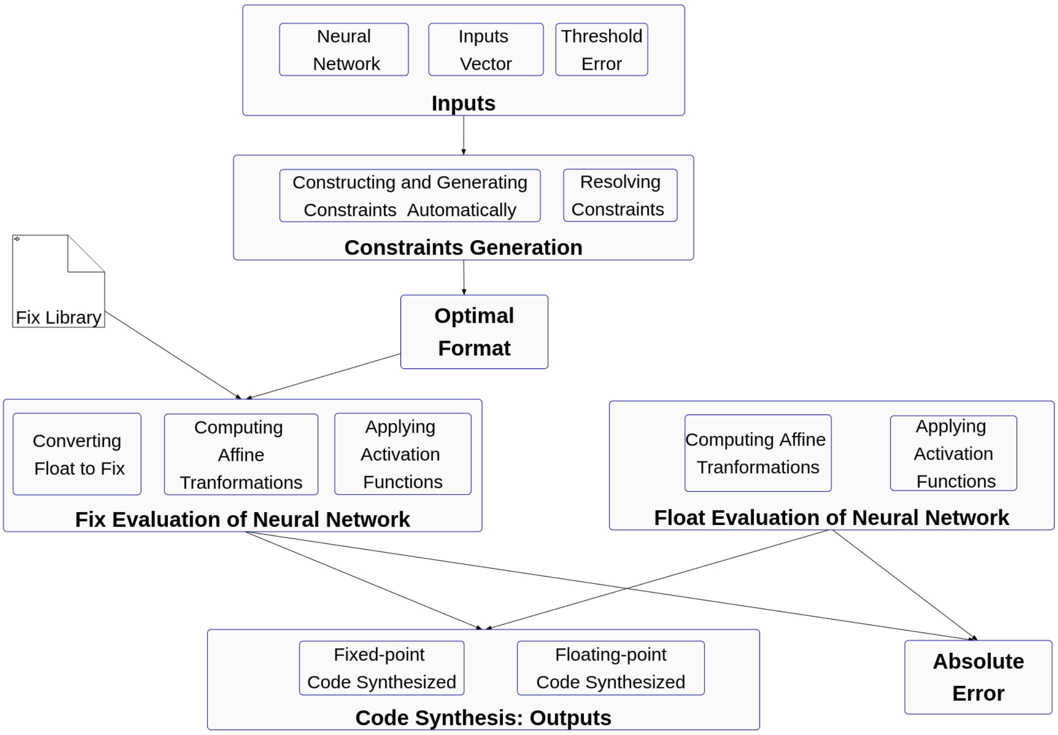

Neural Network
Inputs Vector
Threshold Error
Inputs
Constructing and Generating Constraints Automatically
Resolving Constraints
Constraints Generation
Fix Library
Optimal Format
Converting Float to Fix
Computing Affine Tranformations
Applying Activation Functions
Fix Evaluation of Neural Network
Computing Affine Tranformations
Applying Activation Functions
Float Evaluation of Neural Network
Fixed-point Code Synthesized
Floating-point Code Synthesized
Code Synthesis: Outputs
Absolute Error

Fig. 8. Our tool features.

Then, the optimal formats for each output neuron, input neuron and coefficients of biases and matrices of weights are obtained. The optimal formats are used for the conversion from the floating-point into fixed-point numbers for all the coefficients inside each neuron. To make an evaluation of the NN in fixed-point arithmetic i.e computing the function in Equation (11), a fixed-point library is needed. Our library contains mainly the following functions: fixed-point addition and multiplication, shifts, fixed-point activation functions $\hat{Tanh}$, $\hat{Sigmoid}$, $\hat{ReLU}$ and $\hat{Linear}$. The conversion of a fixed-point number to a floating-point number and the conversion of a floating-point number to a fixed-point number is also available in this library. The last step consists of the fixed-point code synthesis. We also synthesize a floating-point code to make a comparison with the fixed-point synthesized code. We show experiments and results of some NNs in Section 8, and we compare them with floating-point results in terms of memory space (bits saved) and accuracy.

8 Experiments

In this section, we show some experimental results done on seven trained NNs. Four of them are interpolators which compute mathematical functions and the three others are classifiers. These NNs are described in Table 2. The first column gives the name of the NNs. The second column gives the number of layers and the third one shows the number of neurons. The number of connections between the neurons is shown in the fourth column and the number of generated constraints for each NN by our approach is given in the last column.

The NN *hyper* in Table 2 is an interpolator network computing the hyperbolic sine of the point (x, y). It is made of four layers, 48 fully connected neurons (576 connections.)

Table 2. Description of our experimental NNs.

NN \ Desc.	Layers	Neurons	Connections	Constraints
hyper	4	48	576	1980
bumps	2	60	1800	5010
CosFun	4	40	400	1430
Iris	3	33	363	1243
Wine	2	52	1352	3822
Cancer	3	150	7500	21250
AFun	2	200	10000	51700

The number of constraints generated for this NN in order to compute the optimal formats is 1980. The NN *bumps* is an interpolator network computing the *bump* function. The affine function $f(x) = 4 \times x + \frac{3}{2}$ is computed by the *AFun* NN (interpolator) and the function $f(x, y) = x \times cos(y)$ is computed by the *CosFun* NN (interpolator.) The classifier *Iris* is a NN which classifies the *Iris* plant [27] into three classes: *Iris-Setosa*, *Iris-Versicolour* and *Iris-Virginica*. It takes four numerical attributes as input (sepal length in cm, sepal width in cm, petal length in cm and petal width in cm.) The NN *Wine* is also a classifier. It classifies wine into three classes [2] through thirteen numerical attributes (alcohol, malic acid, ash, etc.) The last one is the *Cancer* NN which classifies the cancer into two categories (malignant and benign) through thirty numerical attributes [31] as input. These NNs originally work in IEEE754 single precision. We have transformed them into fixed-point NNs satisfying a threshold error and a data type T (the size of the fixed-point numbers in the synthesized code) set by the user. Then we apply the $\widehat{ReLU}$ activation function defined in Equation (5) or the $\widehat{Linear}$ activation function defined in Equation (6) (or $\widehat{Sigmoid}$ and $\widehat{Tanh}$ also.)

8.1 Accuracy and Error Threshold

The first part of experiments is for accuracy and error threshold. It concerns Table 3, Figure 9 and Figure 11 and shows if the concerned NNs satisfy the error threshold set by the user using the data type T. If the NN is an interpolator, it means that the output of the mathematical function f has an error less than or equal to the threshold. If the NN is a classifier, it means that the error of classification of the NN is less than or equal to $(threshold \times 100)\%$.

The symbol × in Table 3 refers to the infeasability of the solution when the linear programming [30] fails to find a solution or when we cannot satisfy the threshold using the data type T. The symbol $\surd$ means that our linear solver has found a solution to the system of constraints (Section 6.) Each line of Table 3 corresponds to a given NN in some precision and the columns correspond to the multiple values of the error thresholds. For example, in the first line of the first column, the NN *hyper_32* requires a data type $T = 32$ bits and satisfies all the values of threshold (till $10^{-6} \approx 2^{-20}$.) In the fifth column, 2^{-4} means that we require at least four significant bits in the fractional part of the worst output of the last layer of the NNs. The fixed-point NNs *bumps_32, AFun_32, CosFun_32, Iris_32* and *Cancer_32* fulfill the threshold value 2^{-10} which corresponds to ten accurate bits in the fractional part of the worst output. Beyond this value, the linear programming [30] does not find a solution using a data type on 32 bits for these NNs. Using the data type $T = 16$ bits, all the NNs except *AFun_16* have an error less than or equal to 2^{-4} and ensure the correctness at least of four bits after the binary point of the worst output in the last layer. Only the NNs *Wine_8* and *Cancer_8* can ensure one significant bit after the binary point using data type on 8 bits. In the other NNs, only the integer part is correct.

The results can vary depending on several parameters: the input vector, the coefficients of W and b, the activation functions, the error threshold and the data type T. Generally, when the coefficients are between -1 and 1, the results are more accurate because their ufp (Equation(4)) are negative and we can go far after the binary point. The infeasability of the solutions depends also on the size of the data types T, for example if we have a small data type T and a consequent number of bits before the binary point in the coefficients (large value of M), we cannot have enough bits after the binary point to satisfy the small error thresholds.

Figure 9 represents the fixed-point outputs of the interpolator *CosFun_32* (lines) and the floating-point outputs (shape) for multiple inputs. All the fixed-point outputs must respect the threshold 2^{-10} (ten significant bits in the fractional part) and must use the data type $T = 32$ bits in this case. We can see that the two curves are close. This means that the new NN (fixed-point) has the same behavior and answer comparing to the original NN. The result is correct for this NN for any inputs $x \in [-4, 4]$ and $y \in [-4, 4]$.

Table 3. Comparison between the multiple values of error thresholds set by the user and our tool experimental errors using data types on 32, 16 and 8 bits for the fixed-point synthesized code.

Data Types	Threshold NN	$10^0 = 2^0$	$0.5 = 2^{-1}$	$10^{-1} \approx 2^{-4}$	$10^{-2} \approx 2^{-7}$	$10^{-3} \approx 2^{-10}$	$10^{-4} \approx 2^{-14}$	$10^{-5} \approx 2^{-17}$	$10^{-6} \approx 2^{-20}$
32 bits	*hyper_32*	√	√	√	√	√	√	√	√
	bumps_32	√	√	√	√	√	×	×	×
	AFun_32	√	√	√	√	√	×	×	×
	CosFun_32	√	√	√	√	√	×	×	×
	Iris_32	√	√	√	√	√	×	×	×
	Wine_32	√	√	√	√	√	√	×	×
	Cancer_32	√	√	√	√	√	×	×	×
16 bits	*hyper_16*	√	√	√	×	×	×	×	×
	bumps_16	√	√	√	×	×	×	×	×
	AFun_16	√	√	×	×	×	×	×	×
	CosFun_16	√	√	√	×	×	×	×	×
	Iris_16	√	√	√	×	×	×	×	×
	Wine_16	√	√	√	×	×	×	×	×
	Cancer_16	√	√	√	×	×	×	×	×
8 bits	*hyper_8*	×	×	×	×	×	×	×	×
	bumps_8	√	×	×	×	×	×	×	×
	AFun_8	×	×	×	×	×	×	×	×
	CosFun_8	×	×	×	×	×	×	×	×
	Iris_8	√	×	×	×	×	×	×	×
	Wine_8	√	√	×	×	×	×	×	×
	Cancer_8	√	√	×	×	×	×	×	×

Figure 11 shows the results of the fixed-point classifications for the NNs *Iris_32* (right) and *Wine_32* (left) using a data type $T = 32$ bits and a threshold value 2^{-7} for multiple input vectors (= 8). For example, the output corresponding to the input vectors 1 and 2 of the *Iris_32* NN is *Iris-Versicolour*, for the input vectors 3, 5 and 6 *Iris-Virginica* and *Iris-Setosa* for the others. The results are interpreted in the same way for the *Wine_32* NN. We notice that we have obtained the same classifications with the floating-point NNs using the same input vectors, so our new NN has the same behavior as the initial NN.

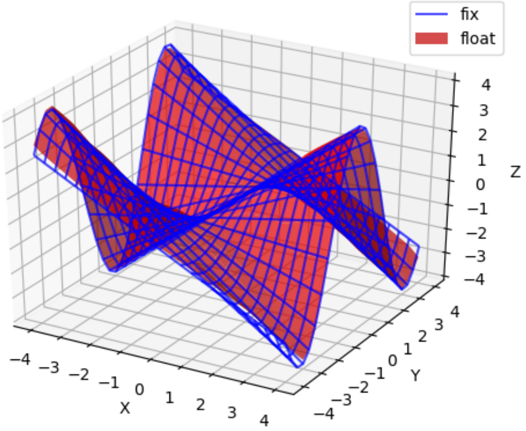


Fig. 9. Fixed-point outputs vs floating-point outputs of the *CosFun_32* NN for multiple inputs using a data type on 32 bits and an error threshold $\approx 2^{-10}$.

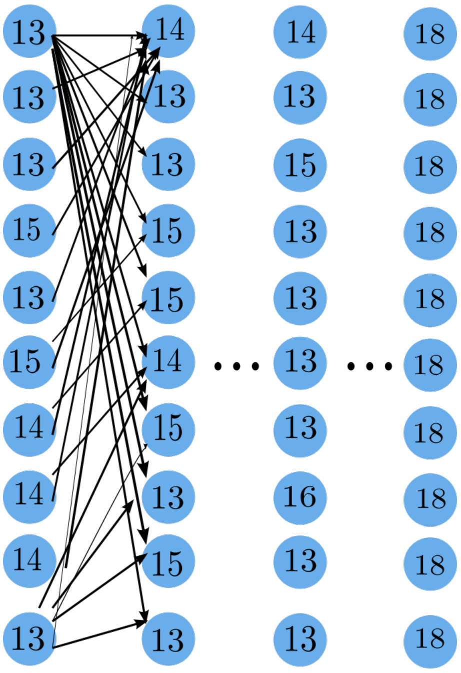


Fig. 10. Number of bits of each neuron of the *CosFun_32* NN after formats optimization for a threshold $\approx 2^{-10}$.

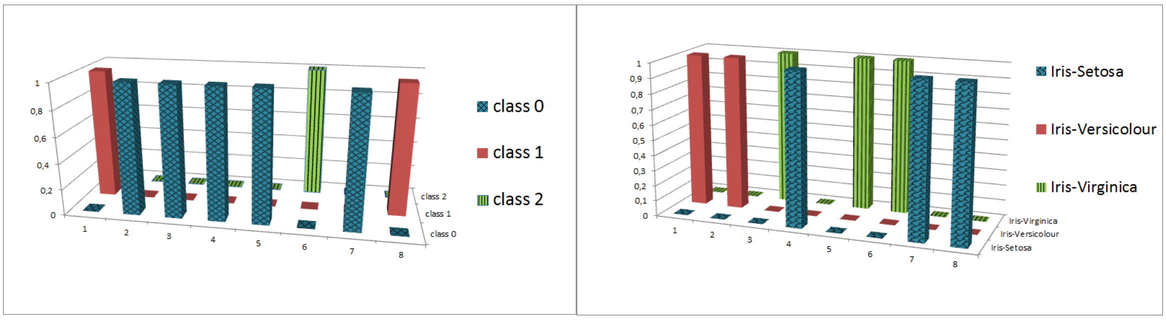


Fig. 11. Results of the fixed-point classification of the NNs *Iris_32* (right) and *Wine_32* (left) respecting the threshold value 2^{-7} and the data type $T = 32$ bits.

8.2 Bits/Bytes Saved

The second part of experiments concerns Figure 10, Figure 12 and Table 4 and aims at showing that our approach saves bytes/bits through the computation of the optimal format for each neuron done in Section 6. At the beginning, all the neurons are represented in $T \in \{8, 16, 32\}$ bits and after the optimization step, we reduce consequently the number of bits for each neuron while respecting the threshold set by the user.

Figure 10 shows the total number of bits of all the neurons for each layer of the *CosFun_32* NN after the optimization of the formats $< M, L >$ in order to satisfy the threshold 2^{-10} and the data type $T = 32$ bits for the fixed-point synthesized code in this case. In the synthesized C code, the data types will not change through this optimization because they are defined at the beginning of the program statically, but it is interesting to use these results in FPGA [4,11] for example. We notice that the size of all the neurons was 32 bits at the beginning and after our tool optimization (resolving the constraints of

Section 6), we need only 18 bits for the neurons of the output layer to satisfy the error threshold. We win 14 bits which represents a gain of 43.75%.

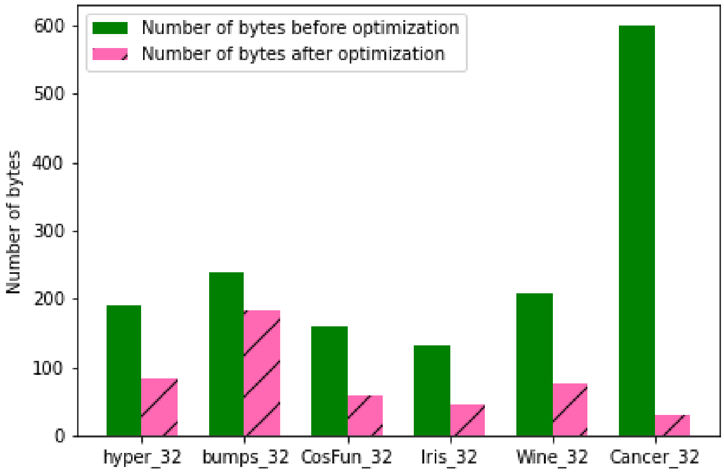


Fig. 12. Initial vs saved bytes for our experimental NNs on 32 bits respecting the threshold 2^{-7}.

In the initial NNs, all the neurons were represented on 32 bits (4 bytes) but after the formats optimization (Section 6) through linear programming [30], we save many bytes for each neuron of each layer and the size of the NN becomes considerably small. It is useful in the case when we use FPGA [4,11], for example. Figure 12 shows the size of each NN (bytes) before and after optimization, and the Table 4 demonstrates the percentage of gain for each NN. For example, in the NN *Cancer_32*, we reduce for 18× the number of bytes comparing to the initial NN (we earn up 94, 66%.) Our approach saves bits and takes in consideration the threshold error set by the user (in this example it is 2^{-7}.)

Table 4. Gain of bytes of the experimental NNs after formats optimization for a threshold $\approx 2^{-7}$ and a data type $T = 32$ bits.

Desc. NN	Size before opt.	Size after opt.	Bytes saved	Gain (%)
hyper_32	192	84	108	**56, 25**
bumps_32	240	183	57	**23.75**
CosFun_32	160	59	101	**63.12**
Iris_32	132	47	85	**64.39**
Wine_32	208	77	131	**62.98**
Cancer_32	600	32	568	**94.66**

8.3 Conclusion

These experimental results show the efficiency of our approach in terms of accuracy and bits saved (memory space.) As we can see in Figure 2, the synthesized code contains only assignments, elementary operations (+, ×, >>, <<) and conditions used in the activation functions. The execution time corresponding to the synthesized code for the experimental NNs of the Table 2 is in only few milliseconds (ms).

9 Related Work

Recently, a new line of research has emerged on compressing machine learning models [15, 16], using other arithmetics in order to run NNs on devices with small memories, integer CPUs [11, 12] and optimizing data types and computations error [14, 18].

In this section, we give an overview of some recent work. We present the multiple tools and frameworks (SEEDOT [11], DEEPSZ [15], Condensa [16]) more or less related to our approach. There is no approach comparable with our method because none of them respects a threshold error set by the user in order to synthesize a C code using only integers for a given trained NN without modifying its behavior. We can cite also FxpNet [7] and Fix-Net [9] which train neural networks using fixed point arithmetic (low bit-width arithmetic) in both forward pass and backward pass. The articles [9, 19] are about quantization which aims to reduce the complexity of DNNs and facilitate potential deployment on embedded hardware. There is also another line of research who has emerged recently on understanding safety and robustness of NNs [8, 10, 17, 25, 28]. We can mention the frameworks Sherlock [8], AI^2 [10], DeepPoly [25] and NNV [28].

The SEEDOT framework [11] synthesizes a fixed-point code for machine learning (ML) inference algorithms that can run on constrained hardware. This tool presents a compiling strategy that reduces the search space for some key parameters , especially scale parameters for the fixed-point numbers representation used in the synthesized fixed-point code. Some operations are implemented (multiplication, addition, exponential, argmax, etc.) in this approach. Both of SEEDOT and our tool generate fixed-point code, but our tool fullfills a threshold and a data type required by the user. SEEDOT finds a scale for the fixed-point representation number and our tool solves linear constraints for finding the optimal format for each neuron. The main goal of this compiler is the optimization of the fixed-point arithmetic numbers and operations for an FPGA and micro-controllers.

The key idea of [14] is to reduce the sizes of data types used to compute inside each neuron of the network (one type per neuron) working in IEEE754 floating-point arithmetic [13, 21]. The new NN with smaller data types behaves almost like the original NN with a percentage error tolerated. This approach generates constraints and does a forward and a backward analysis to bound each data type. Our tool has a common step with this approach, which is the generation of constraints for finding the optimal format for each neuron (fixed-point arithmetic) for us and the optimal size (floating-point arithmetic) for each neuron for this method.

In [12], a new data type called Float-Fix is proposed. This new data type is a trade-off between the fixed-point arithmetic [4,20,23] and the floating-point arithmetic [13,21]. This approach analyzes the data distribution and data precision in NNs then applies this new data type in order to fulfill the requirements. The elementary operations are designed for Float-Fix data type and tested in the hardware. The common step with our approach is the analysis of the NN and the range of its output in order to find the optimal format using the fixed-point arithmetic for us and the the optimal precision for this method using Float-Fix data type. Our approach takes a threshold error not to exceed but this approach does not.

DEEPSZ [15] is a lossy compression framework. It compresses sparse weights in deep NNs using the floating-point arithmetic. DEEPSZ involves four key steps: network pruning, error bound assessment, optimization for error bound configuration and compressed model generation. A threshold is set for each fully connected layer, then the weights of this layer are pruned. Every weight below this threshold is removed. This framework determines the best-fit error bound for each layer in the network, maximizing the overall compression ratio with user acceptable loss of inference accuracy.

The idea presented in [16] is about using weight pruning and quantization for the compression of deep NNs [26]. The model size and the inference time are reduced without appreciable loss in accuracy. The tool introduced is Condensa where the reducing memory footprint is by zeroing out individual weights and reducing inference latency is by pruning 2-D blocks of non-zero weights for a language translation network (Transformer).

A framework for semi-automatic floating-point error analysis for the inference phase of deep learning is presented in [18]. It transforms a NN into a C++ code in order to analyze the network need for precision. The affine and interval arithmetics are used in order to compute the relative and absolute errors bounds for deep NN [26]. This article gives some theoretical results which are shown for bounding and interpreting the impact of rounding errors due to the precision choice for inference in generic deep NN [26].

10 Conclusion & Future Work

In this article, we introduced a new approach to synthesize a fixed-point code for NNs using the fixed-point arithmetic and to tune the formats of the computations and conversions done inside the neurons of the network. This method ensures that the new fixed-point NN still answers correctly compared to the original network based on IEEE754 floating-point arithmetic [13]. This approach ensures the non overflow (sufficient bits for the integer part) of the fixed-point numbers in one hand and the other hand, it respects the threshold required by the user (sufficient bits in the fractional part.) It takes in consideration the propagation of the round off errors and the error of inputs through a set of linear constraints among integers, which can be solved by linear programming [30]. Experimental results show the efficiency of our approach in terms of accuracy, errors of computations and bits saved. The limit of the current implementation is the large number of constraints. We use linprog in Python [30] to solve them but this method does not support a high number of constraints, this is why our experimental NNs are small.

A first perspective is about using another solver to solve our constraints (Z3 for example [22]) which deals with a large number of constraints. A second perspective is to make a comparison study between Z3 [22] and linprog [30] in term of time execution and memory consumption. A third perspective is to test our method on larger, real-size industrial neural networks. We believe that our method will scale up as long as the linear programming solver will scale up. If this is not enough, a solution would be to assign the same format to a group of neurons in order to reduce the number of equations and variables in the constraints system. A last perspective is to consider the other NN architectures like convolutional NNs [1, 3, 10].

References

1. Abraham, A.: Artificial neural networks. Handbook of measuring system design (2005)
2. Aeberhard, S., Coomans, D., de Vel, O.: The classification performance of RDA. Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, Tech. Rep pp. 92–01 (1992)
3. Albawi, S., Mohammed, T.A., Al-Zawi, S.: Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET). pp. 1–6. IEEE (2017)
4. Bečvář, M., Štukjunger, P.: Fixed-point arithmetic in FPGA. Acta Polytechnica **45**(2) (2005)
5. Catrina, O., de Hoogh, S.: Secure multiparty linear programming using fixed-point arithmetic. In: Computer Security - ESORICS 2010, 15th European Symposium on Research in Computer Security. vol. 6345, pp. 134–150. Springer (2010)
6. Çetin, O., Temurtaş, F., Gülgönül, Ş.: An application of multilayer neural network on hepatitis disease diagnosis using approximations of sigmoid activation function. Dicle Medical Journal/Dicle Tip Dergisi **42**(2) (2015)

7. Chen, X., Hu, X., Zhou, H., Xu, N.: Fxpnet: Training a deep convolutional neural network in fixed-point representation. In: 2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017. pp. 2494–2501. IEEE (2017)
8. Dutta, S., Jha, S., Sankaranarayanan, S., Tiwari, A.: Output range analysis for deep feedforward neural networks. In: NASA Formal Methods - 10th International Symposium, NFM. vol. 10811, pp. 121–138. Springer (2018)
9. Enderich, L., Timm, F., Rosenbaum, L., Burgard, W.: Fix-net: pure fixed-point representation of deep neural networks. ICLR (2019)
10. Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., Vechev, M.T.: AI2: safety and robustness certification of neural networks with abstract interpretation. In: 2018 IEEE Symposium on Security and Privacy. pp. 3–18. IEEE Computer Society (2018)
11. Gopinath, S., Ghanathe, N., Seshadri, V., Sharma, R.: Compiling KB-sized machine learning models to tiny IoT devices. In: Programming Language Design and Implementation, PLDI 2019. pp. 79–95. ACM (2019)
12. Han, D., Zhou, S., Zhi, T., Wang, Y., Liu, S.: Float-fix: An efficient and hardware-friendly data type for deep neural network. Int. J. Parallel Program. **47**(3), 345–359 (2019)
13. IEEE: IEEE standard for floating-point arithmetic. IEEE Std 754-2008 pp. 1–70 (2008)
14. Ioualalen, A., Martel, M.: Neural network precision tuning. In: Quantitative Evaluation of Systems, 16th International Conference, QEST. vol. 11785, pp. 129–143. Springer (2019)
15. Jin, S., Di, S., Liang, X., Tian, J., Tao, D., Cappello, F.: Deepsz: A novel framework to compress deep neural networks by using error-bounded lossy compression. In: In International Symposium on High-Performance Parallel and Distributed Computing, HPDC. pp. 159–170. ACM (2019)
16. Joseph, V., Gopalakrishnan, G., Muralidharan, S., Garland, M., Garg, A.: A programmable approach to neural network compression. IEEE Micro **40**(5), 17–25 (2020)
17. Katz, G., Barrett, C.W., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: An efficient SMT solver for verifying deep neural networks. In: Computer Aided Verification, CAV. vol. 10426, pp. 97–117. Springer (2017)
18. Lauter, C.Q., Volkova, A.: A framework for semi-automatic precision and accuracy analysis for fast and rigorous deep learning. In: 27th IEEE Symposium on Computer Arithmetic, ARITH 2020. pp. 103–110. IEEE (2020)
19. Lin, D.D., Talathi, S.S., Annapureddy, V.S.: Fixed point quantization of deep convolutional networks. In: Balcan, M., Weinberger, K.Q. (eds.) Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016. JMLR Workshop and Conference Proceedings, vol. 48, pp. 2849–2858. JMLR.org (2016)
20. Lopez, B.: Implémentation optimale de filtres linéaires en arithmétique virgule fixe. (Optimal implementation of linear filters in fixed-point arithmetic). Ph.D. thesis, Pierre and Marie Curie University, Paris, France (2014)
21. Martel, M.: Floating-point format inference in mixed-precision. In: NASA Formal Methods - 9th International Symposium, NFM. vol. 10227, pp. 230–246 (2017)
22. de Moura, L.M., Bjørner, N.: Z3: an efficient SMT solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings. Lecture Notes in Computer Science, vol. 4963, pp. 337–340. Springer (2008)
23. Najahi, M.A.: Synthesis of certified programs in fixed-point arithmetic, and its application to linear algebra basic blocks. Ph.D. thesis, University of Perpignan, France (2014)
24. Sharma, S., Sharma, S.: Activation functions in neural networks. Towards Data Science **6**(12), 310–316 (2017)
25. Singh, G., Gehr, T., Püschel, M., Vechev, M.T.: An abstract domain for certifying neural networks. Proc. ACM Program. Lang., POPL **3**, 41:1–41:30 (2019)
26. Sun, Y., Huang, X., Kroening, D., Sharp, J., Hill, M., Ashmore, R.: Testing deep neural networks. arXiv preprint arXiv:1803.04792 (2018)
27. Swain, M., Dash, S.K., Dash, S., Mohapatra, A.: An approach for iris plant classification using neural network. International Journal on Soft Computing **3**(1), 79 (2012)
28. Tran, H., Yang, X., Lopez, D.M., Musau, P., Nguyen, L.V., Xiang, W., Bak, S., Johnson, T.T.: NNV: the neural network verification tool for deep neural networks and learning-enabled cyber-physical systems. In: Computer Aided Verification - 32nd International Conference, CAV. vol. 12224, pp. 3–17. Springer (2020)
29. Urban, C., Miné, A.: A review of formal methods applied to machine learning. CoRR **abs/2104.02466** (2021)
30. Welke, P., Bauckhage, C.: Ml2r coding nuggets: Solving linear programming problems. Tech. rep., Technical Report. MLAI, University of Bonn (2020)

31. Wolberg, W.H., Street, W.N., Mangasarian, O.L.: Breast cancer wisconsin (diagnostic) data set [http://archive.(ics. uci. edu/ml/] (1992)
32. Yates, R.: Fixed-point arithmetic: An introduction. Digital Signal Labs **81**(83), 198 (2009)

Authors

H Benmaghnia received a Master degree in High Performance Computing and Simulations from the University of Perpignan in France. She did a Bachelor of Informatic Systems at the University of Tlemcen in Algeria. Currently, she is pursuing her PhD in Computer Science at the University of Perpignan in LAboratoire de Modélisation Pluridisciplinaire et Simulations (LAMPS). Her research interests include Computer Arithmetic, Precision Tuning, Numerical Accuracy, Neural Networks and Formal Methods.

M Martel is a professor in Computer Science at the University of Perpignan in LAboratoire de Modélisation Pluridisciplinaire et Simulations (LAMPS), France. He is also co-founder and scientific advisor of the start-up Numalis, France. His research interests include Computer Arithmetic, Numerical Accuracy, Abstract Interpretation, Semantics-based Code Transformations & Synthesis, Validation of Embedded Systems, Safety of Neural Networks & Arithmetic Issues, Green & Frugal Computing, Precision Tuning & Scientific Data Compression.

Y Seladji received PhD degree in computer science from Ecole Polytechnique, France. Currently, she is associate professor in the University of Tlemcen. Her research interests include Formal Methods, Static Analysis, Embedded Systems.

RESEARCH ON DUAL CHANNEL NEWS HEADLINE CLASSIFICATION BASED ON ERNIE PRE-TRAINING MODEL

Junjie Li and Hui Cao

Key Laboratory of China's Ethnic Languages and Information Technology of Ministry of Education, Northwest Minzu University Lanzhou, China

ABSTRACT

The classification of news headlines is an important direction in the field of NLP, and its data has the characteristics of compactness, uniqueness and various forms. Aiming at the problem that the traditional neural network model cannot adequately capture the underlying feature information of the data and cannot jointly extract key global features and deep local features, a dual-channel network model DC-EBAD based on the ERNIE pre-training model is proposed. Use ERNIE to extract the lexical, semantic and contextual feature information at the bottom of the text, generate dynamic word vector representations fused with context, and then use the BiLSTM-AT network channel to secondary extract the global features of the data and use the attention mechanism to give key parts higher The weight of the DPCNN channel is used to overcome the long-distance text dependence problem and obtain deep local features. The local and global feature vectors are spliced, and finally passed to the fully connected layer, and the final classification result is output through Softmax. The experimental results show that the proposed model improves the accuracy, precision and F1-score of news headline classification compared with the traditional neural network model and the single-channel model under the same conditions. It can be seen that it can perform well in the multi-classification application of news headline text under large data volume.

KEYWORDS

Text Classification, ERNIE, Dual-Channel, BiLSTM, Attention, DPCNN.

1. INTRODUCTION

The essence of news headline classification is text classification, which aims to classify the target text data accurately. In today's era of exponential growth of data information, online news has become the most important and efficient carrier of social public information and even international information dissemination. As the core of news headlines, it is the finishing touch. A vivid and unique headline can quickly attract readers' attention and its authenticity can optimize the network information environment to a certain extent, guide public opinion, and convey the correct value orientation [1].

However, in the era of big data, information is complicated and confusing, news is being generated all the time, and news headlines are constantly increasing. All kinds of news data are efficiently processed, and the use of deep learning to identify and classify news headlines has gradually become a hot topic. Although traditional neural network models such as LSTM, CNN, BiLSTM, and DPCNN can better classify news headlines, they are not enough, and there is still

David C. Wyld et al. (Eds): AISCA, NET, DNLP - 2022
pp. 31-43, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120203

room for improvement. In recent years, pre-training models such as ERNIE have emerged rapidly, which can extract the underlying lexical, semantic, and contextual information of text data, and generate corresponding dynamic word vector representations that integrate contextual context for the input text data. Therefore, the new trend is to combine the pre-trained language model and the deep neural network model, so as to better perform word vector representation and corresponding feature extraction on text data. However, the single-channel network model generally follows the fixed process of the word vector representation of the input text and the feature extraction. Therefore, in order to better perform feature extraction in different ranges and splicing representation of different learned features, a dual-channel neural network model DC-EBAD based on ERNIE pre-trained language model is proposed. The ERNIE pre-training model is used to represent the input text data with dynamic word vector representation of the context, and the BiLSTM-AT channel is used to extract global features and use the attention mechanism to give higher weights to key parts, and the deep local features extracted by DPCNN are spliced with it to form the final feature vector, then the learning of features in different ranges is completed, so as to achieve the propose of extracting and learning features in an all-round way.

2. RELATED WORK

In machine learning, natural language processing tasks focus on feature selection and feature extraction of input data. After learning and training, the final classification result is obtained through the classifier. Li X et al.[2] used term frequency-inverse document frequency (TF-IDF) to calculate the feature weights of keywords, and carried out weighting processing on the naive Bayes algorithm to realize the classification of Internet hot news text data. . In order to strengthen the feature expression of the input text vector, the pre-training model has become the mainstream of vector feature expression. Yuejun X et al. [3] used BERT pre-training to fuse the trained sentence vector with the continuously updated proper noun vector and assign the TF-IDF value of the proper noun as the weight to the vector, which solved the existing patent classification. The method is limited by the problem of unregistered words in the patent text. Chengyu Q et al.[4] proposed a semi-supervised classification method based on graph convolutional neural network (BD-GCN) in response to the problems of lack of annotations in online bidding documents, sparse text semantics, diverse data sources, and complex information structure. Information extraction technology constructs bidding document data into a special knowledge graph model and integrates external text information, and uses graph convolutional neural network to realize semi-supervised classification of bidding documents. Zemin H et al. [5] proposed a text sentiment classification model combining BERT and BiSRU-AT, using BERT to obtain dynamic word vector representations, using BiSRU to extract semantic features and context information twice, and then integrating the Attention mechanism to assign weights to the output to solve the traditional semantics. The model cannot solve the problems of polysemous word representation and the existing sentiment analysis model cannot fully capture long-distance semantic information. Di W et al.[6] proposed an ELMo-CNN-BiGRU dual-channel text sentiment classification method, which uses the ELMo and Glove pre-training models to generate dynamic and static word vectors respectively and stack embedding to generate input vectors, constructing a fusion convolutional neural network(CNN) and bidirectional gated recurrent unit (BiGRU) two-channel neural network model to obtain the local and global features of the text. Keming C et al.[7] classify ANN and text, construct vector space to describe text and extract features of different types of text through text segmentation and word frequency statistics, and use ANN features for learning to complete text classification tasks. The globalization environment is increasingly demanding natural speech processing tasks in small languages. HossainMd. R et al.[8] proposed intelligent text classification models including GloVe embedding and ultra-deep convolutional neural network (VDCNN) classifiers and embedding parameter recognition (EPI) algorithm to select the best embedding parameters for low-resource languages, so that resource-constrained languages can be better used for natural language processing tasks. In summary, the most critical step for natural language processing tasks is the extraction of features, which directly affects the quality of text classification results.

On this basis, for network news headlines that are short and succinct and have different styles of data, a dual-channel model (Dual-Channel ERNIE -BiLSTM with Attention and DPCNN, DC-EBAD), through ERNIE for the underlying text data to perform dynamic word vector representation that contains lexical, semantic and contextual information as the input of the subsequent network channel, BiLSTM overcomes the inability of a one-way LSTM network to reverse Extract the shortcomings of the global semantic information of the text, and use the attention mechanism to give higher weight to the key parts of the global features of the bi-directional text sequence vector extracted by BiLSTM to strengthen the classification feature expression; at the same time, the deep pyramid convolutional neural network model is built to overcome the traditional volume The product neural network CNN cannot obtain the long-distance dependence of the text through convolution, and performs deep-level local feature extraction while greatly saving calculation time. Through the splicing of the output feature vector of the dual-channel model as the input of the Softmax fully connected layer classification, the news headline classification result is finally obtained. The results show that it has a higher accuracy rate than the set comparison model.

3. DC-EBAD DUAL-CHANNEL MODEL

Based on the ERNIE pre-training model combined with the attention mechanism of the bidirectional long-term short-term memory network channel and the deep pyramid convolutional network dual-channel model DC-EBAD structure is shown in the following figure.

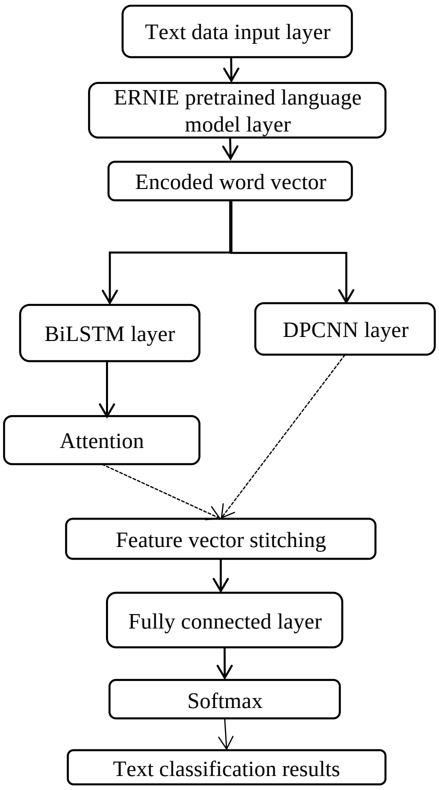


Figure 1. DC-EBAD network model structure

As shown in Figure.1 above, the input of the DC-EBAD neural network model is news headline text data, which is entered as headline short sentences. In the pre-training language model layer, it first passes through the internal word embedding layer to perform static word vector encoding in units of words. Then pass it to the ERNIE layer, and finally generate the corresponding dynamic word vector representation combined with the context. Then it is input into BiLSTM-AT and DPCNN network channels in two channels. In the BiLSTM-AT model channel, perform the global feature acquisition of the secondary context information on the incoming word vector and calculate the weight distribution coefficient through the attention mechanism to give more attention to the key parts; in the DPCNN channel, the incoming text vector Perform deep-level local feature extraction and obtain long text clustering dependencies; finally, the text feature vectors output by the two channels are spliced into the fully connected layer and then the final classification result is calculated by Softmax.

3.1. ERNIE

The knowledge-enhanced representation through knowledge integration (ERNIE) [9] is proposed by Baidu, which uses multi-source data and related prior knowledge for pre-training. The formation process of the input vector of the ERNIE model is the same as that of BERT [10]. In the short sentence-level text classification, the sentence is used as the initial input, and the word embedding is encoded as a static word vector in the unit of word. The sentence embedding and the corresponding position embedding are added together as the input of the ERNIE layer. The input vector formation process of the ERNIE layer is shown in Fig. 2 The input vector formation process of the ERNIE model.

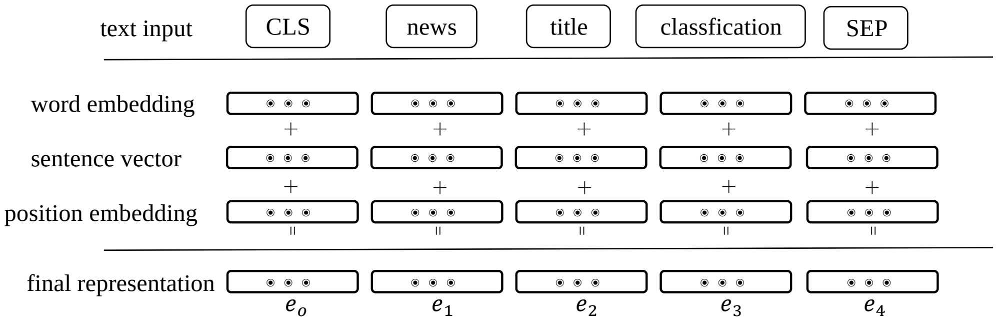


Figure 2. ERNIE model input vector formation process

In Figure.2 above, [CLS] represents a placeholder for the beginning of a sentence, which contains the information of the entire sentence; [SEP] represents a separator used to distinguish different sentences. The static word vector $\{e_o, e_1, e_2, e_3, e_4\}$ is generated by summing the input sentence in the word embedding representation, the whole sentence representation and the position representation vector as the input vector of the "news headline classification" before being passed to the ERNIE layer Characterization. The ERNIE layer extracts the underlying lexical and semantic information from the input vector, and finally generates a dynamic word vector representation that integrates the context.

The granularity of the masking strategy in ERNIE is based on the entity/phrase level. The external knowledge information is not directly input into the model. It is necessary to implicitly learn the knowledge information such as entity relationship and entity attributes, and cover by global information prediction. Content snippets. The concealment strategy is as shown in Figure.3 for a simple illustration of the entity-level concealment in ERNIE.

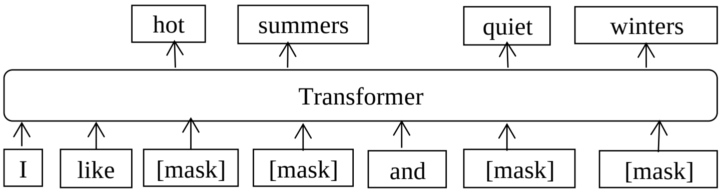


Figure 3. Entity-level masking in ERNIE

The structure design of the ERNIE model is still essentially an encoder based on bidirectional self-attention mechanism. The encoder part comes from the Transformer model and uses a multi-head self-attention mechanism with 12 attention heads. head self-attention) [11], through which to capture the contextual information of the word vector in the text sequence and generate a word vector representation that integrates the context. The relevant formula is calculated as follows.

$$Attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = Softmax\left(\frac{\boldsymbol{QK^T}}{\sqrt{d_k}}\right)\mathbf{V} \quad (1)$$

$$head_1 = Attention(\boldsymbol{Qw_i^Q},\boldsymbol{KW_i^K},\boldsymbol{VW_i^V})(2)$$

$$MultiHead(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) =$$

$$Concat(head_1, \ldots, head_n)\boldsymbol{W^0} \ (3)$$

Among them, $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$represent query matrix, key matrix and value matrix respectively. The function of $\sqrt{d_k}$ is to prevent the inner product from being too large and to overcome the problem of too small gradient that may be caused by high-dimensional matrix operations. $Softmax$ normalizes the calculation result and calculates the weight coefficient of each word. $\boldsymbol{w_i^Q}$,$\boldsymbol{W_i^K}$, and $\boldsymbol{W_i^V}$are weight parameter matrices relative to $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ respectively.

3.2. LSTM and BiLSTM

BiLSTM (bi-drectional long short-term memory, BiLSTM) is a combination of forward long short-term memory (LSTM) and backward LSTM network models. A single LSTM network model can only encode information from front to back. Although it effectively alleviates the long-distance dependence problem of general recurrent neural network (RNN) and the problem of gradient disappearance, it cannot capture the bidirectional semantic features of text vectors. Therefore, using BiLSTM as a channel carrier of the overall model architecture, feature extraction of global contextual semantic information is performed on the input text vector for better feature expression.

The LSTM network model structure is mainly composed of forget gates, inputs, cell state updates, and output gates. It was originally proposed by Hochreiter S and Schmidhuber J [12]. It was the mainstream reference model in the early days of deep learning. The overall network architecture is shown in Figure.4 the simplified structure of the LSTM network model.

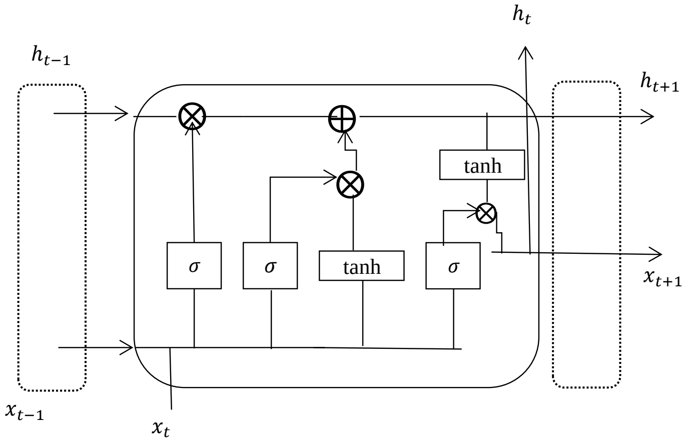


Figure 4. LSTM network model simplified structure.

The activity state between the internal unit structures is shown in formula (4) to formula (9):

$$\boldsymbol{i_t} = \sigma(\boldsymbol{W_e}[\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_i}) \qquad (4)$$
$$\boldsymbol{f_t} = \sigma\left(\boldsymbol{W_f}[\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_f}\right) \qquad (5)$$
$$\boldsymbol{\tilde{c}_t} = tanh(\boldsymbol{W_c}[\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_c}) \qquad (6)$$
$$\boldsymbol{c_t} = \boldsymbol{f_t} \odot \boldsymbol{c_{t-1}} + \boldsymbol{i_t} \odot \boldsymbol{\tilde{c}_t} \qquad (7)$$
$$\boldsymbol{o_t} = \sigma(\boldsymbol{W_o}[\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_o}) \qquad (8)$$
$$h_t = \boldsymbol{o_t} \odot tanh(\boldsymbol{C_t}) \ (9)$$

In the LSTM network model structure, the dashed boxes on the left and right sides represent the structural units of the LSTM at the previous moment and the next moment. In the above formula, $\boldsymbol{i_t}$ represents the input gate, $\boldsymbol{f_t}$ represents the forget gate, and $\mathbf{o_t}$ represents the output gate. This is the classic gate control mechanism (Gate mechanism) in the model. $\boldsymbol{\tilde{c}_t}$ represents the cell state, $\boldsymbol{c_t}$ represents the updated cell state, $\boldsymbol{h_t}$ represents the output of the final hidden layer state, $\boldsymbol{\sigma}$ represents the *Sigmoid* activation function,$\boldsymbol{x_t}$ represents the input at the current moment, $\boldsymbol{b}$ represents the bias value, and $\odot$ represents the Hadamard product.

BiLSTM, as a forward and backward combination model of the LSTM network structure, bidirectionally encodes the input text sequence information through the forward and backward network models, fully considering the contextual semantic information, and can extract more accurate global feature vector expressions. The overall structure is shown in Figure 2.3 BiLSTM network model structure.

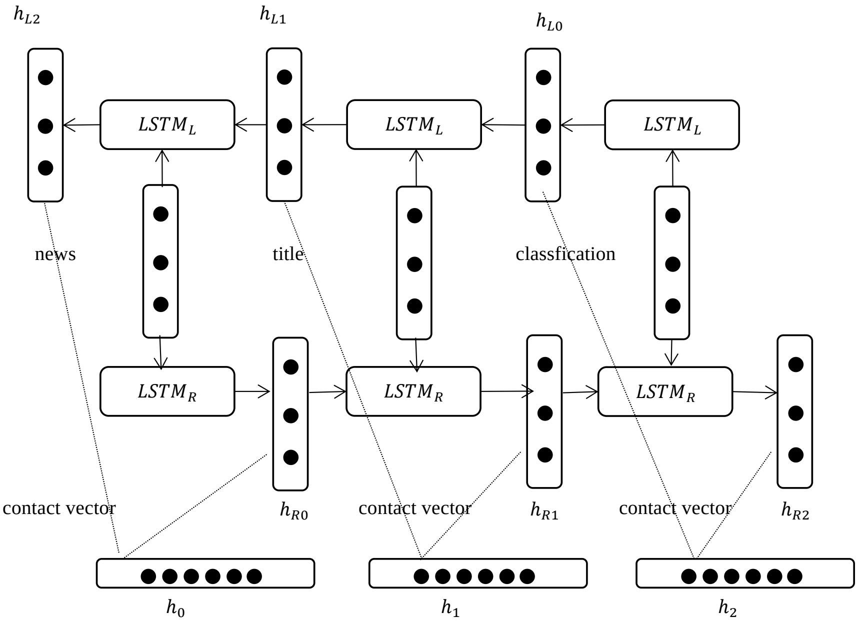


Figure 5. BiLSTM network model structure.

As shown in Figure.5, for the input text sequence "news title classification", the forward input is "news", "title", "category", expressed as a vector $\{h_{R0}, h_{R1}, h_{R2}\}$, and the backward input is "category" , "title", "news", expressed as the vector $\{h_{L0}, h_{L1}, h_{L2}\}$. Then the forward and backward hidden layer state vectors are spliced to obtain $\{[h_{R0}, h_{L0}], [h_{R1}, h_{L1}], [h_{R2}, h_{L2}]\}$, that is, the final result is used as the model place .The text vector to be represented by data is $\{h_0, h_1, h_2\}$.

3.3. BiLSTM-Attention

After the context global feature extraction of the input text vector is completed by BiLSTM, it is given a higher weight to the key parts that play a decisive role through the attention mechanism layer. In essence, it is a weight distribution mechanism. The larger the weight, the more critical the feature of the text vector and the greater the impact on the final news headline classification result. The formula for calculating the attention distribution of the text vector output by BiLSTM is as follows.

$$\boldsymbol{u_t} = tanh(\boldsymbol{W_g H_t} + \boldsymbol{b_g}) \quad (10)$$

$$a_t = \frac{exp(\boldsymbol{u_t})}{\sum exp(\boldsymbol{u_t})} \quad (11)$$

$$\boldsymbol{R} = \sum a_t \boldsymbol{H_t} \quad (12)$$

In the above formula (7)~(9), $\boldsymbol{W_g}$is the weight parameter, $\boldsymbol{b_g}$ is the bias term, $\boldsymbol{H_t}$ is the hidden layer state vector output, $\boldsymbol{u_t}$ is the hidden layer state vector output at each moment and the weight parameter that the model needs to learn Multiply, add the offset term, and de-linearize the vector expression by the *tanh* function. $\boldsymbol{a_t}$is the weight coefficient distribution of each vector feature obtained after the *softmax* function is used to calculate the probability distribution of the weight. The characteristic vector of the final attention distribution is expressed as $\boldsymbol{R}$, which is the result of multiplying the output of the state vector at all times and the calculated weight coefficient and weighting the sum.

The bi-directional long-term and short-term memory network BiLSTM-AT combined with the attention mechanism is mainly used in the text to extract the most important semantic information [13], and its overall model structure is shown in Figure 6 BiLSTM-AT model structure.

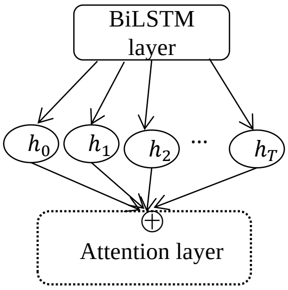


Figure 6. BiLSTM-AT model structure

As shown in Figure.6 above, the feature vector output by the BiLSTM layer is passed to the attention mechanism layer, and the weight coefficient is continuously adjusted through the weighted average to calculate the probability distribution of the attention weight, and the final text feature vector is obtained, which is finally combined with the DPCNN channel The obtained feature vectors are spliced to obtain the final required vector representation.

3.4. DPCNN

The deep pyramid convolutional neural network (deep pyramid convolutional neural network, DPCNN) is a word-level deep convolutional neural network. Compared with the traditional convolutional neural network (convolution neural network, CNN) in text classification tasks The application of [14] mainly overcomes the problem of difficulty in extracting long-distance text sequence dependencies [15], and obtains more accurate local features of text vectors through deep convolution while reducing the amount of calculation. Its model structure Figure 7 shows the structure of the DPCNN network model.

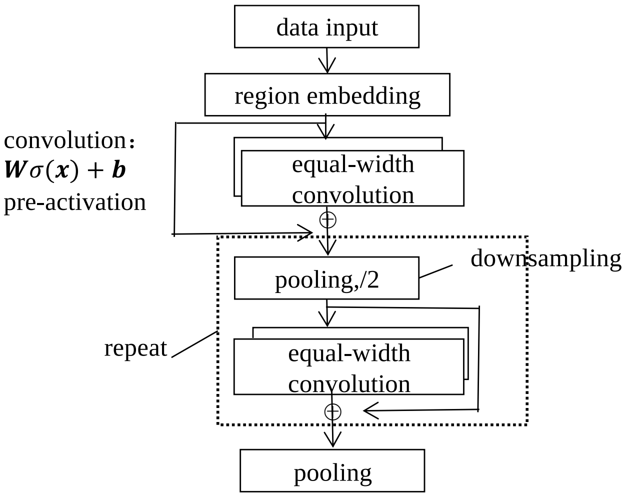


Figure 7. DPCNN network model structure.

As shown in Figure.7 above, Region embedding is essentially a convolutional layer containing convolution filters of different sizes. The input text is convolved here to generate the corresponding word vector encoding. The model contains two convolutional blocks, each convolutional block contains two convolutional layers, and each convolutional layer is equal-width convolution. Assuming that the length of the input sequence is L, the size of the convolution kernel is M, the stride is S, and both ends of the input sequence are filled with N zeros, that is, zero padding. In equal-length convolution, the step size is S=1, the two ends are filled with zero N=(M-1)/2, the output length is still L after convolution, and each layer uses 250 convolution kernels with a size of 3×3 . After each convolution block, a pooling layer with a size of 3 and a step size of 2 is used for 1/2 pooling, that is, downsampling. At the same time, the number of feature maps must be fixed, so the text sequence length is compressed to 1/2 of the original The small fragments of text that can be perceived can be doubled to capture the dependence of long-distance text. And because the size of the text sequence data is halved, the corresponding calculation time will also be halved, thus forming a pyramid structure model. At the same time, use shortcut connect[16] to train the deep learning network, as shown in formula (10), where $\boldsymbol{z}$ is the input vector of the neural network, f stands for skipped layers, and $f(\boldsymbol{z})$ stands for The two-layer equal-length convolution of pre-activation greatly alleviates the problem of gradient dispersion, thereby completing the learning of identity mapping. The pre-activation method is as shown in formula (11).

$$f(\boldsymbol{z})+\boldsymbol{z}(13)$$

$$conv{:}\,\boldsymbol{W\sigma}(\boldsymbol{x})+\boldsymbol{b} \quad (14)$$

That is, when performing a convolution operation in a neural network, directly perform the linear operation of the activation function on the input text vector and then multiply it with the weight parameter $\boldsymbol{W}$ and add it to the offset term $\boldsymbol{b}$, instead of performing the activation function on the entire already-calculated variable Linear conversion. The activation function $\boldsymbol{\sigma}$ in DPCNN represents the *Relu* activation function.

4. EXPERIMENTAL PROCESS AND RESEARCH

4.1. Experimental Data

The news headline classification data set used in the experiment used Tsinghua University's Chinese text classification toolkit THHUCTC[17], and 200,000 news headline data were extracted from the Chinese text classification data set THUCNews, including: finance, real estate , Stocks, education, technology, society, current affairs, sports, games, entertainment ten categories. Each category has a total of 20,000 data, and the length of the text sequence is within 30 words.

4.2. Experimental environment and design

The 200,000 Chinese news headline data sets used in the experiment are set to 180,000 in the training set, 10,000 in the validation set, and 10,000 in the test set. The three data sets contain ten categories, and the number of each category is evenly distributed. , That is, there are 18,000 items per category in the training set, 1,000 items per category in the validation set, and 1,000 items per category in the test set.

In the experiment of the news headline classification task, the model parameters are set as shown in Table 1. The stop_go variable indicates that if the neural network model is training and learning, if the experimental effect of more than 1000 batch_size is not improved, the training will be terminated and the result will be output.

Table 1. DC-EBAD Model Parameters

Model Channel	Parameters	Value
ERNIE Layer	hidden_size	768
BiLSTM-AT-Channel	r_hidden_size	256
	num_layers	2
DPCNN-Channel	num_filters	250
	Kernel_size	3
Hyperparameter	batch_size	128
	text_size	32
	dropout	0.5
	learning_rate	5e-5
	stop_go	1000
	epoch	3

In order to verify the advantages of the proposed model DC-EBAD, single-channel ERN-IE-BiLSTM-AT and ERNIE-DPCNN models us-ing pre-trained language model layers, two-channel BiLSTM-AT-DPCNN models without p-re-training models and other neural networks -are set up Models LSTM, BiLSTM, BiLSTM+Attention, DPCNN, CNN, etc. are used as comparative experiments, and experiments are carried out in the same data set and experiment-al environment.

4.3. Experimental Evaluation Index

The evaluation indicators of news headline classification used in the experiment are accuracy and F1-score. The accuracy rate is to evaluate the classification accuracy of the ten news headline categories as a whole, and the F1-score is for each of the ten categories.

The formula for accuracy, precision, recall and F1-score is defined as follows:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (15)$$

$$precision = \frac{TP}{TP + FP} \quad (16)$$

$$recall = \frac{TP}{TP + FN} \quad (17)$$

$$F1 = \frac{2 * precision * recall}{precision + recall} \quad (18)$$

In the above formula, TP means that the label is true and the prediction result is true; TN means that the label is true and the prediction result is false; FP means the label is false and the prediction result is true, and FN means the label is false and the prediction result is false.

4.4. Analysis of Experimental Results

The experimental evaluation indicators are evaluated from the overall accuracy, precision and F1-score of the data set, as well as the F1-score of each sub-category. For the neural network models ①~⑥ that do not use the ERNIE pre-trained language model, pre-training word vectors are used to construct the vocabulary of the input text data and generate the corresponding word embedding matrix to complete the vectorized representation of the text. Here the pre-training word vector is the word/word pre-training word vector of Sogou News [18]. In the experiment, the input of text data in units of words is completed, and the first line of the pre-training word vector is the word vector File information description: "365076 300", the first number is 365076, that is, the number of words/word vectors contained in the text, and the second indicates that the dimension of the word vector embedding is 300 dimensions. Each subsequent row includes a character/word and its 300-dimensional vector representation.

Table 2. Accuracy of experimental results

Number	Model	Accuracy	Precision	F1-score
①	LSTM	90.69%	90.69%	90.67%
②	CNN	90.74%	90.77%	90.74%
③	DPCNN	91.43%	91.51%	91.45%
④	BiLSTM	90.98%	91.07%	91.01%
⑤	BiLSTM-AT	91.44%	91.49%	91.44%
⑥	**BiLSTM-AT-DPCNN**	**92.02%**	**92.08%**	**92.03%**
⑦	ERNIE-BiLSTM-AT	94.56%	94.58%	94.56%
⑧	ERNIE-DPCNN	94.58%	94.63%	94.59%
⑨	**DC-EBAD**	**94.63%**	**94.66%**	**94.63%**

It can be seen from Table 2 that the conventional CNN and LSTM neural network models performed relatively well on the Chinese news headline data set used in the experiment, with little difference, exceeding 90%. DPCNN, BiLSTM and BiLSTM-AT, as their theoretical optimization models relative to CNN, LSTM and BiLSTM, have also improved various indicators in actual performance. BiLSTM-AT-DPCNN is the dual-channel parallel network model after removing the ERNIE pre-trained language model layer in the DC-EBAD model. Compared with other single-channel neural network comparison models, it is the best and surpasses all three indicators. That's 92%.

After using ERNIE as the pre-training language model layer, the indicators of the three models ⑦~⑨ have reached more than 94%, and the DC-EBAD model has the best performance, surpassing 94.60%, which is more accurate than without ERNIE The dual-channel model BiLSTM-AT-DPCNN of the pre-trained language model layer increased by 2.61%. Compared with the other two single-channel models ERNIE-BiLSTM-AT and ERNIE-DPCNN that use the ERNIE pre-training model, it has increased by 0.07% and 0.05%, respectively.

However, after using the pre-trained language model to represent the word vector of the input text, there will be certain requirements for the hardware facilities of the experimental environment. In this experiment, a single 12G-GPU device is used. In the case of 200,000 pieces of data, The training and learning time of the model has been greatly increased, with ERNIE-DPCNN and DC-EBAD both exceeding 5 hours, while ERNIE-BiLSTM-AT running time exceeds 6 hours.

① ~⑨ in Table III are the corresponding serial numbers of each network model in Table 2.

Table 3. F1-score of each category of experimental results

Category	F1-①	F1-②	F1-③	F1-④	F1-⑤	**F1-⑥**	F1-⑦	F1-⑧	**F1-⑨**
finance	89.92%	90.59%	90.57%	90.76%	**91.10%**	90.89%	**93.63%**	93.09%	93.36%
estate	91.43%	92.41%	92.80%	91.93%	92.39%	**92.89%**	95.69%	**95.81%**	95.68%
stock	84.50%	85.57%	85.48%	84.68%	85.57%	**86.55%**	**90.61%**	90.16%	90.08%
education	93.81%	95.14%	94.43%	94.75%	94.95%	**95.14%**	96.78%	96.82%	**96.92%**
technology	85.37%	86.56%	86.92%	84.88%	87.39%	**88.15%**	91.39%	91.12%	**91.83%**
society	90.73%	88.92%	91.57%	90.76%	90.52%	**91.50%**	94.14%	94.39%	**94.73%**
politics	87.87%	89.30%	89.67%	88.07%	88.66%	**89.74%**	92.79%	**93.22%**	92.26%
sport	97.08%	95.43%	96.88%	**97.73%**	97.01%	97.65%	98.40%	**99.00%**	97.97%
game	93.48%	91.41%	93.50%	93.24%	94.13%	**94.29%**	96.19%	96.00%	**96.68%**
entertainment	92.51%	92.07%	92.68%	93.31%	92.70%	**93.54%**	95.96%	96.25%	**96.79%**

It can be seen from Table 3 that when the usual deep neural network ①~⑥ is used to classify news headlines, the BiLSTM-AT-DPCNN model is slightly inferior to other models in terms of financial and sports classification effects. The performance in the remaining 8 categories is the best. However, it can be seen that the highest F1-score of all models in the stock and technology data sets are 86.55% and 88.15%, respectively. Compared with the highest F1-score of other categories, it can be seen that the classification of the two types of news headline text data is important for the model. It is still somewhat confusing, and the model is still lacking in data feature extraction and learning.

After using the pre-trained language model ERNIE combined with the dual-channel network model, the highest levels of stock and technology category classification reached 90.61% and 91.83%, respectively. Compared with the best-performing dual-channel model BiLSTM- when the pre-trained model was not used, The index on AT-DPCNN increased by 4.06% and 3.68% respectively. The best performance in finance and stocks is the ERNIE-BiLSTM-AT model, and the best performance in real estate, current affairs and sports is the ERNIE-DPCNN model, and its F1-score in sports is as high as 99.00%. Compared with the two single-channel comparison models ERNIE-BiLSTM-AT and ERNIE-DPCNN, the DC-EBAD model has the best performance in the categories of news headlines in education, technology, society, games, and entertainment. The performance of the proposed DC-EBAD Chinese news headline classification model is considerable.

5. CONCLUSION

The DC-EBAD dual-channel model based on the ERNIE pre-training model proposed in the article better realizes the classification of large-scale multi-category news headline data. The pre-training language model is used to extract the lexical and semantic information of the underlying text, and generate input text fusion The dynamic word vector representation of the context information is passed into the dual-channel network to extract the key parts of the global feature and the deep-level local feature extraction respectively, and the splicing vector obtains the final feature vector expression. The evaluation indicators of the experimental results are better than the set comparison model, and most of the F1-score of the sub-class results are also improved. In the future work, considering the optimization of the data input part, such as combining the features of part of speech and other features to further enhance the representation information, so that the neural network can better perform feature learning, and can achieve higher levels in multiple types of news headline tasks.

REFERENCES

[1] Ling W. Discussion on the Importance of News Headlines in New Media [J]. News Research Guide, 2020, 11(22):175-176.

[2] Li X, BoJ, Zhangrui Z. Weighted Naive Bayes News Text Classification Algorithm Based on TF-IDF [J]. Network Security Technology and Application, 2021(11):31-33.

[3] Yuejun X, Honglian L, Le Z , et al. Research on Chinese Patent Text Classification Method Ba-sed on Feature Fusion[J/OL].Data Analysis and Knowledge Discovery:1-14[2021-11-18].http://kns.cnki.net/kcms/detail/10.1478.G2.20211104.1706.002.html.

[4] Chengyu Q, Xiaoge L, Xianyan M, et al. Application of Graph Neural Network in Classificati-on of Bidding Documents [J/OL].Small Microcomputer System:1-7[2021-11-18].http://kns.cnki.net/kcms/detail/21.1106.TP.20211112.1908.002.html.

[5] ZeminH, Xiaolan W, Yinggang W, et al. Chinese text sentiment classification based on BERT and BiSRU-AT [J].Computer Engineering and Science,2021,43(09):1668-1675.

[6] Di W, Ziyu W, Weichao Z. ELMo-CNN-BiGRU dual-channel text sentiment classification metho-d[J/OL]. Computer Engineering:1-10[2021-11-07]. https://doi.org/10.19678/j.issn.1000-3428.0062047.

[7] Keming C, Boyang Z. Text Classification based on ANN [J]. International Core Journal of Engineering, 2021, 7(7):.

[8] Hossain Md. R, Hoque M M, Siddique N, et al. Bengali text document categorization based on very deep convolution neural network[J]. Expert Systems With Applications, 2021,184:.

[9] Sun Y, Wang S, Li Y, et al. ERNIE: Enhanced Representation through Knowledge Integration[J]. 2019.

[10] Jacob D, Ming-Wei C, Kenton L, et al. BERT: pre-training of deep bidirectional transformers for language understanding[C]. Annual Conference of the North American Chapter of the Association for C-omputational Linguistics, Proceedings of Minneapolis, USA, 2019:4171-4186.

[11] Ashish V, Noam S, Niki P, et al. Attention is all you need[C].Neural Information Processing Systems, Proceedings of Long Beach, USA ,2017:5998-6008.

[12] Hochreiter S, Schmidhuber J. Long Short-Term Memory [J]. Neural Computation, 1997, 9(8):1735-1780.

[13] Peng Z, Wei S, Jun T, et, al. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification[C].Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics,2016:207-212.

[14] Yoon K. Convolutional neural networks for sentence classification [C].Conference on Emp-irical Methods in Natural Language Processing, Proceedings of Doha, Qatar, 2014:17 46-1751.

[15] Rie J, Tong Z. Deep Pyramid Convolutional Neural Networks for Text Categorization [C]. In P-roceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017:562-570.

[16] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition [C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016.

[17] Maosong S, Jingyang L, Zhipeng G, et, al. THUCTC: An Efficient Chinese Text Classifier. 2016.

[18] Yuanyuan Q, Hongzheng L, Shen L, et al. Revisiting Correlations between Intrinsic and Extrinsic Evaluations of Word Embeddings. Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. Springer, Cham, 2018.209-221.

DETECTION DATASETS: FORGED CHARACTERS FOR PASSPORT AND DRIVING LICENCE

Teerath Kumar[1], Muhammad Turab[2], Shahnawaz Talpur[2],
Rob Brennan[1] and Malika Bendechache[1]

[1]CRT AI and ADAPT, School of Computing, Dublin City University, Ireland
[2]Department of Computer Systems Engineering, Mehran University of Engineering and Technology, Jamshoro, Pakistan

ABSTRACT

Forged characters detection from personal documents including a passport or a driving licence is an extremely important and challenging task in digital image forensics, as forged information on personal documents can be used for fraud purposes including theft, robbery etc. For any detection task i.e. forged character detection, deep learning models are data hungry and getting the forged characters dataset for personal documents is very difficult due to various reasons, including information privacy, unlabeled data or existing work is evaluated on private datasets with limited access and getting data labelled is another big challenge. To address these issues, we propose a new algorithm that generates two new datasets named forged characters detection on passport (FCD-P) and forged characters detection on driving licence (FCD-D). To the best of our knowledge, we are the first to release these datasets. The proposed algorithm first reads the plain image, then performs forging tasks i.e. randomly changes the position of the random character or randomly adds little noise. At the same time, the algorithm also records the bounding boxes of the forged characters. To meet real world situations, we perform multiple data augmentation on cards very carefully. Overall, each dataset consists of 15000 images, each image with size of 950 x 550. Our algorithm code, FCD-P and FCD-D are publicly available.

KEYWORDS

Character detection dataset, Deep learning forgery, Forged character detection.

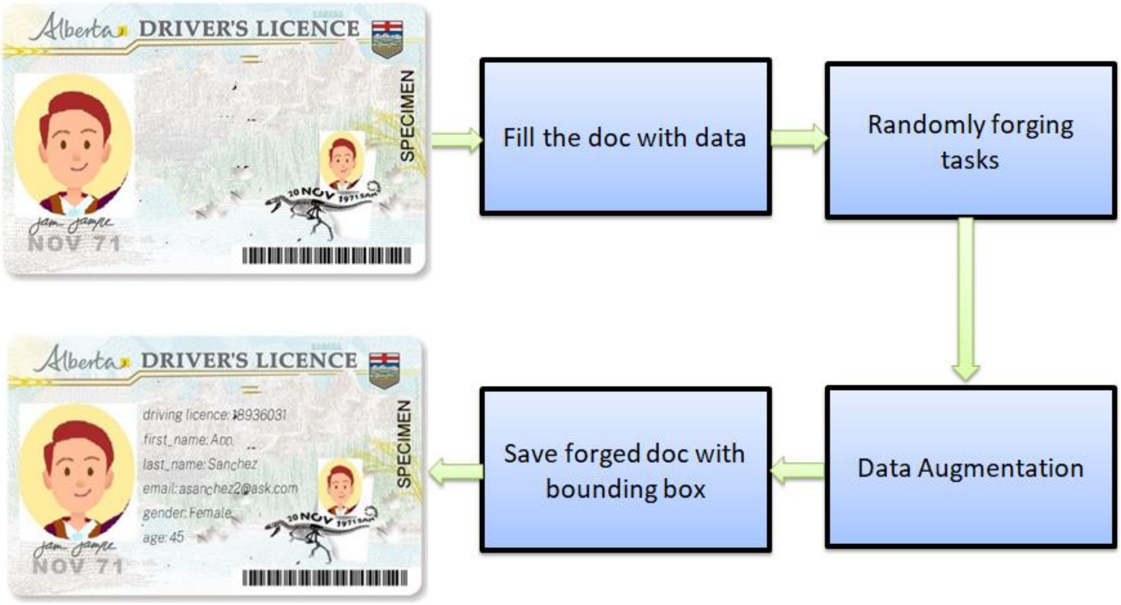
1. INTRODUCTION

Personal documents including passport or driving licence, contain key information of a person and these documents are used for various purposes including critical office work, bank account access, any type of insurance and others. But these documents can easily be modified with deep learning algorithms and used for many fraud purpose including theft, robbery, terrorism, etc. [1]. Recent deep learning algorithms demonstrated that character can easily be forged using convolutional neural networks [2, 3, 4, 5, 6] in sequence to sequence manner. These deep learning algorithms can forge the documents from text, colour or background perspectives, but they are computationally very expensive. To detect the forged characters on the documents is a benchmark challenge. As DL algorithms are data-hungry [7], and to detect the forged characters, these algorithms require the high computational resources and labelled training data. Finding dataset(s) for documents is restricted due to many reasons, information privacy, unlabeled data

David C. Wyld et al. (Eds): AISCA, NET, DNLP - 2022
pp. 45-54, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120204

and many other reasons. To reduce this gap, first we propose an algorithm that generates a dataset using the plain background documents of five different countries, and we are the first to release the synthetic two datasets one for passport and other for driving licence using the algorithm.

Previously several methods have been used for forged character detection [8, 9] from document plain text. Algorithm [8] automatically detects tampered characters from document images by measuring distance between feature vectors of Hu moments. Algorithm calculates possible conception errors by considering principal inertia axis, horizontal axis and character size; further character is classified as real or fake based on the score system. Algorithm [9] detects whether the character is real or fake with the help of geometric parameter distortion mutation, for a single character algorithm estimated distortion parameters based on translation and rotation distortion. This algorithm [10] detects characters from an ID card using a traditional image processing method consisting of four stages: pre-processing, text-area extraction, segmentation, and recognition. In the above mentioned works, either dataset is restricted or algorithms are evaluated on private datasets, to bridge the gap, we propose a new algorithm that generates the forged characters dataset. Furthermore we release two datasets i.e. FCD-P and FCD-D using the proposed algorithm.

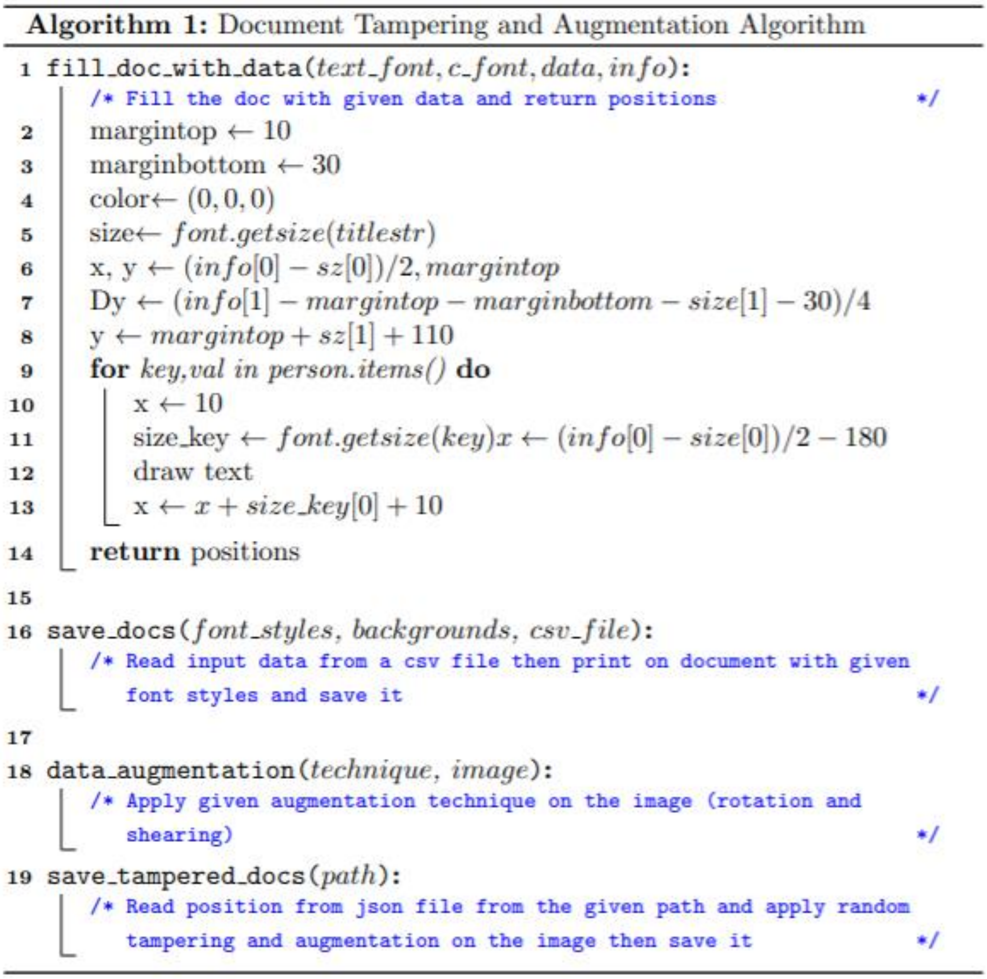


driving licence: 18936031
first_name: Ann
last_name: Sanchez
email: asanchez2@ask.com
gender: Female
age: 45

Figure 1. Proposed Algorithm Workflow

2. Description

Each released dataset consists of 15000 RGB images of dimension 900 x 550 each. First we get a plain background image of either passport or driving licence, taken online of five different countries including Australia, Canada, Ireland, Pakistan and USA, as driving licence and passport images are shown in figure 2 and figure 3, respectively, and their sources of the image acquisition are described in table 1. First we remove unwanted text and images on the passport or driving licence using an online website to make those plain, then we apply the proposed algorithm. The proposed algorithm consists of 4 steps, as shown in Figure 1 and described in algorithm 1. 4 steps are described as following:

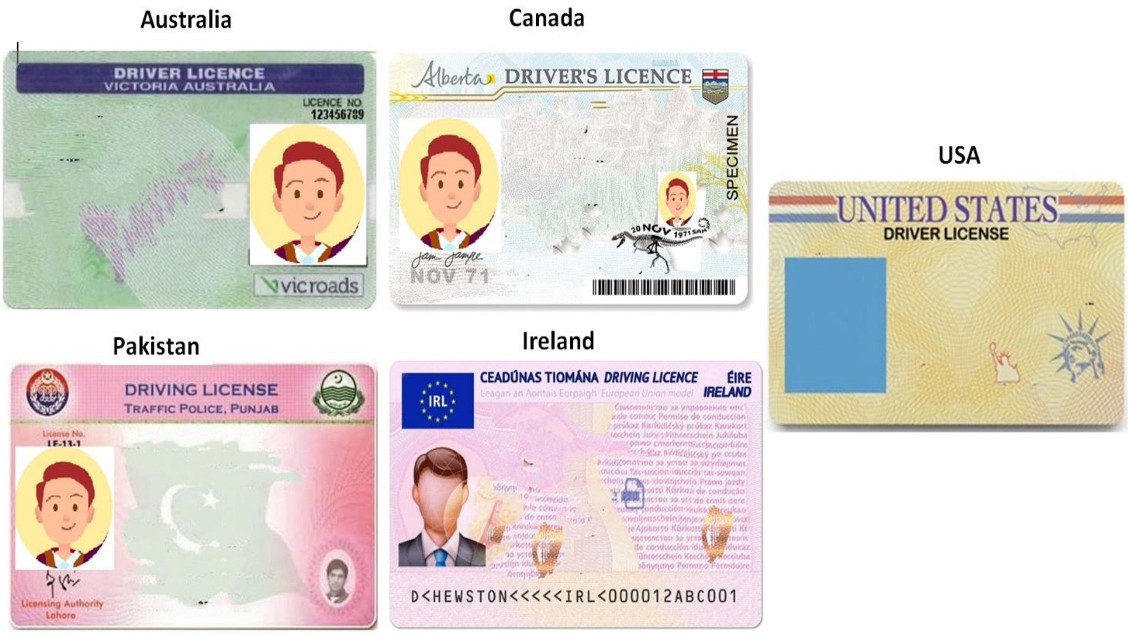


```
Algorithm 1: Document Tampering and Augmentation Algorithm
fill_doc_with_data($text\_font, c\_font, data, info$):
    /* Fill the doc with given data and return positions */
    margintop ← 10
    marginbottom ← 30
    color← (0, 0, 0)
    size← $font.getsize(titlestr)$
    x, y ← $(info[0] - sz[0])/2, margintop$
    Dy ← $(info[1] - margintop - marginbottom - size[1] - 30)/4$
    y ← $margintop + sz[1] + 110$
    for $key,val$ in $person.items()$ do
        x ← 10
        size_key ← $font.getsize(key)x \leftarrow (info[0] - size[0])/2 - 180$
        draw text
        x ← $x + size\_key[0] + 10$
    return positions

save_docs($font\_styles, backgrounds, csv\_file$):
    /* Read input data from a csv file then print on document with given
       font styles and save it */

data_augmentation($technique, image$):
    /* Apply given augmentation technique on the image (rotation and
       shearing) */
save_tampered_docs($path$):
    /* Read position from json file from the given path and apply random
       tampering and augmentation on the image then save it */
```

Algorithm 1. Proposed algorithm

2.1. Fill the document with data

First we read a plain background image of either passport or driving licence using the PIL library in python, then read a csv data file, data [11] was taken from kaggle platform which consist of five attributes including first name, last name, email, gender and age. We get a single record from a data file, and adjust it on a plain background image.


Australia

Canada

USA

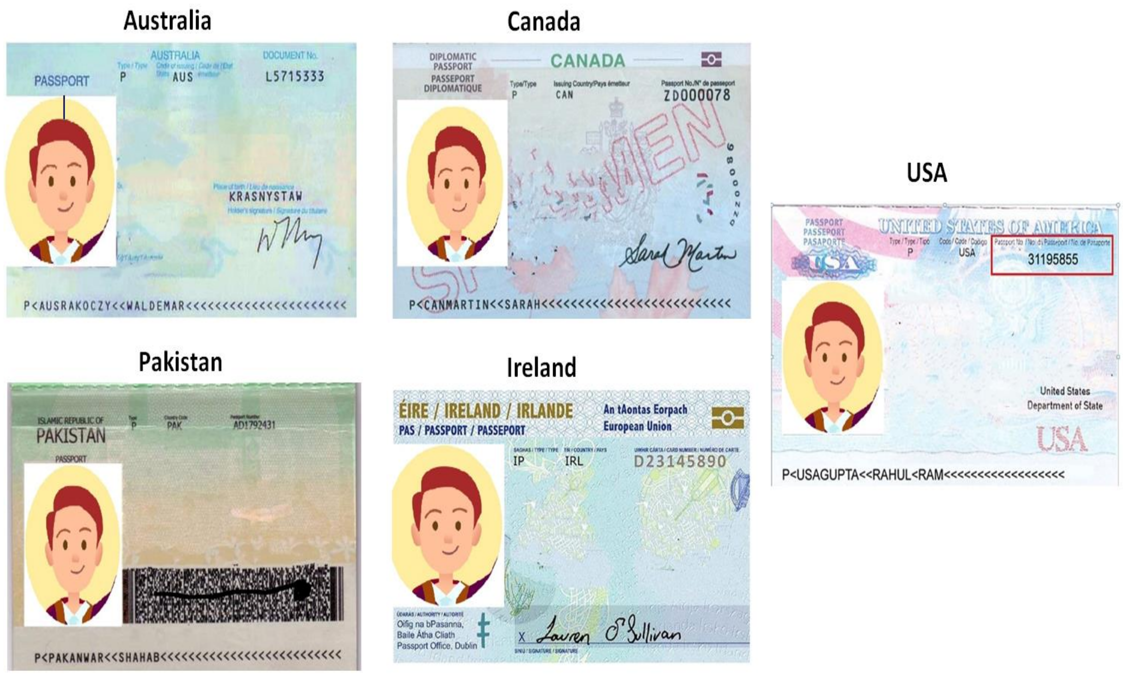
Pakistan

Ireland

Figure 2. Plain Driving Licences

Table 1. Source of passport and licence sample images

Passport	Australia	Canada	Ireland	Pakistan	USA
Driving licence	Australia	Canada	Ireland	Pakistan	USA


Australia

Canada

USA

Pakistan

Ireland

PASSPORT
AUSTRALIA
P
AUS
L5715333
KRASNYSTAW
P<AUSRAKOCZY<<WALDEMAR<<<<<<<<<<<<<<<<<<<<<
CANADA
DIPLOMATIC PASSPORT
PASSEPORT DIPLOMATIQUE
P
CAN
ZD000078
P<CANMARTIN<<SARAH<<<<<<<<<<<<<<<<<<<<<<<<<
UNITED STATES OF AMERICA
PASSPORT
USA
31195855
United States Department of State
USA
P<USAGUPTA<<RAHUL<RAM<<<<<<<<<<<<<<<<<<
PAKISTAN
PAK
AD1792431
P<PAKANWAR<<SHAHAB<<<<<<<<<<<<<<<<<<<<<<<<<
ÉIRE / IRELAND / IRLANDE
PAS / PASSPORT / PASSEPORT
An tAontas Eorpach
European Union
IP
IRL
D23145890
Oifig na bPasanna, Baile Átha Cliath
Passport Office, Dublin

Figure 3. Plain Passports

2.2. Randomly forging tasks

When documents are forged, there are two possibilities, either the forged character is not aligned with other characters or the forged character has a little bit of noise in the background. To keep these possibilities in mind, we randomly pick any character and change character location either up or down as shown in **figure 4 (A)** where in email ‘s’ of the census is moved down, or add a little bit of uniform noise in the character background as shown in **figure 4 (B)** where driving licence, first name, last name and email have a noise in one character of each.

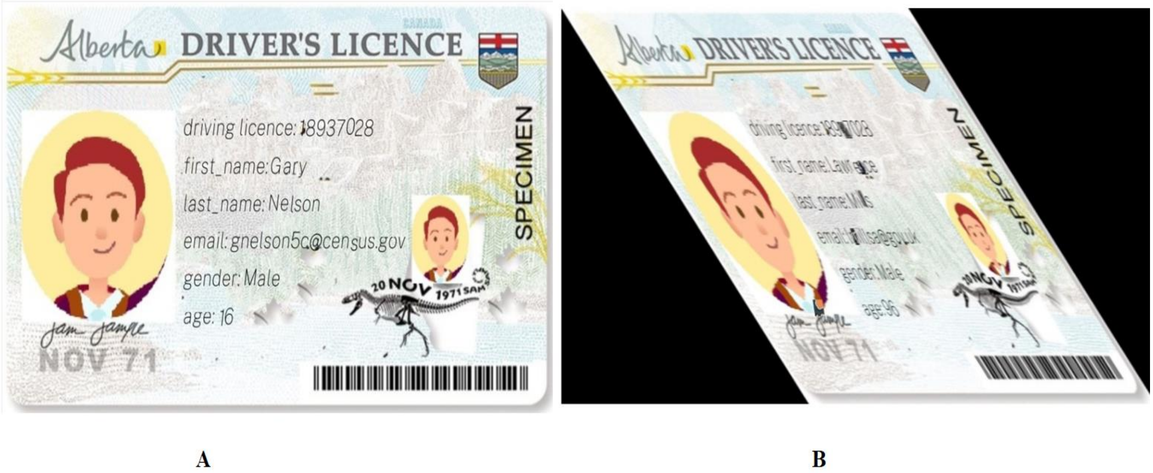


Figure. 4. Forging Tasks

2.3. Data augmentation

In the real world, documents are not placed as straight as shown in the input column of figure 5 and figure 6, documents can be placed at any angle or stretch. To meet the real world scenario, we perform two augmentations namely rotation and shearing for the driving licences and passports image as shown in columns rotation and shearing of each figure 5 and figure 6. The bounding boxes of tampered characters are rotated using the below formula.

$$\begin{bmatrix} alpha & beta & (1-alpha)*x-beta*y \\ -beta & alpha & beta*x+(1-alpha)*y \end{bmatrix}$$

where
*alpha = scale * cos*
*beta = scale * sin*
and theta is the rotation angel
For this case, we use scale=1

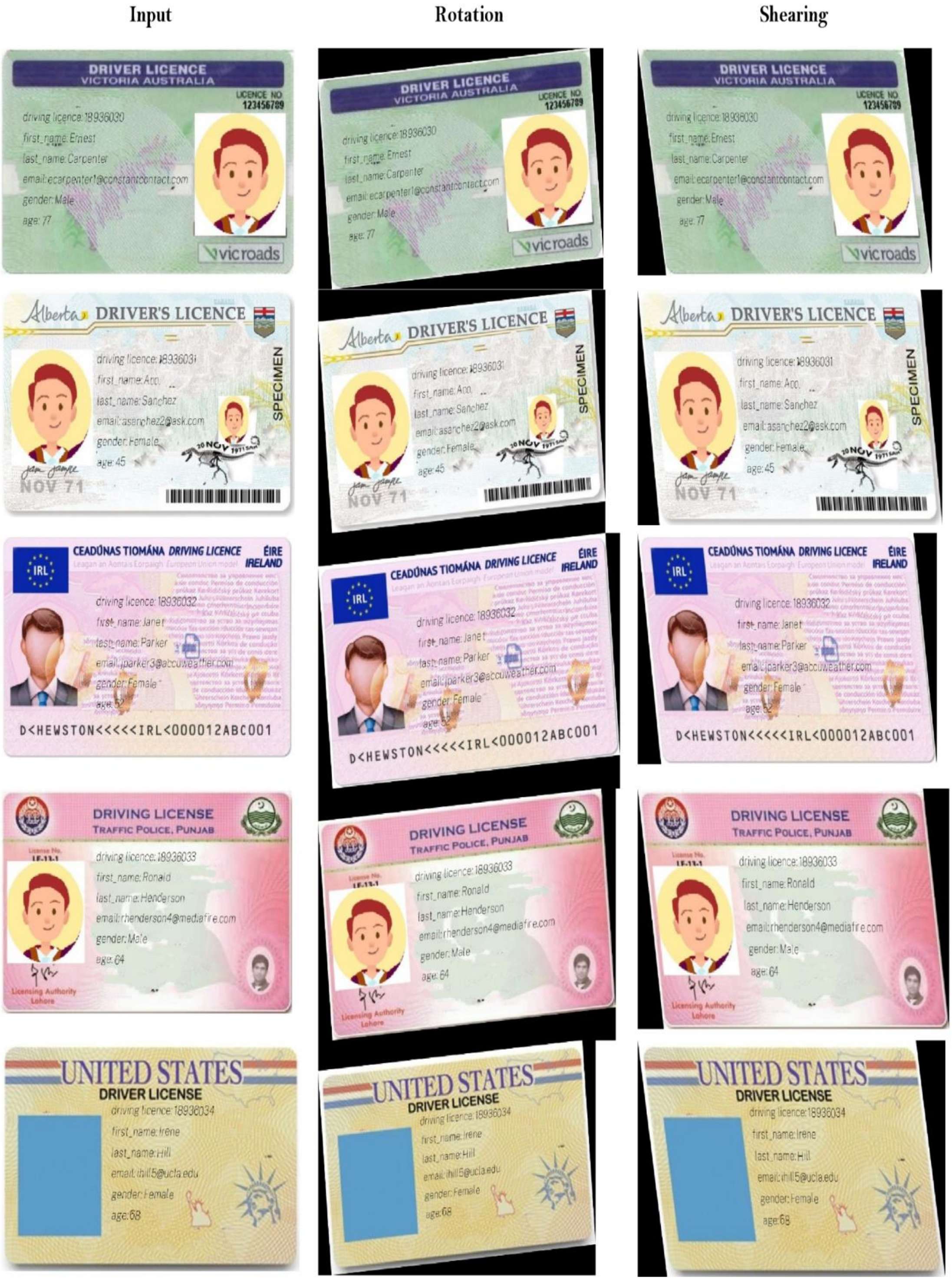


Figure 5. Driving licence with applied data augmentation Right to left, input, rotation augmentation and shearing augmentation.

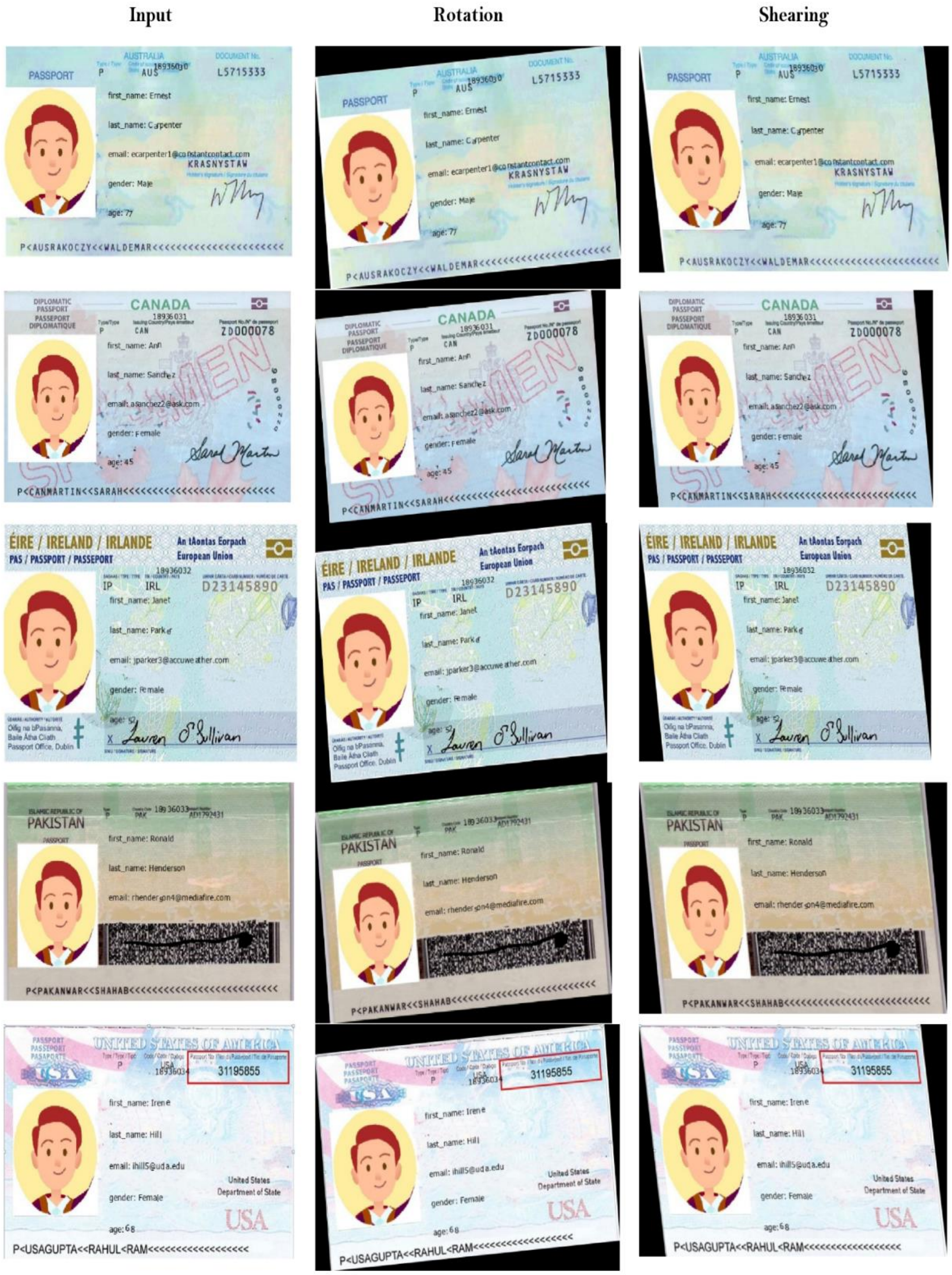


Figure 6. Passport with applied data augmentation Right to left, input, rotation augmentation and shearing augmentation.

2.4. Save forged document with bounding box

In this step we describe the motivation of recording the bounding box of tempered character, as unlabeled data can be found online in large volume, but to get the data labelled is extremely difficult, time consuming, tedious and expensive task [12]. To mitigate that issue, our proposed algorithm records the bounding boxes of the tampered characters and saves them in the json files so that the research community can use it for training networks for detection. Above four step process is described for one document. To synthesise more data, we repeat this process to generate 3000 documents for each country document, finally we save forged document images with their bounding boxes in a json format file.

3. Libraries / Package Used

We used multiple libraries to forge the character in the passport and driving licences. Libraries/packages including PIL [13] library to perform operation of image reading, setting and drawing font with its position, numpy [14] to deal rotation and other mathematical operations, random [15] to perform randomness for position and random noise adding, os [16] library to deal with file listing and path handling, json [17] library to deal with json file as annotation and cv [18] library to deal augmentation.

4. Conclusions

This paper addresses the gap of forged character detection of documents i.e. passport and driving licence, due to datasets unavailability. To fill this gap, this paper presents a new algorithm of synthesising data considering real world scienarios and releases two new datasets named forged characters detection on passport (FCD-P) and forged characters detection on driving licence (FCD-D), using five different countries' passport and licence. This research work opens new challenges for forged character detection on passport and driving licence. Finally, we release our code and datasets and the research community can use it for their research purpose. Possible future work is to include more countries' passports and driving licences and apply state-of-the-art detection algorithms.

Acknowledgements

This publication has emanated from research [conducted with the financial support of/supported in part by a grant from Science Foundation Ireland under Grant number 18/CRT/6223 and is supported by the ADAPT Centre for Digital Content Technology which is funded under the SFI Research Centres Programme (Grant 13/RC/2106/_P2), Lero SFI Centre for Software (Grant 13/RC/2094/_P2) and is co-funded under the European Regional Development Fund. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

[1] Fake identity brits warned that their lives are in danger, Online Available:https://www.independent.co.uk/news/world/middle-east/fake-identity-brits-warned-that-their-lives-are-in-danger-1905971.html . Accessed on:

[2] Wu, L., Zhang, C., Liu, J., Han, J., Liu, J., Ding, E., & Bai, X. (2019, October). Editing text in the wild. In *Proceedings of the 27th ACM international conference on multimedia* (pp. 1500-1508).

[3] Yang, Q., Huang, J., & Lin, W. (2020). Swaptext: Image based texts transfer in scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 14700-14709).

[4] Roy, P., Bhattacharya, S., Ghosh, S., & Pal, U. (2020). STEFANN: scene text editor using font adaptive neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 13228-13237).

[5] Korshunov, P., & Marcel, S. (2018). Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685.*

[6] Zhao, L., Chen, C., & Huang, J. (2021). Deep Learning-based Forgery Attack on Document Images. *arXiv preprint arXiv:2102.00653*.

[7] Adadi, A. (2021). A survey on data-efficient algorithms in big data era. *Journal of Big Data*, *8*(1), 1-54.

[8] Bertrand, R., Gomez-Krämer, P., Terrades, O. R., Franco, P., & Ogier, J. M. (2013, August). A system based on intrinsic features for fraudulent document detection. In *2013 12th International conference on document analysis and recognition* (pp. 106-110). IEEE.

[9] Shang, S., Kong, X., & You, X. (2015). Document forgery detection using distortion mutation of geometric parameters in characters. *Journal of Electronic Imaging*, *24*(2), 023008.

[10] Ryan, M., & Hanafiah, N. (2015). An examination of character recognition on ID card using template matching approach. *Procedia Computer Science*, *59*, 520-529.

[11] https://www.kaggle.com/avkash/5feature30kcsv/version/1 (accessed on 1/17/2022)

[12] Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, *3*(1), 1-130.

[13] https://pillow.readthedocs.io/en/stable/ (accessed on 1/17/2022)

[14] https://numpy.org/ (accessed on 1/17/2022)

[15] https://docs.python.org/3/library/random.html (accessed on 1/17/2022)

[16] https://docs.python.org/3/library/os.html (accessed on 1/17/2022)

[17] https://docs.python.org/3/library/json.html (accessed on 1/17/2022)

[18] https://pypi.org/project/opencv-python/ (accessed on 1/17/2022)

AUTHORS

Teerath kumar received his Bachelor's degree in Computer Science with distinction from National University of Computer and Emerging Science (NUCES), Islamabad, Pakistan, in 2018. Currently, he is pursuing PhD from Dublin City University, Ireland. His research interests include advanced data augmentation, deep learning for medical imaging, generative adversarial networks and semi-supervised learning.

Muhammad Turab is an undergraduate final year student at Computer Systems Engineering MUET, Jamshoro. He has done 60+ projects with java and python, all projects can be found on GitHub. His research interests include deep learning, computer vision and data augmentation for medical imaging.

Shahnawaz Talpur is the chairman of Computer Systems Engineering Department at Muet Jamshoro. He has done his masters from MUET and PhD from Beijing Institute of Technology, China. His research interests include high performance computing, computer architecture and big data.

R. Brennan is an Assistant Professor in the School of Computing, Dublin City University, Chair of the DCU MA in Data Protection and Privacy Law and a Funded investigator in the Science Foundation Ireland ADAPT Centre for Digital Content Technology which is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund, His main research interests are data protection, data value, data quality, data privacy, data/AI governance and semantics.

M. Bendechache is an Assistant Professor in the School of Computing at Dublin City University, Ireland. She obtained her Ph.D. degree from University College Dublin, Ireland in 2018. Malika's research interests span the areas of Big data Analytics, Machine Learning, Data Governance, Cloud Computing, Blockchain, Security, and Privacy. She is an academic member and a Funded Investigator of ADAPT and Lero research centres.

ADVANCED SERVICE DATA PROVISIONING IN RoF-BASED MOBILE BACKHAULS/FRONTHAULS

Mikhail E. Belkin, Leonid Zhukov and Alexander S. Sigov

Russian Technological University MIREA, Moscow, Russia

ABSTRACT

A new cost-efficient concept to realize a real-time monitoring of quality-of-service metrics and other service data in 5G and beyond access network using a separate return channel based on a vertical cavity surface emitting laser in the optical injection locked mode that simultaneously operates as an optical transmitter and as a resonant cavity enhanced photodetector, is proposed and discussed. The feasibility and efficiency of the proposed approach are confirmed by a proof-of-concept experiment when optically transceiving high-speed digital signal with multi-position quadrature amplitude modulation of a radio-frequency carrier.

KEYWORDS

5G and beyond, access network, RoF-based mobile fronthaul/backhaul, real-time monitoring, QoS metrics, OIL-VCSEL.

1. INTRODUCTION

In recent years, with the maturing of 5G NR systems, the design of access networks (ANs) has acquired some significant changes. In particular, the centralized radio access network (C-RAN) that passed from the 4G-LTE network, where a fronthaul interface connects various small cells deployed as remote radio heads (RRHs) to a centralized macro-cell deployed as a baseband unit (BBU) [1, 2], was standardized [3] into the next generation RAN (NG-RAN). Following it, some newer functional blocks, such as a central unit (CU), a distributed unit (DU), and a radio unit (RU) were introduced that is considered in detail in [4]. The key reason for this transformation was the use of the Common Public Radio Interface (CPRI) with time division multiplexing (TDM) between the BBU and RRH [4]. This approach led to data transfer rates up to hundreds of Gbps, which makes this interface economically unjustified for 5G and beyond.

Figure 1 depicts a typical architecture of NG-RAN, where CU, DU, and a set of RUs are duplex connected via fiber-optics communication lines (FOCLs), while RUs and mobile user terminals (UTs) intercommunicate wirelessly.

Generally, as optical networks evolve to fulfil highly flexible connectivity and dynamicity requirements, and supporting ultra-low latency services, it is increasingly important that a NG-RAN also provides reliable connectivity and improved network resource efficiency. Collection of different types of data from various sources is necessary for applying automation techniques to network management. However, the network must also support the capability to extract knowledge and form perception for performance monitoring, troubleshooting, and maintains network service continuity over a wide range of elements at various levels. Such scalability and

David C. Wyld et al. (Eds): AISCA, NET, DNLP - 2022
pp. 55-63, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120205

flexibility are particularly important for the wide area network, in particular, for streaming telemetry [5]. Moreover, an efficient optical performance monitoring (OPM) design should consider different scenarios including large-scale disaster [6], when a prompt reaction is needed but limited bandwidth is available now.

Projecting the above problem that is common for telecom networks, for the purpose of this paper, we can conclude that in the newer generation, ANs providing the function of low-cost real-time monitoring and the quality-of-service (QoS) metrics are a matter of great importance from the point of view of their maintenance. In this case, the issue can be solved by additionally introducing a special node into the DU circuit (see Fig. 1), which is responsible for the accumulation and processing of monitoring results. However, a more promising solution from the point of view of reducing the total cost and latency, in our opinion, might be the introduction of an additional function from the existing indispensable element of its circuitry, return transmission of the optical signal to the CU, and its processing there. A promising technique for implementing this approach through the simultaneous use of an optically injection-locked vertical cavity surface emitting laser (OIL-VCSEL) as a laser source and a resonant cavity enhanced photodetector (RCE-PD) was proposed in [7] for a bi-directional optical communication and recently developed by us [8] referred to microwave photonics circuits.

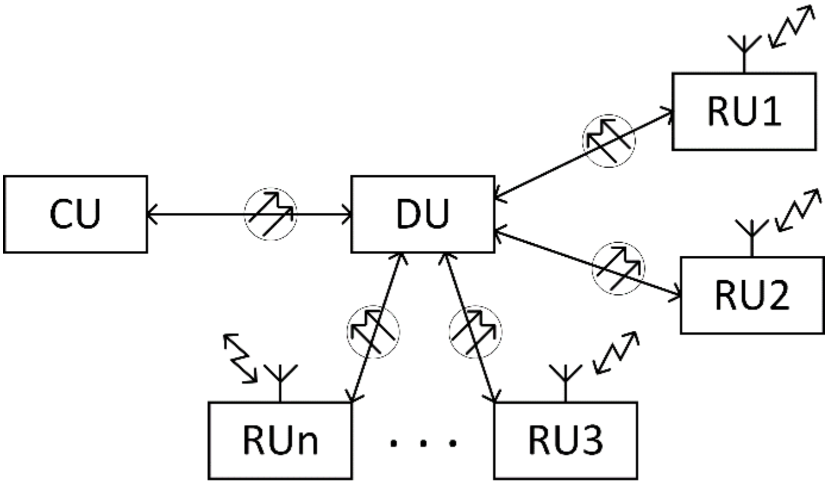


Figure 1. Conceptual block-diagram of 5G's NG-RAN

Elaborating the approach, in this paper after reviewing the modern service provisioning techniques in ongoing optical networks in Section 2, a newer design concept of OIL-VCSEL-based transmitter/receiver, which receives information data from CU via the downlink channel and simultaneously re-transmits them to RUs via the uplink channel for processing them at CU, is proposed and discussed in Section 3. The feasibility and efficiency of the proposed solution are confirmed in Section 4 by a proof-of-concept experiment, when optically transmitting a high-speed digital signal with 64-position quadrature amplitude modulation (QAM) of a 5-GHz radio frequency (RF) carrier. Section 5 concludes the paper.

2. Modern Tendencies to Service Provisioning in Ongoing NG-RANs

With the development of techniques and technologies of 5G NR networks, it became clear that this is not just a new standard for mobile communications. In general, the widespread worldwide introduction of core and access networks of the 5th and subsequent generations in the long term should transform our worldview and lead to a social transformation of the world community, radically changing the principles of communication, architecture, economy and the level of service of local and global telecom networks. For this purpose, known and newer for cellular communications concepts, paradigms, approaches, scenarios, technologies, mechanisms, tools and services are being developed. In particular, they include noted in Introduction NG-RANs, Radio-over-Fiber (RoF)-based mobile backhauls/fronthauls, as well as Enhanced Mobile Broadband (eMMB) [9], Ultra-Reliable Low Latency Communication (URLLC) [10], Massive Machine-Type Communications (mMTC) [11], Internet of Everything [12], Slice-based Networks for Heterogeneous Environments [13], Software Defined Networking (SDN) [14], Network Function Virtualization (NFV) [15] and so on.

The results of the above innovations should lead to significant improvements in the QoS and key parameters of NG RANs. Thus, the recommendation 3GPP TR 38.913 identified the following outstanding key indicators of new generation networks:

- downlink peak data rate up to 20 Gbps with spectral efficiency 30 bps/Hz
- uplink peak data rate up to 10 Gbps with spectral efficiency 15 bps/Hz
- the minimum delay in the radio access subsystem for URLLC services is 0.5 ms, for eMBB services - 4 ms
- the maximum density of the IoT devices connected to the network in urban territories is 1'000'000 devices/sq. km
- autonomous operation of the IoT devices without recharging the battery for 10 years;
- vehicle mobility at a maximum speed of 500 km/h.

Along with these, a critical problem arose related to ensuring the above parameters during the maintenance of realistic NG-RANs through the development of advanced operational management's principles, approaches and schemes. In general, advanced monitoring framework of optical networks aimed at the continuous, remote, automatic and cost-effective supervision of the physical layer has to satisfy the basic set of the requirements:

(i) Fast and accurate detection of the performance degradation and service disruptions
(ii) Accurate tracking down location of the network failure
(iii) Monitoring should be non-intrusive and not affecting normal network operation
(iv) Compatible with various types of optical networks.

Besides, to meet the goals of 5G NG and beyond, network infrastructures should facilitate a high level of flexibility and automation. In particular, monitoring and data analytics give rise to estimate accurately the QoS of new light paths, to anticipate capacity exhaustion and degradations, or to predict and localize failures, among others to facilitate this automation [16]. At the same time, network operations and management (OAM) increasingly relies on the ability to stream and process in real-time data from network equipment. An integral part of the OAM is to make sure whether the operational conditions are normal or not and intervene, if needed, by quickly recovering and mitigating the occurred problems [17]. The goal to have network management automation and abstraction of open line systems (OLS) could be possible by the software defined network (SDN) technologies, which requires accurate OPM data from the elements of the network [18].

3. DESIGN CONCEPT

Based on the results of the review in Section 2, it can be unambiguously concluded that any existing approach to monitoring the QoS of a FOCLs leads to the complication of the DU circuitry and operation, and consequently to the increase in its cost and the latency of signal transmission via the AN. Therefore, the solution related to the introduction of an additional function to the existing indispensable element of its circuitry, namely the simultaneous use of an OIL-VCSEL as a laser source and a RCE-PD, is promising in principle.

Revealing the proposed concept, Figure 2 demonstrates a block diagram of a communication channel, containing the all three functional units of NG-RAN (see Figure 1). It is worth noting that the block-diagram presented in this Figure has three key distinguishing features in according to NG-RAN concept. First, at the CU, in order to simplify the circuitry of subsequent units, the conversion of the modulation format of the optical carrier from a baseband (BB) to QAM of RF subcarrier is performed so that the digital signal is then transmitted over FOCLs using a RF equal to the allocated frequencies of the corresponding RU [19]. Secondly, a duplex optical channel is introduced between the CU and DU, where the information signal is transmitted in the downlink direction, and the QoS data signal - in the uplink direction. Thirdly, in the DU, an OIL-VCSEL is connected through an optical circulator, where reflected optical output is transmitted via downlink to the RU, and the detected RF signal with added QoS data is again converted into the optical range using a standard low-cost optical transmitter and returned to the CU for the subsequent processing.

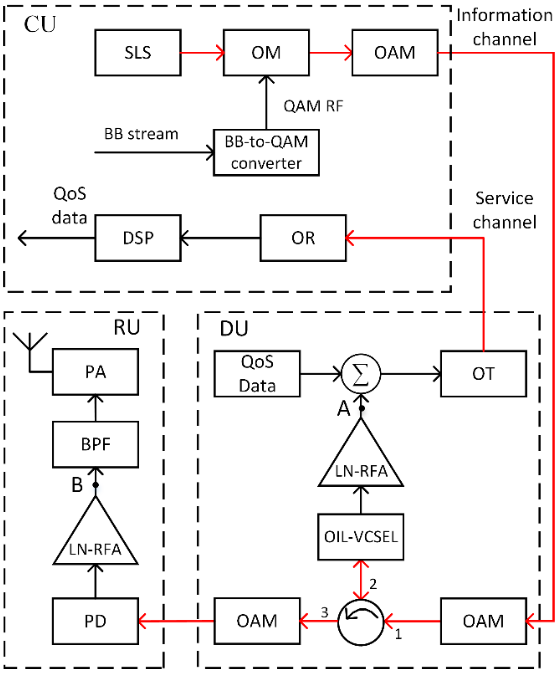


Figure 2. Block-diagram of the proposed duplex communication channel for NG-RAN, where SLS, OM, OAM, DSP, OR, OT, PD, LN-RFA, BPF, and PA stand for semiconductor laser source, optical modulator, optical amplifier, digital signal processor, optical receiver, optical transmitter, photodetector, low-noise RF amplifier, bandpass RF filter, and power RF amplifier (Optical connections are painted in red, electrical connections – in black)

4. Proof-of-Concept Experiment

To validate the proposed concept a laboratory experiment was realized where a wafer-fused long-wavelength vertical cavity surface emitting laser of RTI Research, LLC in the form of a chip, optically injection locked by a master laser, was used. All measurements were carried out on the Probe Station EP6 from Cascade Microtech using coplanar RF probe and fiber-optics probe. A detailed description of the research object is given in [8]. The purpose of the experiment is to confirm the functionality and effectiveness of the block-diagram of a duplex communication channel for NG-RAN described in Section 3, when optically transmitting a high-speed digital signal with multi-position QAM of a RF carrier. Note that during the experiment the OIL-VCSEL operates in forward DC biased mode without switching to reverse DC bias in photodetector mode as required with a standard pin-photodiode, which is ensured by optical injection locking [8].

Figure 3 shows the testbed for proof-of-concept experiment, the layout of which is based on Figure 2 with the exclusion of non-essential for the confirmation elements after points A and B. The testbed contains pin-photodiode (Finisar, BPDV2150: 43-GHz bandwidth, 0.6-A/W responsivity), a pair of low-noise RF amplifiers (Mini-circuits ZX60-542LN-S+: 4.4-5.4-GHz frequency band, 24-dB gain, 1.9-dB noise figure), and two coils of single-mode fibers SMF-28+, as well as measuring tools including Vector Signal Generator (Keysight MXG N5182B) and 4-channel Mixed Signal Oscilloscope (Keysight Infinium MSOS804A).

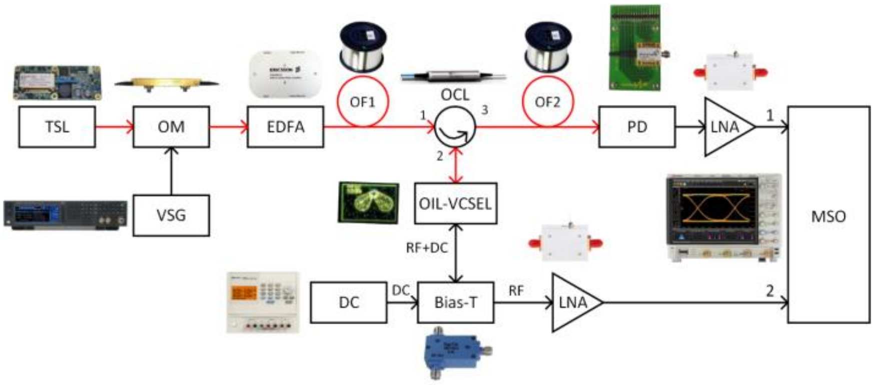
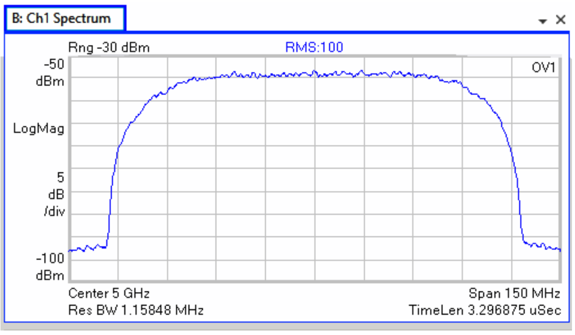
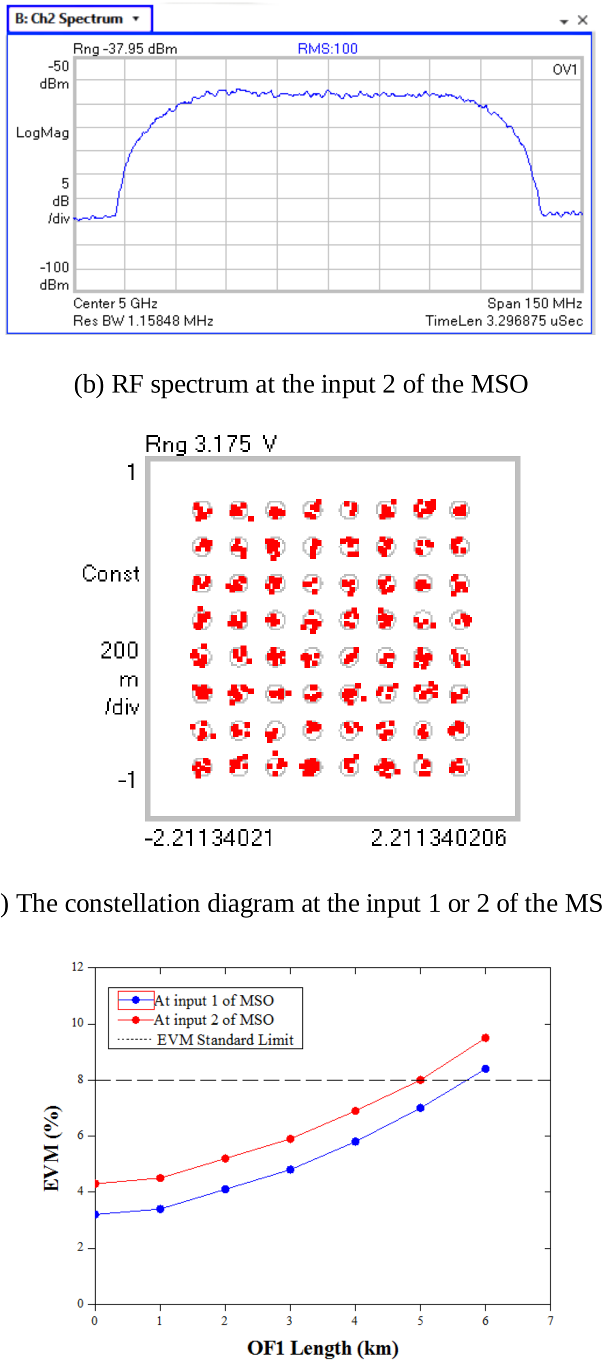


Figure 3. Testbed for the proof-of-concept experiment where TSL, OM, EDFA, OCL, OF, PD, LNA, DC, VSG, and MSO stand for tunable semiconductor laser (master laser), optical modulator, erbium-doped fiber amplifier, optical circulator, optical fiber, photodetector, low-noise amplifier, DC source, vector signal generator, and mixed signal oscilloscope, respectively. (Optical connections are painted in red, electrical connections – in black)

Figure 4 presents the results of the experiment, where optical carrier of the frequency near 192.2 THz is intensity modulated by the 560 Mbps 64-QAM RF signal of 5 GHz. Namely, in Figures 4 (a, b), MSO RF spectra are shown at the inputs 1 and 2, correspondingly. Figure 4 (c) shows the constellation diagram at the input 1 or 2 of the MSO, and Figure 4 (d) shows EVM values vs OF1 length at the inputs 1 and 2 of the MSO.



(a) RF spectrum at the input 1 of the MSO



(b) RF spectrum at the input 2 of the MSO

(c) The constellation diagram at the input 1 or 2 of the MSO

(d) EVMs vs OF1 length at the inputs 1 and 2 of the MSO

Figure 4. The results of the proof-of-concept experiment

The outcome that can be drawn from the above results is the following. In the DU under test, the values of error vector magnitude in back-to-back configuration are 3.2% for downlink channel to RU and 4.3% for uplink channel to CU, which is significantly less compared to the standard EVM threshold for 64-QAM of 8% [20]. This maximum acceptable transmission quality is achieved, when the distance between CU and DU is increased up to 5 km, which is quite realistic in a practical 5G access network based on small cell concept.

5. CONCLUSIONS

In this paper, a new cost-efficient concept to realize real-time monitoring of quality-of-service metrics and other service data in 5G and beyond access networks using a small-cell architecture is proposed and discussed. The essence of the proposed solution is investigated in detail on the example of a block diagram of a duplex communication channel, including a microcell base station and a functionally split picocell base station containing a central unit, a distribution unit, and a set of remote radio units each connected via optical fiber link. A distinctive feature of the proposed approach, which provides an improvement in the power and cost characteristics, is to use in the distribution unit, a vertical cavity surface emitting laser in the optical injection locked mode that simultaneously operating as an optical transmitter and as a resonant cavity enhanced photodetector. Thanks to this, it becomes possible to transmit information data to distribution unit via the downlink channel and real-time service data related to the status, quality of service, etc. via the uplink channel for processing them at the central unit. The feasibility and efficiency of the proposed solution are confirmed by a proof-of-concept experiment when optically transmitting a high-speed digital signal with 64-position quadrature amplitude modulation of a 5-GHz radio-frequency carrier, which is widely exploited in access networks of fifth-generation cellular communication systems based on Radio-over-Fiber technology and small cell architecture scenario.

The further research in this direction will focus on a detailed studying the VCSEL-based optical transmission with real-time monitoring function for ongoing 5G and beyond access networks of millimeter-wave band. The fundamental feasibility of this path due to optical injection locking has already been considered in a number of scientific publications, for example, in [21] and confirmed experimentally [22] using the same VCSEL chip as in this study.

REFERENCES

[1] G. Pandey, A. Choudhary, and A. Dixit, “Radio over Fiber Based Fronthaul for Next-Generation 5G Networks,” Proceedings of 22nd International Conference on Transparent Optical Networks (ICTON), Bari, Italy, July 19-23, p. 1-4, 2020.

[2] M. Kamalian-Kopae, M. E. Belkin, and S. K. Turitsyn, “The Design Principles of Fibre-Wireless Integration in the Incoming Mobile Communication Networks,” Proceedings of 22nd International Conference on Transparent Optical Networks (ICTON 2020), Bari, Italy, July 19-23, p. 1-4, 2020

[3] “Study on new radio access technology: radio access architecture and interfaces,” 3GPP Technical Report TR38.801, v14.0.0, 2017.

[4] J. Zou, S. A. Sasu, M. Lawin, A. Dochhan, J.-P. Elbers, and M. Eiselt, “Advanced optical access technologies for next-generation (5G) mobile networks,” Journal of Optical Communications and Networking, Vol. 12, No. 10, p. D86-D98, October 2020.

[5] A. Sadasivarao, et al. "High performance streaming telemetry in optical transport networks." Optical Fiber Communication Conference. Optical Society of America, 2018

[6] S. Xu, et al. "Emergency OPM recreation and telemetry for disaster recovery in optical networks," Journal of Lightwave Technology, vol. 38, No 9, p. 2656-2668, May 2020.

[7] Q. Gu, W. Hofmann, M.-C. Amann, L. Chrostovski, “Optically Injection-Locked VCSEL for Bi-directional Optical Communication,” 2008 Conference on Lasers and Electro-Optics (CLEO2008), p. 1-2, 2008.

[8] M. E. Belkin, L. I. Zhukov, D. A. Fofanov, M. G. Vasil’ev, and A. S. Sigov. “Studying a LW-VCSEL-Based Resonant Cavity Enhanced Photodetector and its Application in Microwave Photonics Circuits,” Chapter in IntechOpen book “ Light-Emitting Diodes and Photodetectors - Advances and Future Directions”, 2021, 20 pp. https://www.intechopen.com/online-first/studying-a-lw-vcsel-based-resonant-cavity-enhanced-photodetector-and-its-application-in-microwave-ph

[9] Li-Sheng Chen, et al. “AMC With a BP-ANN Scheme for 5G Enhanced Mobile Broadband,” IEEE Access, Sept. 2020, DOI: 10.1109/ACCESS.2020.3024726.

[10] B. Singh, O. Tirkkonen, Z. Li and M. A. Uusitalo, "Interference Coordination in Ultra-Reliable and Low Latency Communication Networks," 2018 European Conference on Networks and Communications (EuCNC), 2018, pp. 1-255, doi: 10.1109/EuCNC.2018.8443229.

[11] S. Chae, S. Cho, S. Kim and M. Rim, "Coded random access with multiple coverage classes for massive machine type communication," 2016 International Conference on Information and Communication Technology Convergence (ICTC), 2016, pp. 882-886, doi: 10.1109/ICTC.2016.7763321.

[12] S. Higginbotham, "Network included - [Internet of Everything]," in IEEE Spectrum, vol. 57, no. 11, pp. 22-23, Nov. 2020, doi: 10.1109/MSPEC.2020.9262153.

[13] Peng Xu, Xiaqi Liu, Z. Sheng, Xuan Shan and Kai Shuang, "SSDS-MC: Slice-based Secure Data Storage in Multi-Cloud Environment," 2015 11th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE), 2015, pp. 304-309.

[14] T. Theodorou and L. Mamatas, "CORAL-SDN: A software-defined networking solution for the Internet of Things," 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2017, pp. 1-2, doi: 10.1109/NFV-SDN.2017.8169870

[15] S. Bijwe, F. Machida, S. Ishida and S. Koizumi, "End-to-End reliability assurance of service chain embedding for network function virtualization," 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2017, pp. 1-4, doi: 10.1109/NFV-SDN.2017.8169853

[16] A. Sgambelluri, et al. "Exploiting Telemetry in Multi-Layer Networks," 22nd International Conference on Transparent Optical Networks (ICTON), 2020.

[17] V.-M. Alevizaki, et al. "Joint fronthaul optimization and SDN controller placement in dynamic 5G networks." International IFIP Conference on Optical Network Design and Modeling. Springer, Cham, 2019.

[18] W.-K. Jia, et al. "A Survey on All-Optical IP Convergence Optical Transport Networks," 2019 7th International Conference on Information, Communication and Networks (ICICN). IEEE, 2019

[19] M. E. Belkin, T. Bakhvalova, and A.C. Sigov, “Design Principles of 5G NR RoF-Based Fiber-Wireless Access Network”; In book “Recent Trends in Communication Networks”, IntechOpen, London, UK, p. 121-145, 2020. DOI: 10.5772/intechopen.90074.

[20] ETSI, "Minimum requirements for Error Vector Magnitude," in TECHNICAL SPECIFICATION, LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception (3GPP TS 36.101 version 14.3.0 Release 14), ETSI, 2017-04, p. 215

[21] Erwin K. Lau, Liang Jie Wong, Ming C. Wu, “Enhanced Modulation Characteristics of Optical Injection-Locked Lasers: A Tutorial,” IEEE Journal of Selected Topics in Quantum Electronics, v. 15, no. 3, p. 618-633, 2009.

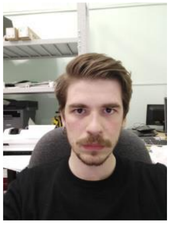
[22] Mikhail E. Belkin, Leonid Zhukov, “OIL-VCSEL-Based Microwave-Photonics Transceiver for a Millimeter-Wave Fronthaul,” Accepted at the IEEE 2nd International Conference on Signal, Control and Communication SCC 2021. December 20 -22, 2021, Tunis, Tunisia (unpublished)

AUTHORS

Prof. Dr. Mikhail E. Belkin - received an engineering degree in radio and television from Moscow Institute of Telecommunications, in 1971, Ph. D. degree in telecommunication and electronic engineering from Moscow Technical University of Telecommunications and Informatics, in 1996, and Sc. D. degree in photonics and optical communications from Moscow State Technical University of Radio-Engineering, Electronics and Automation, in 2007. The theme of his Sc. D. degree thesis is ‘Analog Fiber Optic Systems with multiplexing on RF and Microwave Subcarriers and Multiservice HFC Networks on their Base’. He has written more than 250 scientific works in English and Russian. The major current R&D fields are fiber-optic devices and systems, microwave photonics, photonic ICs, incoming cellular communication networks, computer-aided design.

At present, M. E. Belkin is the Director of the Scientific and Technological Center “Integrated Microwave

Photonics", Professor of the department "Optical and Optoelectronic Devices and Systems", Institute of Physics and Technology, MIREA - Russian Technological University, and he is a member of IEEE's MTTS, LEOS (now PhS), and COMSOC from 2006, and a member of OSA from 2018.



Leonid Zhukov graduated from Moscow technical university of communications and informatics in 2019, and works at Russian Technological University MIREA 2016 as a laboratory assistant researcher after an engineer at the STC 'Integrated Microwave Photonics'. He is a co-author of 1 monographs published by IntechOpen. His main research interests are microwave-photonics, and telecommunication technologies.

Professor Dr. Alexander. S. Sigov is an expert in Solid State Physics and Electronics. He contributed extensively to the phenomenology of magnets, ferroelectrics and multiferroics, physics of ferroic-based heterostructures, thin films, etc. The results of his scientific activity are reflected in more than 300 papers, reviews, book chapters, 19 monographs and textbooks, including the well-known "Defects and Structural Phase Transitions" together with A. Levanyuk. For many years, he has chaired the Department of Nanoelectronics in MIREA. He created his own school, inspiring and mentoring many talented scientists. In 2006, he was elected a Member of the Russian Academy of Sciences. He is the head of the Russian Academy Council on Dielectrics and Ferroelectrics, member of numerous scientific societies, Associate Editor of international journals Ferroelectrics and Integrated Ferroelectrics, Editor and member of Boards of more than ten Russian national journals, Chair of the Council on Physics and Astronomy of the Russian Foundation for Basic Research. At present, Alexander Sigov is the Head of Nanoelectronics Dept. and President of MIREA – Russian Technological University, Moscow, Russia, Doctor of Physics, and Fellow member of the Russian Academy of Sciences.

A NEW APPROACH FOR SPEECH KEYWORD SPOTTING IN NOISY ENVIRONMENT

Peiwen Ye and Hancong Duan

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, China

ABSTRACT

Keyword Spotting works to detect wake-up keywords in a continuous voice stream, which is widely used in products such as mobile devices and smart home. Recently, DNNs dominate keyword spotting and dramatically improve performance. However, few researchers concerned about noise in speech keyword recognition. Thus, we propose an architecture for the detection under noisy scenario. Our framework combines attention mechanism and residual structure based on the CNN backbone. In addition, we use separable convolution to reduce the number of model's parameters, which makes it applicable in the embedded devices. Noises from various scenes are utilized for data augmentation to boost performance. The proposed method achieves an accuracy of 94.93% on the noisy test set based on the Google Speech Commands dataset. We also compare performance between the proposed method and RNN-based algorithm, and prove our model achieve higher accuracy with fewer parameters.

KEYWORDS

Keyword Spotting, Noise robustness, Command Recognition, Attention Mechanism, Data augmentation.]

1. INTRODUCTION

Deep learning algorithms have entered a challenging new phase. In various cognitive tasks, including image classification [1] and speech recognition [2], the accuracy has surpassed humans. More and more artificial intelligent products appeared in our daily life, and people's productivity has been greatly improved. Among them, voice control further liberates people's hands and eyes, familiar voice products on the market include smart speakers such as Baidu Xiaodu, Xiaomi Xiaoai, Ali Tmall Genie, and voice assistants such as Apple Siri and Microsoft Cortana [3]. The devices wait for the wake-up word and activates the voice interaction program accordingly when the wake-up word occurs.

The release of the Google Speech Commands dataset provides a common basis for KWS system evaluation and it is allowed for scientific research [4]. With the publishing of the dataset, Warden also provided a baseline model based on the convolution architecture [5], with an accuracy rate of 85.4%. Hidden Markov Models (HMM), a traditional keyword recognition method, which model keywords and backgrounds together [6][7][8]. Recently, more significant architectures are applied to KWS tasks. Deep neural networks (DNN), starting with the fully-connected networks, have been shown to perform efficiently [9]. A major disadvantage of DNN-based KWS algorithms is that they cannot effectively model local temporal and spectral correlations in speech features. CNN overcomes this shortcoming by treating the input time-domain and spectral-domain features as images and perform two-dimensional convolution operations on them [10][11]. Recent studies have shown that RNN-based keyword recognition using LSTM cells can exploit longer temporal contexts with gating mechanisms and internal states [12][13][14].

David C. Wyld et al. (Eds): AISCA, NET, DNLP - 2022
pp. 65-76, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120206

Most of current KWS researches are based on clean data, and ignore the diversity of usage scenarios. We directly test their model on a noisy test set and found that the performance was not satisfactory. At the same time, it also needs to meet the requirements of low memory and low power consumption. We summarize our contribution as follows:

- We train with a large amount of noisy data to simulate a wide variety of noisy environments, to improve the noise robustness of the model.
- To further improve the accuracy of the model, inspired by natural language process, we propose an architecture, combining attention module with residual mechanism, and we use depth separable convolution replace the ordinary convolution to reduce the resource occupation.
- We also apply attention mechanism to RNN and CNN backbones with different parameter size to compare the performance.

2. Related Work

2.1. Keyword Spotting

Keyword spotting is used to detect the preset words from a continuous speech, and is usually deployed in devices that require low power consumption and small memory, such as smart speakers and smart watches. Considering about the function, in addition to detecting target words, the classifier also needs to distinguish between silence and words or sounds that are not in the target list. The pipeline is as Figure 1.

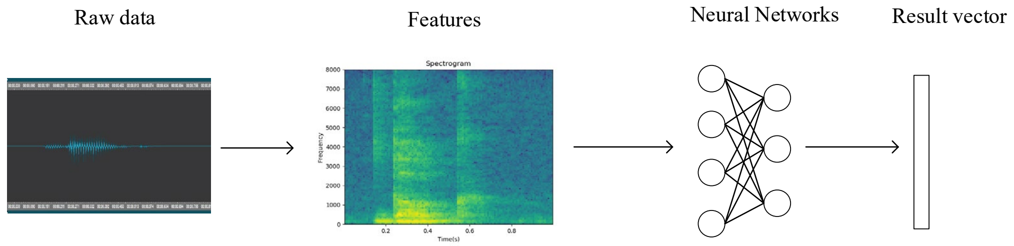


Figure 1. Keyword spotting overview

Zhang et al. compared and analyzed several network structures [15] for quiet speech command recognition. Among them, DNNs consist of a stack of fully connected layers and nonlinear activation layers. Each fully connected layer is followed by a ReLU activation functions. And then output is probabilities of the k keywords generated by a linear layer and a softmax layer. DNN has less operations/inference and hence suit well to systems that have limited computing capability, but the accuracy is not very impressive. RNN has shown excellent performance in many sequence tasks, especially speech recognition, language modeling and translation [16], etc. RNNs not only exploit the temporal relationship between input signals, but also utilize a “gating” mechanism to capture long-term dependencies. RNN cells can be LSTM cells or Gated Recurrent Unit (GRU) cells [17][18]. Since weights are reused across all T time steps, RNNs tend to have a smaller number of parameters compared to DNNs. However, since RNNs are time-series dependent, it cannot be computed in parallel. CNNs treat the input temporal and spectral domain features as images and performs two-dimensional convolution operations on them. Convolutional layers are usually followed by batch normalization [19], ReLU activation functions, and sometimes max or average pooling layers for the purpose of reducing the dimensionality of features. During inference, the batch-normalized parameters can be collapsed into the weights of the convolutional layers. To reduce parameters and speed up training, a linear low-rank layer is added between the convolution and dense layers. For better performance, CRNN, combination of CNN and RNN, exploits local temporal/spatial correlation using convolution layers and obtain global temporal dependencies in the speech features with recurrent layers [20]. Most of the later studies are based on the above-mentioned architectures.

2.2. Attention mechanism

The target of the attention mechanism is to selectively filter a small amount of important information from a large amount of information, focus on key information with limited attention. It ignores most unimportant information to save resources, and obtains the most effective information efficiently and quickly [21]. Attention was firstly introduced to deal with the problem that the frame generated by encoder-decoder is too long and information lost in the seq2seq task [22]. As shown in Figure 2, the results obtained from cells are concatenated and fed to the attention layer. In simple terms, attention gives weights to words according to their contribution.

Since the encoder encode input data of any length into a fixed-length vector, when inputting a very long data, it will cause information loss. Attention solves this problem by encoding the input data of any length into a vector sequence instead of a vector. The attention module can not only be applied to the encoder-decoder model, but is a general idea and does not depend on a specific framework. In the field of speech recognition, the main contribution of the attention mechanism is to align the output characters with the input speech signal.

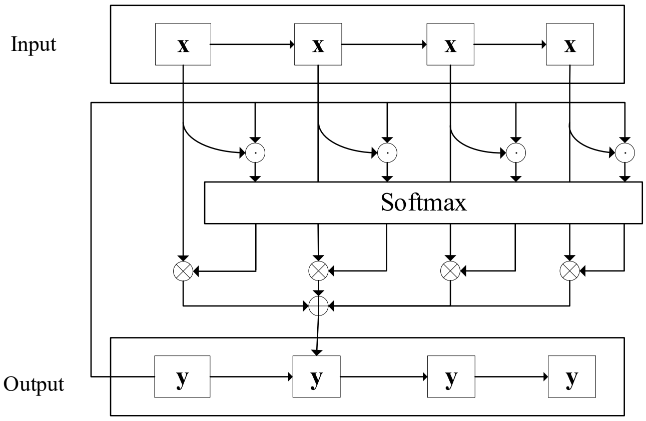


Figure 2. Attention mechanism in encoder-decoder architecture

3. METHOD

Based on CNN, we propose an architecture that incorporates attention mechanism and residual mechanism on CNN backbone.

3.1. Backbone

Most of machine learning is shallow model (such as GMM, HMM)[23][24][25], which means weak nonlinear transformation ability, so it is not enough to describe the complex high-dimensional features of speech. Since the input of GMM is a single frame, the influence of co-pronunciation is ignored, so we use the spliced frame as the input of the neural network to model the observation probability. Both traditional HMM and RNN can model time series. Since HMM is a shallow model and RNN has the problem of gradient disappearance, a long short-term memory network (Long short-term memory, LSTM) is used to model time series [26]. Long short-term memory, a special kind of RNN, mainly used to solve the problem of gradient disappearance and gradient explosion when input a long sequence. Simply put, compared to ordinary RNNs, LSTM can perform better in longer sequences.

Under normal circumstances, speech recognition is based on the speech spectrum after time-frequency analysis, and the speech time-frequency spectrum has structural characteristics. With regard to improve the speech recognition performance, it is necessary to overcome the various diversity faced by the speech signal, including the diversity of the speakers and the diversity of the environment, and so on. A convolutional neural network carries out translation-invariant convolution in time and space respectively. Thanks to the idea of applying convolutional neural network to the acoustic model of speech recognition, the invariance of convolution can be used to surmount the diversity of the speech signal. From this perspective, it can be considered that the time-frequency spectrum obtained by the analysis of the entire speech signal is treated as an image, and the deep convolutional network widely used in the image is introduced to recognize it. The earliest widely used neural network DNN is a standard feedforward neural network composed of some fully connected layers and nonlinear activation layers. The input of DNN is a flattened feature matrix, and the network structure contains a stack of d hidden layers, and each layer has n neurons. One of the main shortcomings of the DNN-based KWS algorithm is that it cannot effectively simulate the local time and spectral correlations in speech features. CNN makes use of this correlation of the time domain and spectral domain to process the features into an image, and perform a two-dimensional convolution operation on it. The convolutional layer is usually followed by batch normalization, ReLU activation functions and optional max/average pooling layers [27], which reduce the dimensionality of features. In the inference process, the batch normalized parameters can be folded into the weight of the convolutional layer. In some cases, in order to reduce parameters and speed up training, a linear low-rank layer is added between the convolutional layer and the dense layer. It is just a fully connected layer without nonlinear activation. We use LSTM and CNN as the backbone networks to study speech keyword recognition in noisy environments.

3.2. Attention Mechanism

The attention mechanism identifies key features in the data by using the weights of the new attention layer. Attention module includes spatial attention and channel attention. The spatial attention shows attention to different positions on the feature map, similarly, the channel attention shows attention to different data channels. For speech data, that means, the attention to time domain and frequency domain respectively.

Channel attention module passes the input feature map through global max pooling and global average pooling based on width and height respectively, and then Multilayer Perceptron (MLP). The outputs of the MLP are concatenated based on element-wise, and then the sigmoid activation operation is performed to generate the final channel attention feature map. Channel attention module makes use of the elementwise multiplication of the channel attention feature map and the input feature map to generate the input features required by the spatial attention module. The feature map is compressed in the spatial dimension to obtain a one-dimensional vector for next operate. When compressing in the spatial dimension, average pooling and max pooling are both considered. Average pooling and max pooling can be used to aggregate spatial information of feature maps, which will be fed to a shared network to compress the spatial dimension of input feature maps. Finally, element-wise summation is performed to produce a channel attention map.

The spatial attention module uses the feature map generated by channel attention module. The algorithm firstly carries out a channel-based global max pooling and global average pooling, and then concatenates the results based on the channel. Next, it reduces the dimension to 1 by a convolution operation. By using a sigmoid activation function, the spatial attention feature is generated. Finally, we obtain the final generated feature by multiplying the input feature of the module to the spatial attention feature. The spatial attention mechanism compresses the channel, and performs average pooling and max pooling respectively in the channel dimension. The operation of max pooling is to extract the maximum value on the channel, and the number is the result of multiplying the height by the width; the operation of average pooling is to extract the average value on the channel, and the number is the same as the previous one. Finally, the feature maps of 1-channel are merged and obtain a 2-channel feature map.

The spatial attention ignores the information in the channel domain, and treats the image features in each channel equally. This limits the spatial transformation method only be used in the original image feature extraction stage, and it is not very interpretable if it is applied to other layers of the neural network. Similarly, the channel attention directly uses global average pooling in a channel, ignoring the local information in each channel.

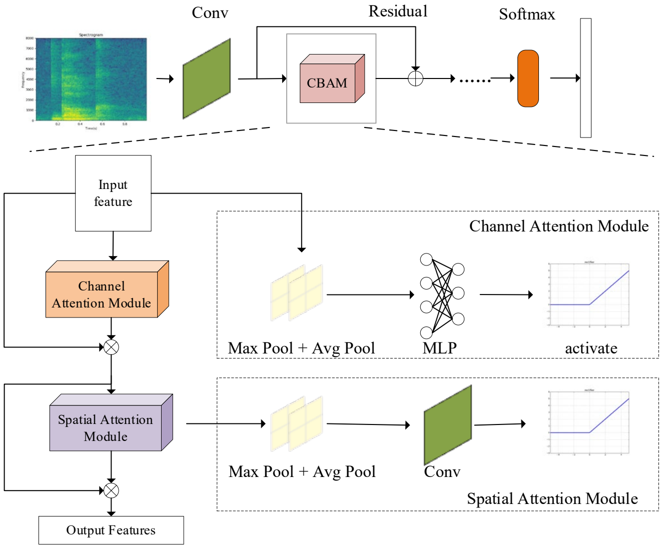


Figure 3. Model architecture with attention mechanism

For the above two kinds of attention, the spatial attention ignores the information in the channel domain, and treats the image features in each channel equally. This limits the spatial transformation method to the original image feature extraction stage, and it is not very interpretable if it is applied to other layers of the neural network. Similarly, the channel attention directly uses global average pooling in a channel, ignoring the local information in each channel. The model architecture with attention mechanism is in Figure 3. Combining channel attention and spatial attention can not only save parameters and computing power, but also perform better.
we combine above two attention mechanism and residual module. Residual structure was firstly proposed by He et al. to eliminate the problem of difficulty in training and accuracy degradation with excessive depth networks [31]. The proposal of residual network is a milestone event in the history of CNN images. At ILSVRC and COCO 2015, Deep residual network (ResNet) achieved 5 firsts, and once again refreshed the history of the CNN model on ImageNet. So far, it has become a classic structure in the field of computer vision. The residual mechanism concatenates past feature and transformed features to maintain information, considering the similarity between speech keyword recognition and image recognition, we applied the residual method to our attention module, we concatenate the input features and output feature of the attention module as

the input for the next operation. Therefore, not only the mask is added according to the information of the current network layer, but also the information of the previous layer is passed down. This can effectively prevent the problem of information loss due to the deepening of network layers.

3.3. Depthwise Separable Convolution

For the purpose of further reducing the power consumption of the neural network and adapt to embedded wearable devices, separable convolution is used instead of ordinary convolution, mainly using depthwise separable convolution (DSC), which is the first to appear in a doctoral dissertation called "Rigid-motion scattering for image classification" [28]. DSC is composed of Depthwise Convolution and Pointwise Convolution. Each channel of the Depthwise Convolution input feature map uses a convolution kernel, and then the outputs of all convolution kernels are spliced to get its final output, as shown in Figure 4. Because the quantity of output channels of the convolution is equal to the number of convolution kernels, and only one convolution kernel is used for each channel in Depthwise Separable Convolution, the quantity of output channels of a single channel after the convolution operation is also 1. Then if the channels' number of the input feature map is N, a single convolution kernel is used for each of the N channels to obtain N feature maps with 1 channel. Then, the N feature maps are spliced in order to obtain an output feature map with channel N. This operation enormously cuts down the number of parameters of the convolution kernel. The depthwise separable convolution module is shown in Figure 4.

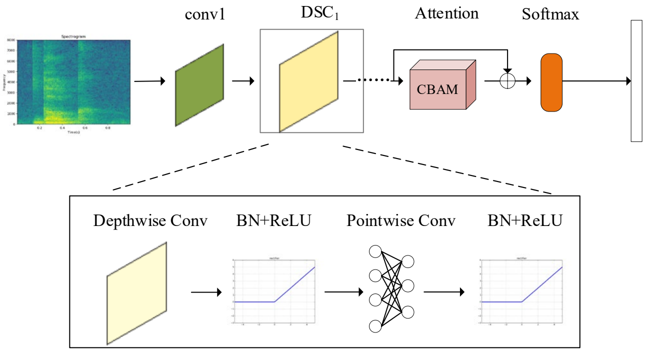


Figure 4. Depthwise separable convolution

4. EXPERIMENT

The audio signal is pre-processed into a discrete matrix vector by the acoustic feature-extract module, which is used as the input feature of our keyword detection model. We extract 40-dimensional feature vectors within a 25-milliseconds frame, with a frame offset of 10 milliseconds. The neural network model is used to process the input MFCC features. After the classification model, a fixed-length vector, which has the same size with the categories, is generated, and the output value is exactly the probability of the data belonging to each category. The pipeline of the entire model is in the Figure 1.

4.1. Datasets and Input Pre-processing

We used authoritative Google Speech Commands dataset as the benchmark dataset. Google Speech Commands dataset version 1 has 65K utterances from different speakers, each of which is 1 second long. Each utterance belongs to one of 30 categories, including common words such as "Go", "Stop", "Left", and "Down". The dataset's sampling rate is 16kHz. In our experiment, we considered to recognize 10 words and additional tags "unknown" and "silent". Besides, we also try to recognize all 35 words. For the both tasks and each architecture, we measured the top-1 classification accuracy.

We use feature extraction module to obtain MFCC features, which will be fed to the neural network [29]. In addition to the traditional signal processing steps such as pre-enhancement, framing, windowing, Fourier transformation, and Mel filtering, some other data enhancement methods are introduced, including randomly shifting the voice signal on the time axis, randomly enhancing the power spectrum, and appropriately doing some time masking and frequency masking [30]. Time domain masking means randomly selecting 0-50 consecutive frames, and set all their mel-filter groups value to 0, and the frequency domain mask similarly is to randomly select 1 to 30 continuous dimensions from 40 mel-filter groups, and set their values to 0, as shown in Figure 5. For each mini-batch, 1/3 only has time domain masking, 1/3 only has frequency domain masking, and the others have all masking. These data augmentation means can enrich the diversity of data effectively.

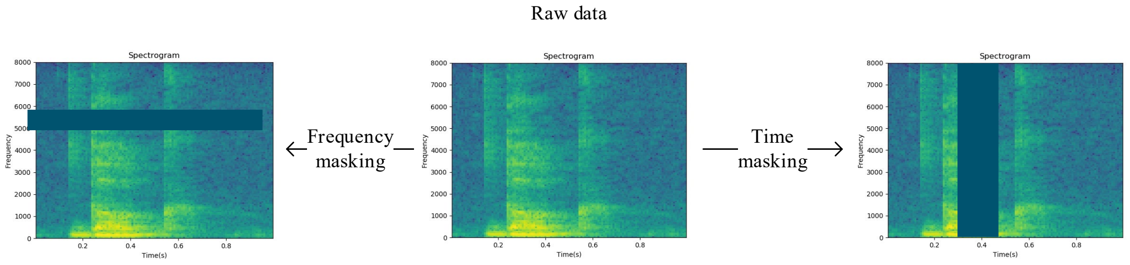


Figure 5. Data augmentation

In addition to the data augmentation measures described above, a variety of noisy data in various scenes with different signal-to-noise ratio is added to the training data to enhance the performance of the model in a noisy environment. Noise comes from scenes including kitchen, living, washing, field, park, river, hallway, meeting, office, resto, station, squares, traffic, car and metro and so on. Each category contains 16 pieces of noise data, and the length of each noise wav is 300s [31]. The format of noise data is also mono 16KHz. By detecting recognition accuracy on noisy data, the results proved that the noise mixed training method has greatly advantages.

4.2. Model Training

We used the tensorflow framework to train the model. The hyperparameters used are shown in the Table 1.For Data processing, we set the time window length as 25ms, and time window stride as 10ms. Extract MFCC features using 40 mel filters.

Table 1. Hyperparameters used in our experiment

Data processing	
Time window length	25ms
Time window stride	10ms
Regularization	
Wight decay	10^{-5}
Dropout	0

training	
Batch size	100
Weight initialization	Xavier
Optimizer	Adam
Training steps	3000
Schedule	exponential
Learning rate	0.001

For training, the mini batch size is 100. We use the Adam optimizer to train with an exponential decay of the learning rate for all architecture, and the learning rate decays by 0.1 after every 10,000 steps start from 0.001. The model ends training after two attenuations. The cross-entropy loss function is used to measure the loss between labels and predicted values. We also use dropout and regularization to prevent overfitting, and the L2 weight decay is 10-5.

4.3. Results

We add additional synthetic keyword samples for each keyword category, including clean and low SNR, based on 65K Google Speech Command Dataset utterances. The noise comes from airport, babble, car, exhibition, restaurant, street, subway and train. For each category, there are 900~1500 new utterances, both for clean and noisy. We shuffle the original datasets and our additional samples to generate training, validating and testing data randomly.

The increased number of samples is shown in Table 2. The "original" means sample size of Google, and the "total" is the amount of the original data plus the added data, including clean and noisy, and the "ratio of noise" means the ratio of the added noisy data to the total.

Table 2. Data size in our experiments

Original(K)	Total(K)	Ratio of noisy data(%)
65	147	27.9

We test the performance of depthwise separable convolution. Since this method replaces the convolution operation of the CNN model, we directly add the attention module based on the CNN model for comparison. The CNN model contains two convolutional layers, one linear layer and two fully connected layers. The structure is shown in Table 3.

Table 3. Architecture of CNN_attention

Input	Layer	c	f	s
B×49×10×1	Conv	28	(10,4)	(1,1)
B×40×7×28	Conv	30	(10,4)	(2,1)
B×16×4×30	flatten	-	-	-
B×1920	Linear	-	-	-
B×16	Fc	-	-	-
B×128	Fc	-	-	-

Most scholars like to test models in 12 categories with additional silence and non-crucial words when using Google Speech Command dataset. Therefore, we use LSTM and CNN as the backbone network to test the recognition accuracy of the model on the noisy test data for comparison. At the same time, in order to test the performance of the model on a larger data set, the experiment also involved all 35 classes of the dataset. The following Table 4 displays the results of the model for 12 classes.

Table 4. Noisy testing dataset accuracy of 12-classes.

Backbone	**V1**	**V2**	**V3**
LSTM	94.03%	94.58%	94.72%
CNN	94.48%	94.66%	94.93%

Among them, v1 means testing the original model in noisy test data, as a comparison, v2 mixed training data with noise based on v1, v3 introduces the attention module based on v2. The following Table 5 shows the performance of the model for 35 classes.

Table 5. Noisy testing dataset accuracy of 35-classes.

Backbone	**V1**	**V2**	**V3**
LSTM	92.60%	92.64%	93.12%
CNN	93.49%	93.56%	93.79%

V1~v3 are the same meaning as the above Table 4. It can be seen from the above two tables that the model performs well on both datasets. The method of mixing noise with training data promotes. And the attention module also performs well.

We take a small CNN as an example, Table 3 display the input for each layer, and "c" is channel for convolution, "f" represents the size of filter, s means the stride for x and y respectively. The total parameter size is about 69.9KB. We test three different scales of CNN to compare accuracy and memory footprint. The results are as Table 6.

Table 6. Performance of different convolution

Network	**Params(K)**	**Accuracy(%)**
CNN+attention	69.9	91.72
	179.7	92.08
	504.3	93.16
+DSC	24.4	94.48
	144.1	94.66
	498.3	94.93

Our model greatly reduces the amounts of parameters through separable convolution. We obtain better performance than normal CNN with much smaller parameter size. Separable convolution effectively reduces the number of parameters by decomposing convolution kernel and drops the correlation of convolution operations between channels, and has little impact on the calculation results.

5. CONCLUSIONS

This article introduces the application of LSTM and CNN in the speech keyword recognition system. Our works are achieved with the latest release of Google Speech Command dataset, which provides a general benchmark for this task. In past studies, most scholars published their results on clean test data. However, in fact we are surrounded by a variety of noisy environments. Our research shows that adding attention module to the model is conducive to the task of keyword spotting. At the same time, according to the noise test results, it is necessary to add additive noise with different signal-to-noise ratios in different scenarios to the training data for joint training.

We also considered the lightweight method of separable convolution. In theory, depthwise separable convolution requires less computation. However, since the computational intensity of deep convolution (the ratio of FLOPs to memory access) is too low, it is difficult to effectively use hardware, so it is difficult to effectively implement deep separable convolution in practice. Therefore, in the future, I will focus on effective ways to reduce the computing consumption of the model, meantime explore the model structure to further improve the recognition ability of the model.

REFERENCES

[1] Chen Y, Li J, Xiao H, et al. Dual path networks [J]. arXiv preprint arXiv:1707.01629, 2017.

[2] Xiong W, Wu L, Alleva F, et al. The Microsoft 2017 conversational speech recognition system[C]//2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2018: 5934-5938.

[3] McGraw I, Prabhavalkar R, Alvarez R, et al. Personalized speech recognition on mobile devices[C]//2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016: 5955-5959.

[4] Warden P. Speech commands: A dataset for limited-vocabulary speech recognition [J]. arXiv preprint arXiv:1804.03209, 2018.

[5] Chen G, Parada C, Heigold G. Small-footprint keyword spotting using deep neural networks[C]//2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014: 4087-4091.

[6] Richard C Rose and Douglas B Paul, "A hidden markov model based keyword recognition system" in Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on. IEEE, 1990, pp. 129–132.

[7] Rohlicek J R. Continuous HMM for Speaker Independent Word Spotthing[J]. Proc. ICASSP, May. 1994, 1994.

[8] Wilpon J G, Miller L G, Modi P. Improvements and applications for key word recognition using hidden Markov modeling techniques[C]//[Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing. IEEE, 1991: 309-312.

[9] Guoguo Chen, Carolina Parada, and Georg Heigold, "Small-footprint keyword spotting using deep neural networks," in Acoustics, speech and signal processing (icassp), 2014 ieee international conference on. IEEE, 2014, pp. 4087–4091

[10] Sainath T, Parada C. Convolutional neural networks for small-footprint keyword spotting[J]. 2015.

[11] Chen X, Yin S, Song D, et al. Small-footprint keyword spotting with graph convolutional network[C]//2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2019: 539-546.

[12] Wollmer M, Eyben F, Keshet J, et al. Robust discriminative keyword spotting for emotionally colored spontaneous speech using bidirectional LSTM networks[C]//2009 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2009: 3949-3952.

[13] Ming Sun, Anirudh Raju, George Tucker, Sankaran Panchapagesan, Gengshen Fu, Arindam Mandal, Spyros Matsoukas, Nikko Strom, and Shiv Vitaladevuni, "Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting," in Spoken Language Technology Workshop (SLT), 2016 IEEE.IEEE, 2016, pp. 474–480.

[14] Chai S, Zhang W Q, Lv C, et al. An End-to-End Model Based on Multiple Neural Networks with Data Augmentation for Keyword Spotting [J]. International Journal of Asian Language Processing, 2020, 30(02): 2050006.

[15] Zhang Y, Suda N, Lai L, et al. Hello edge: Keyword spotting on microcontrollers [J]. arXiv preprint arXiv:1711.07128, 2017.

[16] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Advances in neural information processing systems, pages 3104–3112, 2014.

[17] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. Learning precise timing with lstm recurrent networks. Journal of machine learning research, 3(Aug):115–143, 2002.

[18] Jozefowicz R, Zaremba W, Sutskever I. An empirical exploration of recurrent network architectures[C]//International conference on machine learning. PMLR, 2015: 2342-2350.

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning, pages 448–456, 2015.

[20] Sercan O Arik, Markus Kliegl, Rewon Child, Joel Hestness, Andrew Gibiansky, Chris Fougner, Ryan Prenger, and Adam Coates. Convolutional recurrent neural networks for small-footprint keyword spotting. arXiv preprint arXiv:1703.05390, 2017.

[21] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.

[22] Shi B, Yang M, Wang X, et al. Aster: An attentional scene text recognizer with flexible rectification [J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 41(9): 2035-2048.

[23] Liu Y, He L, Zhang W Q, et al. Investigation of Frame Alignments for GMM-based Digit-prompted Speaker Verification[C]//2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, 2018: 1467-1472.

[24] Zimmermann M, Ghazi M M, Ekenel H K, et al. Visual speech recognition using PCA networks and LSTMs in a tandem GMM-HMM system[C]//Asian conference on computer vision. Springer, Cham, 2016: 264-276.

[25] Nankaku Y, Sumiya K, Yoshimura T, et al. Neural Sequence-to-Sequence Speech Synthesis Using a Hidden Semi-Markov Model Based Structured Attention Mechanism [J]. arXiv preprint arXiv:2108.13985, 2021.

[26] Wilkinson N, Niesler T. A Hybrid CNN-BiLSTM Voice Activity Detector[C]//ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021: 6803-6807.

[27] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778..

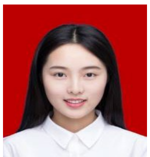
[28] Sifre L, Mallat P S. Rigid-Motion Scattering For Image Classification Author[J]. English. Supervisor: Prof. Stéphane Mallat. Ph. D. Thesis. Ecole Polytechnique, 2014.

[29] Huizen R R, Kurniati F T. Feature extraction with mel scale separation method on noise audio recordings[J]. arXiv preprint arXiv:2112.14930, 2021.

[30] Park D S, Chan W, Zhang Y, et al. Specaugment: A simple data augmentation method for automatic speech recognition[J]. arXiv preprint arXiv:1904.08779, 2019.

[31] Büchel J, Zendrikov D, Solinas S, et al. Supervised training of spiking neural networks for robust deployment on mixed-signal neuromorphic processors[J]. arXiv preprint arXiv:2102.06408, 2021.

AUTHORS



Peiwen Ye Received the B.S. degree in computer science (2015) from Southwest University. Now she is seeking a master's degree in computer science at the University of Electronic Science and Technology of China, her research direction is deep learning and speech recognition.

Hancong Duan received the B.S. degree in computer science from Southwest Jiaotong University in 1995, the M.E. degree in computer architecture in 2005, and the Ph.D. degree in computer system architecture from UESTC in 2007. Currently he is a professor of computer science at UESTC. His current research interests include Deep Learning, Large-Scale P2P Content Delivery Network, Distributed Storage and Cloud Computing.

DISCOVERY OF ASSOCIATION RULES OF THE RELATIONSHIP BETWEEN FOOD CONSUMPTION AND LIFE STYLE DISEASES FROM SWISS NUTRITION’S (MENUCH) DATASET & MULTIPLE SWISS HEALTH DATASETS FROM 1992 TO 2012

Timo Lustenberger[1], Helena Jenzer[2] and Farshideh Einsele[1]

[1]Section of Business Information,
Bern University of Applied Sciences, Switzerland
[2]Hospital of Psychiatry, University of Zurich, Switzerland

ABSTRACT

This article demonstrates that using data mining methods such as Weighted Association Rule Mining (WARM) on an integrated Swiss database derived from a Swiss national dietary survey (menuCH) and 25 years of Swiss demographical and health data is a powerful way to determine whether a specific population subgroup is at particular risk for developing a lifestyle disease based on its food consumption patterns. The objective of the study was to discover critical food consumption patterns linked with lifestyle diseases known to be strongly tied with food consumption. Food consumption databases from a Swiss national survey menuCH were gathered along with data of large surveys of demographics and health data collected over 25 years from Swiss population conducted by Swiss Federal Office of Public Health (FOPH). These databases were integrated and reported in a previous study as a single integrated database. A data mining method such as WARM was applied to this integrated database. A set of promising rules and their corresponding interpretation was generated. As an example, the found rules of the sample show that the consumption of alcohol in small quantities does not have a negative impact on health, whereas the consumption of vegetables is important for the supply of vitamins of the B group, which help the energy metabolism to pro-vide energy. These vitamins are particularly lacking in alcoholics and should then be taken with supplements. Another finding is that dietary supplements do little specially by diabetes. Applying WARM algorithm was beneficial for this study since no interesting rules were pruned out early and the significance of the rules could be highly increased as compared to a previous study using pure Apriori Algorithm.

KEYWORDS

Data Mining, WARM Association Analysis, Diet & Chronical Diseases, Health Informatics

1. INTRODUCTION

Lifestyle diseases are diseases that increase in frequency as countries become more industrialized and people get more aged. Lifestyle diseases include obesity, hypertension (blood pressure), heart disease, type 2 diabetes, cancer, mental disorders, and many others. They differ from the

David C. Wyld et al. (Eds): AISCA, NET, DNLP, BDHI - 2022
pp. 77-93, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120207

infectious diseases originated from malnutrition, also called communicable diseases (CD) due to their contagious, dispersive nature. Lifestyle diseases are therefore among the so-called NC (non-communicable diseases) diseases. According to World Health Organization (WHO), the growing epidemic of chronic diseases afflicting both developed and developing countries are related to dietary and lifestyle changes [1].

Several researchers studied the relationship be-tween nutritional habits and lifestyle diseases aka chronic diseases. A. Fardet and Y. Boirie have agregated 304 pooled/meta-analyses and systematic reviews to obtain a qualitative overview of the associations between 17 food and beverage groups and the risk of diet-related chronic disease. The review of these authors confirmed that plant food groups were more protective than animal food groups against diet-related chronic diseases. Their results show that overweight, obesity, type 2 diabetes, cancer, and cardiovascular diseases accounted for 289 of the pooled/meta-analyses and systematic reviews [2]. Further, S. Fardet et al. conducted additional pooled analyses and meta-analyses of cohort studies and randomized controlled trials that linked fruit consumption with the risk of chronic disease and metabolic deregulation. Their results show that the degree of processing influences the health effects of fruit-based products. Fresh and dried fruits appeared to have a neutral or protective effect on health, 100% fruit juices had intermediary effects, and high con-sumption of canned fruit and sweetened fruit juice was positively associated with the risk of all-cause mortality and type 2 diabetes, respectively [3]. S. Schneider and al. conducted a mini–Nutritional Assessment as a promising score for evaluating malnutrition in the elderly, since nutrition intervention shortens the length of stay by diminishing the rate of complication and to identify malnourished patients and those who are at nutritional risk to treat and prevent malnutrition by chronic diseases by elderly [4].

Machine Learning and Data Mining methodologies for chronic diseases prediction and prevention in relationship with nutritional habits have been explored by different researchers Internationally. S. Lee et al conducted a study using stepwise logistic regression (SLR) analysis, decision tree, random forest, and support vector machine as an alternative and complement to the traditional statistical approaches to identify the factors that affect the health-related quality of life (HRQoL) of the elder-ly with chronic diseases and to subsequently develop from such factors a prediction model [5]. D. Qudsi and al. report in [6] from a study that aims to identify the potential benefits that data mining can bring to the health sector, using Indonesian Health Insurance company data as case study. Decision tree as a classification data mining method, was used to generate the prediction model by visualizing the tree to perform predictive analysis of chronic diseases. Z. Lei et al report in [7] of studying the relationship between nutritional ingredients and diseases such as diabetes, hypertension, and heart disease by using data mining methods. They have identified the first two or three nutritional ingredients in food that can benefit the rehabilitation of those diseases. R. McCabe et al. report in [8] of creating a simulation test environment using characteristic models of physician decision strategies and simulated populations of patients with type 2 diabetes, they state of employing a specific data mining technology that predicts encounter-specific errors of omission in representative databases of simulated physician-patient encounters and test the predictive technology in an administrative database of real physician-patient encounter data. D.W. Haslam and W.P. James re-port in [9] of an investigation in a population - based sample of 1140 children performed to derive dietary patterns related to children's obesity status. Their findings reveal that Rules derived through a data mining approach revealed the detrimental influence of the increased consumption of fried food, delicatessen meat, sweets, junk food and soft drinks. K. Lange et al. state in [10] that big data studies may ultimately lead to personalized genotype-based nutrition which could permit the prevention of diet-related diseases and improve medical therapy. A. Hearty and M. Gibney evaluate the usability of supervised data mining methods as ANNs and decision trees to predict an aspect of dietary quality an aspect of dietary quality based on dietary intake with a food-based coding system and a novel meal-based coding system [11]. A. von Reusten et al. used data from 23 531 participants of the EPIC-Potsdam study to analyze the associations between 45 single food groups and risk of major chronic diseases, namely, cardiovascular diseases (CVD), type 2 diabetes and cancer using multivariable-adjusted Cox regression. Their results show that higher intakes of low-fat dairy, butter, red meat, and sauce were associated with higher risks of chronic diseases [12]. E. Yu et al. demonstrate in [13] the usability of supervised data mining methods to extract the food groups related to bladder cancer. Their results show that beverages (non-milk); grains and grain products; vegetables and vegetable products; fats, oils and their products; meats and meat products were associated with bladder cancer risk.

To gain understanding about the impact of using data mining techniques for the analysis of lifestyle diseases that can be influenced by nutrition, we conducted a preliminary study on this matter [14]. In this preliminary previous study, we used a big database gained from a grocery store chain over a certain period along with associated health data of the same region. Association rule mining was successfully used to describe and predict rules linking food consumption patterns with lifestyle diseases. Additionally, we conducted a further study using real world health and nutritional data from Swiss population and gained interesting rules which showed the link between nutritional habits and chronical diseases [15]. In the current study, we use the same national Swiss dietary survey (menuCH) with a five times larger dataset (collected over 25 years) from the national Swiss health survey including demographical information. Based on the finding of the previous study [15], where it used the pure Apriori algorithm which resulted that some critical health-related dietary features were pruned out early in course of data mining, we have applied the Weighted Association Mining Rules (WARM) analysis to gain more accurate association rules that show the link between Swiss nutritional habits and chronical diseases.

2. DATABASE SELECTION

The data comes from the national surveys menuCH and the health survey that were carried out in Switzerland.

The national food survey menuCH [16] was carried out for the first time from January 2014 to February 2015. Over 2000 people living in Switzerland were asked about their eating habits and food consumption. The data resulting from the survey is the first representative, national nutritional survey data available in Switzerland from BLV.

The second data source comprises health data on the state of health and health-related behavior of the Swiss resident population over a period of 25 years. The Federal Statistical Office has been collecting health data from the population living in Switzerland every five years using a written and telephone questionnaire [17]. As part of this study, representative data from around 85,000 people from 1992, 1997, 2002, 2007 and 2012 are available. This data has already been pre-cleaned, attributes have been partially selected from the database and the data has been already transformed as reported in [18].

3. DATA PREPARATION FOR DATA MINING PURPOSES

In addition to the previous database reported in [18], a further reduction of the data was carried out due to the objective of the study. All data sets that could not be assigned to one of the four subject areas examined (alcohol, blood pressure, cholesterol, and diabetes) were deleted. This enabled the data volume to be massively reduced and the performance of operations with the MySQL database to be increased. A total of around 10 million responses from people to individual questions were available as a data set. A MySQL database was used to ensure the integrated collection of data from different sources on a permanent basis. The MySQL database server is very fast, reliable, and easy to use.

4. DATA TRANSFORMATION

Categories were taken over from our previous study [15] on blood pressure, cholesterol, diabetes, and alcohol consumption. Blood pressure was reduced into 6 categories. The cholesterol data was reduced to 4 categories. The diabetes data was reduced to 4 categories and finally alcohol consumption data was reduced to 4 categories. As an example, the alcohol consumption data was reduced as follows:

- Daily alcohol consumption up to 18 grams,
- Daily alcohol consumption > 18-23 grams,
- Daily alcohol consumption> 23-28 grams,
- Daily alcohol consumption> 28 grams

5. CREATION OF INTEGRATED, RELATIONAL DATABASES

The new reduced health database with 25 years data but 4 selected categories was then integrated with already existing menuCH database from our previous study [15] using five common demographical attributes available in both databases were used, such as gender, age group, household, marital status, and language to link the two databases into an integrated relational database. Figure 1 shows the revised scheme of the integrated database of health and nutrition data. The big advantage of the new structure is that the database can easily be expanded with additional topics for future investigations without having to adapt the database schema. This is very efficient and timesaving if the survey catalogue of the national health survey is expanded over the next few years and additional lifestyle topics are covered and analysed.

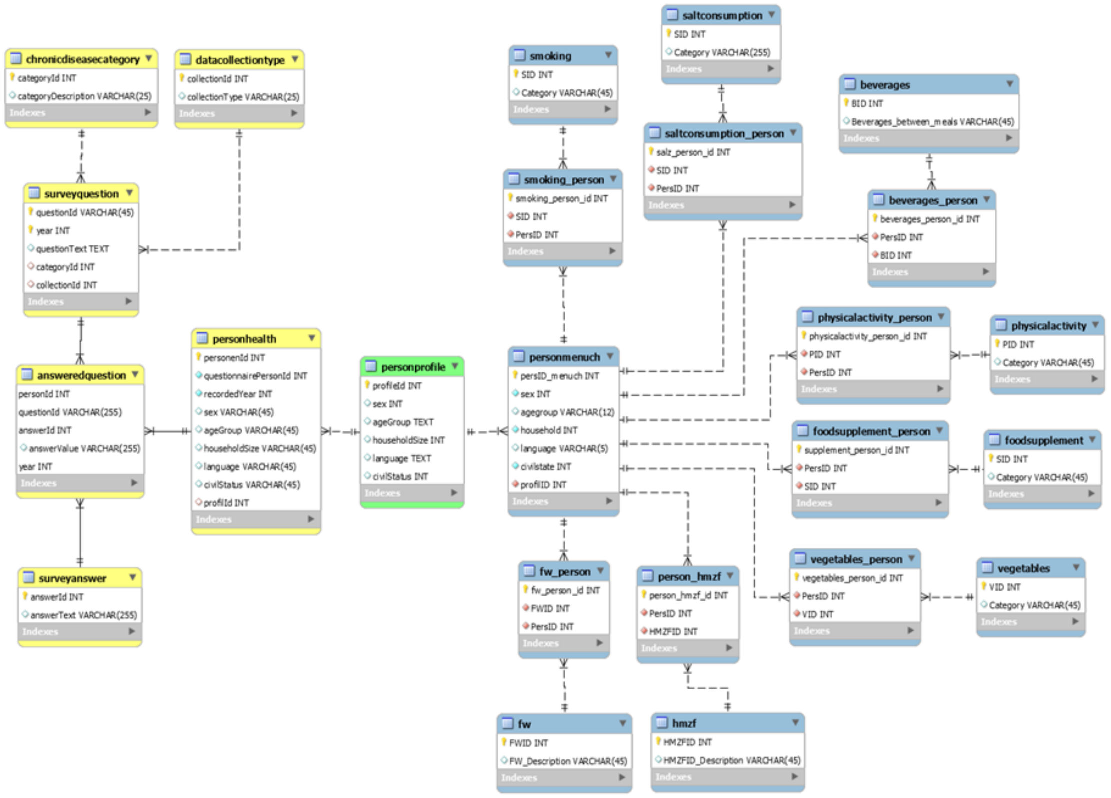


Figure 1. Scheme of integrated Database

6. ASSOCIATION ANALYSIS USING APRIORI ALGORITHM

The most common method for association analysis is the Apriori algorithm [19], for which there exists several variants. In comparison, the Frequent Pattern (FP)-Growth algorithm was considered, which also examines the frequent occurrence of things together [20]. The two algorithms were contrasted and compared for the investigation using the criteria of complexity, performance, memory consumption, accuracy, iterations, and weighting. The algorithm was selected based on the relevant criteria for the task and considers improvement suggestions from the study for the survey year Occurred in 2012 [15]. Mewes et al. conclude in their study that *the weighting of features should be considered so that rules with higher technical relevance would not prun out too early in the data mining process.* Due to its simplicity and its sufficient efficiency and based on the findings of Mewes et al. in [15], we chose *Apriori algorithm in combination with the weighted association analysis (WARM)* over FR-Growth. In the weighted association analysis, the weight *w_*i assigned to each item i reflects the relative im-portance of one item to other items. The weight of an item set is derived from the individual weights for the items contained in a rule according to [21]. The aim of the association analysis is to find rules of the form "if feature A occurs, then feature B occurs with the probability of the confidence level" (A-> B). The calculation parameters support, confidence and lift were used to evaluate the rules. The algorithm continues until no item set fulfils the mini-mum support (Agrawal and Srikant, 1994). Item sets for rule formation were selected from theses 9 items. The item sets with the highest support and confidence value were selected for rule formation. In this study Apriori algorithm was applied to find rules for a set of 9 items as follows: 8 items from menuCH database and 1 item was the categorized chronical diseases from Swiss health database as described previously (see sec. categorization of health and menuCH data). Our multidisciplinary team consists of a dietician expert who carried out the assignment of weights of the 8 items in the menuCH table for application of weighted Apriori algorithm as previously described in this section.

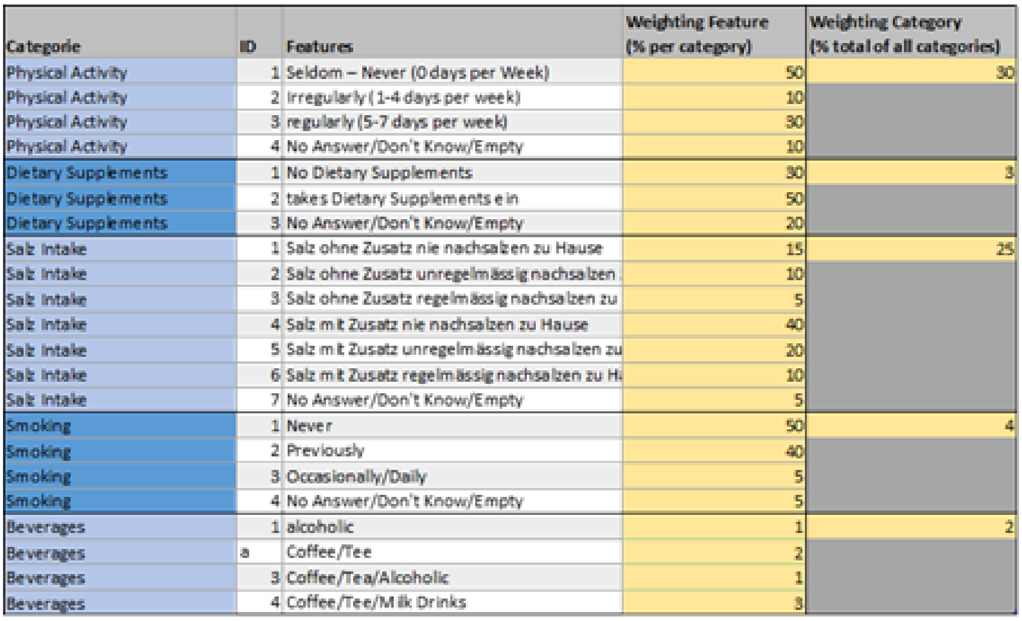
6.1. Exemplary stepwise implementation of the Apriori algorithm with WARM for Blood Pressure

The association analysis with the Apriori algorithm in combination with WARM was carried out step by step. The pre-processed and transformed data of the integrated database of health and nutritional data form the basis for the analysis process. The total amount of data samples is calculated as total amount of interviewees multiplied by the number of asked questions for each investigated chronical disease in this study.

6.1.1. Step1: Weighting of the Characteristics

As a first step in the weighted association analysis, a specific weight was assigned to each characteristic as an item. Due to the different effects of nutritional characteristics on the chronic diseases examined, a different weighting was carried out for each subject area. The weighting was carried out in two stages, in which a weight was defined as the first stage for each characteristic within a category. The second stage is made up of all categories, which in turn have different meanings for the corresponding lifestyle disease. The following eight categories from the national nutrition survey were used for the association analysis and have positive or negative effects on chronic diseases.: Number of main meals, Activity, Consumption of vegetables, Beverages, Food supplements, Smoking, Salt consumption, Number of warm meals

The assignment of the weights per characteristic and category was carried out by a specialist in health and nutrition, as she has the necessary specialist expertise and can assess the effects of consumption on health. Figure 2 shows a section of the weighting carried out for the characteristics and categories in the clinical picture of blood pressure.

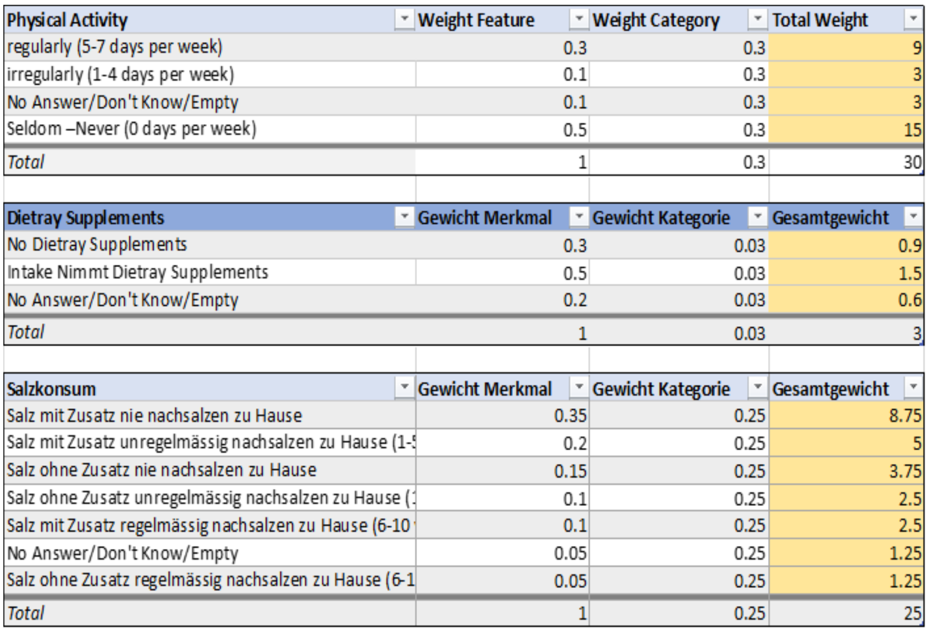


Categorie	ID	Features	Weighting Feature (% per category)	Weighting Category (% total of all categories)
Physical Activity	1	Seldom – Never (0 days per Week)	50	30
Physical Activity	2	Irregularly (1-4 days per week)	10	
Physical Activity	3	regularly (5-7 days per week)	30	
Physical Activity	4	No Answer/Don't Know/Empty	10	
Dietary Supplements	1	No Dietary Supplements	30	3
Dietary Supplements	2	takes Dietary Supplements ein	50	
Dietary Supplements	3	No Answer/Don't Know/Empty	20	
Salz Intake	1	Salz ohne Zusatz nie nachsalzen zu Hause	15	25
Salz Intake	2	Salz ohne Zusatz unregelmässig nachsalzen	10	
Salz Intake	3	Salz ohne Zusatz regelmässig nachsalzen zu	5	
Salz Intake	4	Salz mit Zusatz nie nachsalzen zu Hause	40	
Salz Intake	5	Salz mit Zusatz unregelmässig nachsalzen zu	20	
Salz Intake	6	Salz mit Zusatz regelmässig nachsalzen zu H	10	
Salz Intake	7	No Answer/Don't Know/Empty	5	
Smoking	1	Never	50	4
Smoking	2	Previously	40	
Smoking	3	Occasionally/Daily	5	
Smoking	4	No Answer/Don't Know/Empty	5	
Beverages	1	alcoholic	1	2
Beverages	a	Coffee/Tee	2	
Beverages	3	Coffee/Tea/Alcoholic	1	
Beverages	4	Coffee/Tee/Milk Drinks	3	

Figure 2. 1-Weighting of features and categories for Blood Pressure

6.1.2. Step 2: Calculation of the total weight per characteristic

Since the user-specific weighting was carried out by our dietary expert in two stages per category and characteristic, the total weight per characteristic was calculated in the second step. The total weight was required in the further course of the association analysis to derive the weight of an association rule from it. Figure 3 shows a section of the calculated total weights for the items in the categories exercise, dietary supplements, and salt consumption in the subject area of blood pressure. The two minimum values support and confidence were defined as user-specific parameters for each disease. Furthermore, the minimum and a maximum standard length for the number of characteristics of a disease was de-fined (Cengiz et al., 2019, p. 3). With the minimum rule length, the requirement was met that all highly weighted categories are included in the rules. The weights of all included categories add up to at least 95% of the total weight. As a result, only marginally relevant categories are usually not considered. For the maximum standard length, the clinical picture with all eight available categories from nutritional patterns was always used.



Physical Activity	Weight Feature	Weight Category	Total Weight
regularly (5-7 days per week)	0.3	0.3	9
irregularly (1-4 days per week)	0.1	0.3	3
No Answer/Don't Know/Empty	0.1	0.3	3
Seldom –Never (0 days per week)	0.5	0.3	15
Total	1	0.3	30

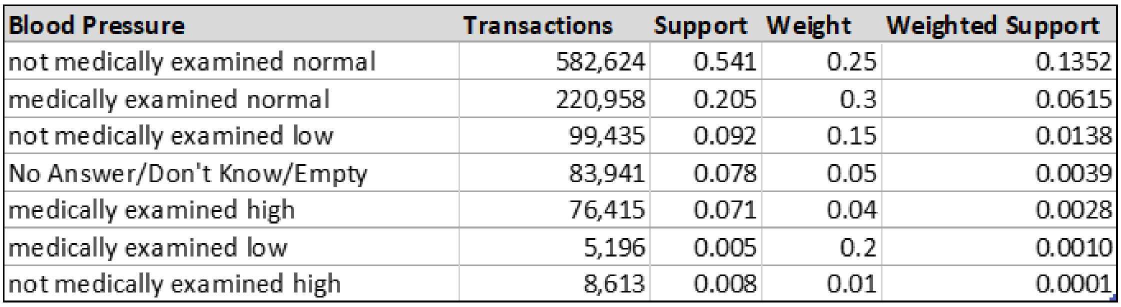
Dietray Supplements	Gewicht Merkmal	Gewicht Kategorie	Gesamtgewicht
No Dietray Supplements	0.3	0.03	0.9
Intake Nimmt Dietray Supplements	0.5	0.03	1.5
No Answer/Don't Know/Empty	0.2	0.03	0.6
Total	1	0.03	3

Salzkonsum	Gewicht Merkmal	Gewicht Kategorie	Gesamtgewicht
Salz mit Zusatz nie nachsalzen zu Hause	0.35	0.25	8.75
Salz mit Zusatz unregelmässig nachsalzen zu Hause (1-5	0.2	0.25	5
Salz ohne Zusatz nie nachsalzen zu Hause	0.15	0.25	3.75
Salz ohne Zusatz unregelmässig nachsalzen zu Hause (1	0.1	0.25	2.5
Salz mit Zusatz regelmässig nachsalzen zu Hause (6-10	0.1	0.25	2.5
No Answer/Don't Know/Empty	0.05	0.25	1.25
Salz ohne Zusatz regelmässig nachsalzen zu Hause (6-1	0.05	0.25	1.25
Total	1	0.25	25

Figure 3. Calculation of total weight per category by Blood Pressure

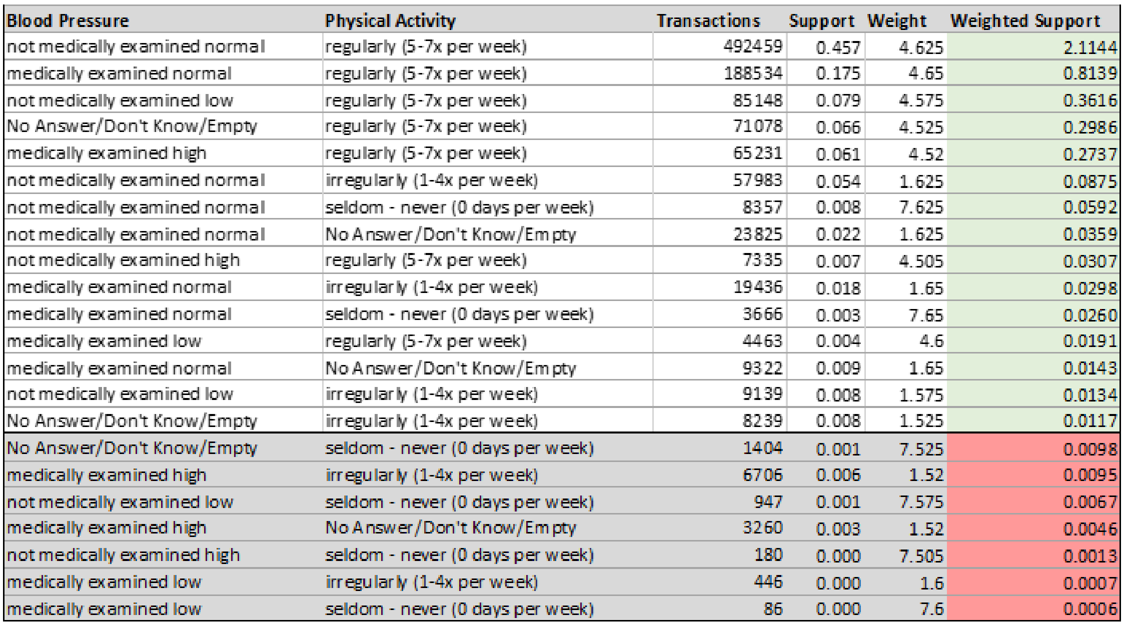
6.1.3. Step 3: Building frequent 2-Itemsets for Blood Pressure

The characteristics of the movement were combined with the characteristics of the blood pressure and from this the support, the weight and the weighted support were calculated. Movement was used for blood pressure for the 2-item set because it is the category with the highest user-specific weight (30%). The generation of all candidates from 2-item sets, 3-item sets, and all other item sets would have been too extensive for the scope of this study. For this reason, the procedure was descending according to the weight of the category and the characteristics were added to the item sets according to relevance. Sorting according to weight and the minimum length as parameters for the determination of association rules ensures that all characteristics of the clinical picture are included, which make up 95% of the total weight. See Figure 6. As can be seen from Figure 5, some item sets do not meet the predefined minimum support of 0.01. These subsets are marked in red in the figure and were no longer used to generate further candidates.



Blood Pressure	Transactions	Support	Weight	Weighted Support
not medically examined normal	582,624	0.541	0.25	0.1352
medically examined normal	220,958	0.205	0.3	0.0615
not medically examined low	99,435	0.092	0.15	0.0138
No Answer/Don't Know/Empty	83,941	0.078	0.05	0.0039
medically examined high	76,415	0.071	0.04	0.0028
medically examined low	5,196	0.005	0.2	0.0010
not medically examined high	8,613	0.008	0.01	0.0001

Figure 4. 1-itemset for Blood Pressure

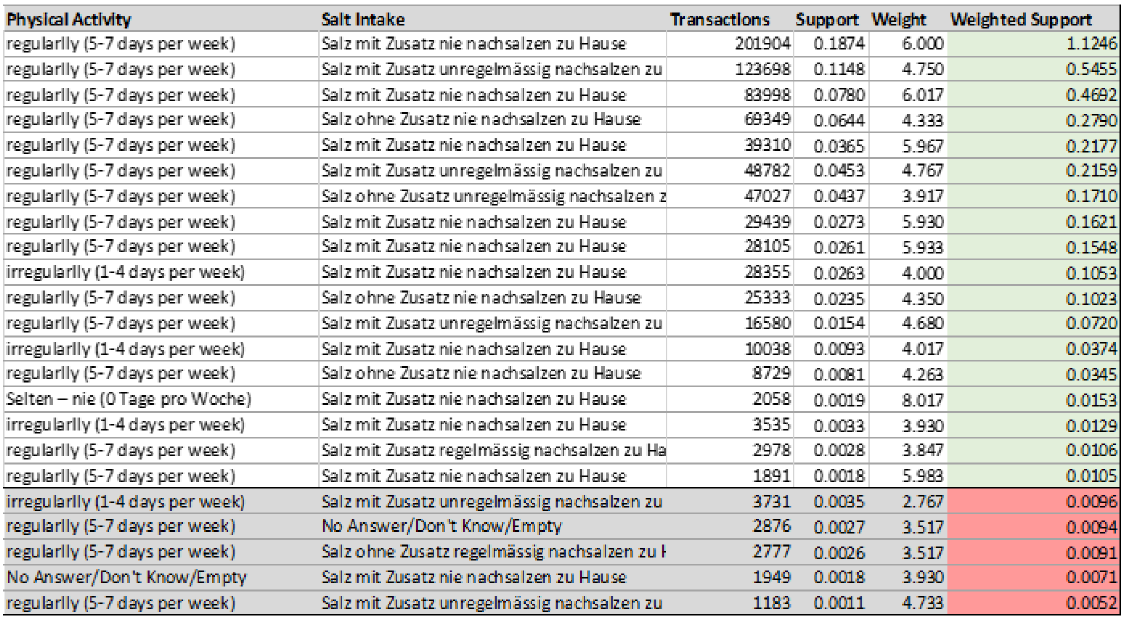


Blood Pressure	Physical Activity	Transactions	Support	Weight	Weighted Support
not medically examined normal	regularly (5-7x per week)	492459	0.457	4.625	2.1144
medically examined normal	regularly (5-7x per week)	188534	0.175	4.65	0.8139
not medically examined low	regularly (5-7x per week)	85148	0.079	4.575	0.3616
No Answer/Don't Know/Empty	regularly (5-7x per week)	71078	0.066	4.525	0.2986
medically examined high	regularly (5-7x per week)	65231	0.061	4.52	0.2737
not medically examined normal	irregularly (1-4x per week)	57983	0.054	1.625	0.0875
not medically examined normal	seldom - never (0 days per week)	8357	0.008	7.625	0.0592
not medically examined normal	No Answer/Don't Know/Empty	23825	0.022	1.625	0.0359
not medically examined high	regularly (5-7x per week)	7335	0.007	4.505	0.0307
medically examined normal	irregularly (1-4x per week)	19436	0.018	1.65	0.0298
medically examined normal	seldom - never (0 days per week)	3666	0.003	7.65	0.0260
medically examined low	regularly (5-7x per week)	4463	0.004	4.6	0.0191
medically examined normal	No Answer/Don't Know/Empty	9322	0.009	1.65	0.0143
not medically examined low	irregularly (1-4x per week)	9139	0.008	1.575	0.0134
No Answer/Don't Know/Empty	irregularly (1-4x per week)	8239	0.008	1.525	0.0117
No Answer/Don't Know/Empty	seldom - never (0 days per week)	1404	0.001	7.525	0.0098
medically examined high	irregularly (1-4x per week)	6706	0.006	1.52	0.0095
not medically examined low	seldom - never (0 days per week)	947	0.001	7.575	0.0067
medically examined high	No Answer/Don't Know/Empty	3260	0.003	1.52	0.0046
not medically examined high	seldom - never (0 days per week)	180	0.000	7.505	0.0013
medically examined low	irregularly (1-4x per week)	446	0.000	1.6	0.0007
medically examined low	seldom - never (0 days per week)	86	0.000	7.6	0.0006

Figure 5. Frequent 2-itemsets for Blood Pressure

6.1.4. Step 4: Building frequent 3-Itemsets

To generate the 3-item set, salt consumption was added as an additional feature to the shortened 2-item set of blood pressure and exercise. Salt consumption is the nutritional trait with the second highest weighting (25%) after exercise. Figure 6 shows an excerpt from the frequent subsets as a 3-item set, consisting of the three attributes mentioned for the topic of blood pressure.



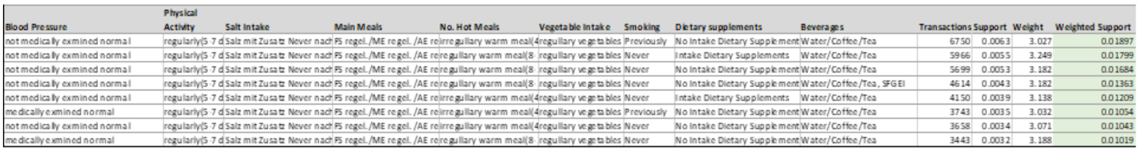
Physical Activity	Salt Intake	Transactions	Support	Weight	Weighted Support
regularlly (5-7 days per week)	Salz mit Zusatz nie nachsalzen zu Hause	201904	0.1874	6.000	1.1246
regularlly (5-7 days per week)	Salz mit Zusatz unregelmässig nachsalzen zu	123698	0.1148	4.750	0.5455
regularlly (5-7 days per week)	Salz mit Zusatz nie nachsalzen zu Hause	83998	0.0780	6.017	0.4692
regularlly (5-7 days per week)	Salz ohne Zusatz nie nachsalzen zu Hause	69349	0.0644	4.333	0.2790
regularlly (5-7 days per week)	Salz mit Zusatz nie nachsalzen zu Hause	39310	0.0365	5.967	0.2177
regularlly (5-7 days per week)	Salz mit Zusatz unregelmässig nachsalzen zu	48782	0.0453	4.767	0.2159
regularlly (5-7 days per week)	Salz ohne Zusatz unregelmässig nachsalzen z	47027	0.0437	3.917	0.1710
regularlly (5-7 days per week)	Salz mit Zusatz nie nachsalzen zu Hause	29439	0.0273	5.930	0.1621
regularlly (5-7 days per week)	Salz mit Zusatz nie nachsalzen zu Hause	28105	0.0261	5.933	0.1548
irregularlly (1-4 days per week)	Salz mit Zusatz nie nachsalzen zu Hause	28355	0.0263	4.000	0.1053
regularlly (5-7 days per week)	Salz ohne Zusatz nie nachsalzen zu Hause	25333	0.0235	4.350	0.1023
regularlly (5-7 days per week)	Salz mit Zusatz unregelmässig nachsalzen zu	16580	0.0154	4.680	0.0720
irregularlly (1-4 days per week)	Salz mit Zusatz nie nachsalzen zu Hause	10038	0.0093	4.017	0.0374
regularlly (5-7 days per week)	Salz ohne Zusatz nie nachsalzen zu Hause	8729	0.0081	4.263	0.0345
Selten – nie (0 Tage pro Woche)	Salz mit Zusatz nie nachsalzen zu Hause	2058	0.0019	8.017	0.0153
irregularlly (1-4 days per week)	Salz mit Zusatz nie nachsalzen zu Hause	3535	0.0033	3.930	0.0129
regularlly (5-7 days per week)	Salz mit Zusatz regelmässig nachsalzen zu Ha	2978	0.0028	3.847	0.0106
regularlly (5-7 days per week)	Salz mit Zusatz nie nachsalzen zu Hause	1891	0.0018	5.983	0.0105
irregularlly (1-4 days per week)	Salz mit Zusatz unregelmässig nachsalzen zu	3731	0.0035	2.767	0.0096
regularlly (5-7 days per week)	No Answer/Don't Know/Empty	2876	0.0027	3.517	0.0094
regularlly (5-7 days per week)	Salz ohne Zusatz regelmässig nachsalzen zu I	2777	0.0026	3.517	0.0091
No Answer/Don't Know/Empty	Salz mit Zusatz nie nachsalzen zu Hause	1949	0.0018	3.930	0.0071
regularlly (5-7 days per week)	Salz mit Zusatz unregelmässig nachsalzen zu	1183	0.0011	4.733	0.0052

Figure 6. Frequent 3-itemsets for Blood Pressure

6.1.5. Further steps: Building frequent 9-Itemsets

This iterative procedure was repeated until all categories of nutritional behavior in combination with the corresponding clinical picture were includ-ed as an item set. For each frequent item set, the support, weight and weighted support were calcu-lated. All subsets that met the weighted minimum support were then used to generate the next candi-date item set. The last iteration is the generation of the 9-itemset with all available features. A section of this can be seen in Figure 7 for the clinical pic-ture of blood pressure.

After all candidate item sets had been generat-ed, the confidence value for each item set was calculated. For this purpose, the support of the corre-sponding item set was divided by the support of the disease as a premise. To identify the association rules in the next step, all item sets are used that meet both the minimum weighted support and the predefined minimum confidence value.



Blood Pressure	Physical Activity	Salt Intake	Main Meals	No. Hot Meals	Vegetable Intake	Smoking	Dietary supplements	Beverages	Transactions	Support	Weight	Weighted Support
not medically exmined normal	regularly(5-7 d	Salz mit Zusatz Never nach	FS regel./ME regel. /AE re	irregullary warm meal(4	regullary vegetables	Previously	No Intake Dietary Supplement	Water/Coffee/Tea	6750	0.0063	3.027	0.01897
not medically exmined normal	regularly(5-7 d	Salz mit Zusatz Never nach	FS regel./ME regel. /AE re	regullary warm meal(8	regullary vegetables	Never	Intake Dietary Supplements	Water/Coffee/Tea	5966	0.0055	3.249	0.01799
not medically exmined normal	regularly(5-7 d	Salz mit Zusatz Never nach	FS regel./ME regel. /AE re	regullary warm meal(8	regullary vegetables	Never	No Intake Dietary Supplement	Water/Coffee/Tea	5699	0.0053	3.182	0.01684
not medically exmined normal	regularly(5-7 d	Salz mit Zusatz Never nach	FS regel./ME regel. /AE re	regullary warm meal(8	regullary vegetables	Never	No Intake Dietary Supplement	Water/Coffee/Tea, SFGEI	4614	0.0043	3.182	0.01363
not medically exmined normal	regularly(5-7 d	Salz mit Zusatz Never nach	FS regel./ME regel. /AE re	irregullary warm meal(4	regullary vegetables	Never	Intake Dietary Supplements	Water/Coffee/Tea	4150	0.0039	3.138	0.01209
medically exmined normal	regularly(5-7 d	Salz mit Zusatz Never nach	FS regel./ME regel. /AE re	irregullary warm meal(4	regullary vegetables	Previously	No Intake Dietary Supplement	Water/Coffee/Tea	3743	0.0035	3.032	0.01054
not medically exmined normal	regularly(5-7 d	Salz mit Zusatz Never nach	FS regel./ME regel. /AE re	irregullary warm meal(4	regullary vegetables	Never	No Intake Dietary Supplement	Water/Coffee/Tea	3658	0.0034	3.071	0.01043
medically exmined normal	regularly(5-7 d	Salz mit Zusatz Never nach	FS regel./ME regel. /AE re	regullary warm meal(8	regullary vegetables	Never	No Intake Dietary Supplement	Water/Coffee/Tea	3443	0.0032	3.188	0.01019

Figure 7. Excerpt of frequent 9-itemsets for Blood Pressure

6.1.6. Final step: Building Association Rules

The identification of the association rules is the second phase of the a priori procedure. The frequent patterns of the database from the previous step were used to derive interesting association rules using the calculated confidence value. From the frequent quantities of the four subject areas, all item sets were extracted that meet the specified minimum confidence value and the minimum standard length. Thereafter, all subsets were removed whose disease diagnosis was not assessed by a healthcare professional. Finally, the results were sorted in descend-ing order by confidence.

7. RESULTS USING WEIGHTED APRIORI ALGORITHM (WARM)

After completion of the algorithm rules were found that show the relationship between nutrition and chronic diseases. We report in the following gained rules for alcohol, blood pressure, cholesterol, and diabetes.

7.1. Rules for Alcohol

Rule 1: 4.1% of the sample, who consume 0-17 grams of alcohol daily, show the following characteristics: they eat 3 times a day regularly, they drink water, coffee or tea between the meals, they eat irregularly hot meals, they prepare more than twice a week vegetable, they exercise regularly five to 10 times a week and take nutritional supplements. This rule occurs in 3.4% of the sample

Rule 2: 3.6% of the sample, who consume 0-17 grams of alcohol daily, show the following characteristics: the following characteristics: they eat 3 times a day regularly, they drink water, coffee, or tea between the meals, they eat regularly hot meals, they prepare more than twice a week vegetable, they exercise regularly five to 10 times.

Rule 3: 3.1% of the sample, who consume 18-22 grams of alcohol daily, show the following characteristics: the following characteristics: they eat 3 times a day regularly, they drink water, coffee, or tea between the meals, they eat irregularly hot meals, they prepare more than twice a week vegetable, they exercise regularly five to 10 times a week and take nutritional supplements. This rule occurs in 0.2% of the sample.

Rule 4: 2.9% of the sample, who consume 18-22 grams of alcohol daily, show the following characteristics: the following characteristics: they eat 3 times a day regularly, they drink water, coffee or tea between the meals, they eat regularly hot meals, they pre-pare more than twice a week vegetable, they exercise regularly five to 10 times a week and take no nutritional supplements. This rule occurs in 0.19% of the sample.

Rule 5: 2.3% of the sample, who consume more than 22 grams of alcohol daily, show the following characteristics: the following characteristics: they eat 3 times a day regularly, they drink water, coffee, or tea between the meals, they eat seldom or almost no hot meals, they don't prepare vegetables, they exercise regularly five to 10 times a week and take no nutritional supplements. This rule occurs in 0.17% of the sample.

Rule 6: 6.6% of the sample, who have a medic, show the following characteristics: they eat 3 times a day regularly, they drink water, coffee, or tea between the meals, they eat seldom or almost no hot meals, they prepare vegetables more than twice a week, they exercise regularly five to 10 times a week and take no nutritional supplements. This rule occurs in 0.16% of the sample.

7.2. Rules for Blood Pressure

Rule 1: 6.6% of the sample, who have medically assessed high blood pressure, show the following characteristics: they eat 3 times a day regularly, they eat regularly hot meals, they prepare vegetables more than twice a week, they exercise regularly five to 7 times a week, they cook with salt but don't add salt during eating and don't smoke or haven't ever smoked. This rule occurs in 0.47% of the sample.

Rule 2: 5.1% of the sample, who have medically assessed high blood pressure, show the following characteristics: they eat 3 times a day regularly, they eat irregularly hot meals, they prepare vegetables more than twice a week, they exercise regularly five to 7 times a week, they cook with salt but don't add salt during eating and don't smoke or haven't ever smoked. This rule occurs in 0.36% of the sample.

Rule 3: 4.1% of the sample, who have medically assessed high blood pressure, show the following characteristics: they eat 3 times a day regularly, they eat irregularly hot meals, they prepare vegetables more than twice a week, they exercise regularly five to 7 times a week, they cook with salt but don't add salt during eating and don't smoke but have previously smoked. This rule occurs in 0.29% of the sample.

Rule 4: 6.5% of the sample, who have medically assessed normal blood pressure, show the following characteristics: they eat 3 times a day regularly, they eat irregularly hot meals, they prepare vegetables more than twice a week, they exercise regularly five to 7 times a week, they cook with salt but don't add salt during eating and don't smoke or haven't ever smoked. This rule occurs in 1.33% of the sample.

Rule 5: 5.3% of the sample, who have medically assessed normal blood pressure, show the following characteristics: they eat 3 times a day regularly, they eat regularly hot meals, they prepare vegetables more than twice a week, they exercise regularly five to 7 times a week, they cook with salt but don't add salt during eating and don't smoke or haven't ever smoked. This rule occurs in 1% of the sample.

Rule 6: 5% of the sample, who have medically assessed normal blood pressure, show the following characteristics: they eat 3 times a day regularly, they eat regularly hot meals, they prepare vegetables more than twice a week, they exercise regularly five to 7 times a week, they cook with salt but don't add salt during eating and don't smoke but have previously smoked. This rule occurs in 0.78% of the sample.

7.3. Rules for Cholesterol

Rule 1: 6.3% of the sample, who have medically assessed high cholesterol, show the following characteristics: they eat 3 times a day regularly, they eat irregularly hot meals, they prepare vegetables more than twice a week, they exercise regularly five to 7 times a week, and don't smoke or haven't ever smoked. This rule occurs in 0.21% of the sample.

Rule 2: 4.8% of the sample, who have medically assessed high cholesterol, show the following characteristics: they eat 3 times a day regularly, they eat regularly hot meals, they prepare vegetables more than twice a week, they exercise regularly five to 7 times a week, they cook with salt but don't add salt during eating and don't smoke or haven't ever smoked. This rule occurs in 1.33% of the sample.

Rule 3: 3.8% of the sample, who have medically assessed high cholesterol, show the following characteristics: they eat 3 times a day regularly, they eat regularly hot meals, they prepare vegetables more than twice a week, they exercise regularly five to 7 times a week, they cook with salt but don't add salt during eating and don't smoke but have previously smoked. This rule occurs in 1.33% of the sample.

Rule 4: 6.3% of the sample, who have medically assessed normal cholesterol, show the following characteristics: they eat 3 times a day regularly, they eat regularly hot meals, they prepare vegetables more than twice a week, they exercise regularly five to 7 times a week, they cook with salt but don’t add salt during eating and don’t smoke or haven’t ever smoked. This rule occurs in 0.76% of the sample.

Rule 5: 4.4% of the sample, who have medically assessed normal cholesterol, show the following characteristics: they eat 3 times a day regularly, they eat irregularly hot meals, they prepare vegetables more than twice a week, they exercise regularly five to 7 times a week, they cook with salt but don’t add salt during eating and don’t smoke or haven’t ever smoked. This rule occurs in 0.54% of the sample

7.4. Rules for Diabetes

Rule 1: 6.3% of the sample, who have medically assessed high diabetes, show the following characteristics: they eat 3 times a day regularly, they drink water, coffee, or tea between meals, they eat regularly hot meals, they prepare vegetables more than twice a week, they exercise 2 to 5 times a week, they don’t take any meal supplements. This rule occurs in 0.04% of the sample.

Rule 2: 3.3% of the sample, who have medically assessed high diabetes, show the following characteristics: they eat 3 times a day regularly, they drink water, coffee, or tea between meals, they eat regularly hot meals, they prepare vegetables more than twice a week, they exercise 2 to 5 times a week, they take any meal supplements. This rule occurs in 0.04% of the sample.

Rule 3: 3.3% of the sample, who have medically assessed high diabetes, show the following characteristics: they eat 3 times a day regularly, they drink water, coffee, or tea between meals, they eat irregularly hot meals, they prepare vegetables more than twice a week, they exercise 2 to 5 times a week, they don’t take any meal supplements. This rule occurs in 0.04% of the sample.

Rule 4: 3.4% of the sample, who have medically assessed normal diabetes, show the following characteristics: they eat 3 times a day regularly, they drink water, coffee, or tea between meals, they eat regularly hot meals, they prepare vegetables more than twice a week, they exercise 2 to 5 times a week, they don’t take any meal supplements. This rule occurs in 0.14% of the sample.

Rule 5: 3.3% of the sample, who have medically assessed normal diabetes, show the following characteristics: they eat 3 times a day regularly, they drink water, coffee, or tea between meals, they eat regularly hot meals, they prepare vegetables more than twice a week, they exercise 2 to 5 times a week, they take meal supplements. This rule occurs in 0.13% of the sample.

Rule 6: 2.9% of the sample, who have medically assessed normal diabetes, show the following characteristics: they eat 3 times a day regularly, they drink water, coffee, or tea between meals, they eat regularly hot meals, they prepare vegetables more than twice a week, they exercise 2 to 5 times a week, they take any meal supplements. This rule occurs in 0.11% of the sample.

8. KNOWLEDGE INTERPRETATION

8.1. Alcohol

The consumption of alcohol is culturally strongly anchored in Western society and has been an integral part of social life for centuries. Excessive alcohol consumption is nevertheless a major cause of premature mortality and damage to physical (especially liver and digestive disorders), mental and social health. It is also a cause of violent behavior, accidents, early disability, lost work, or social exclusion.

Rules 1 and 2 are probably people with a high level of health and nutrition awareness. Warm meals are an expression of an even greater nutritional awareness.

Rules 3 and 4 are about people who have found a good balance between nutrition and health awareness on the one hand, and enjoyment of life and culture on the other. If alcohol and cuisine are sensibly combined, this indicates a balanced person.

Rules 5 and 6 with an alcohol consumption of more than 28 grams of alcohol per day result in significant health disadvantages, especially on the liver. Without vegetables, important B vitamins are missing, which can lead to a manifest deficiency for alcoholics. In this case, these micronutrients should be taken with supplements.

According to the rules, alcohol consumption in quantities of 18-22 grams per day is ideal for health. A complete renunciation of alcohol or slightly above-average consumption is not absolutely necessary as long as the alcohol dehydrogenase can oxidize the amount absorbed into acetaldehyde and acetic acid. If this metabolic reaction is permanently overwhelmed, it ultimately leads to a hardened liver (cirrhosis), which increasingly hinders the blood flow. This blood flow is necessary insofar as the liver is the most important detoxification organ of the organism. This gives people an advantage who can find a good balance between indulgence in modest amounts and avoiding chronic consumption of alcohol. Exercise can help improve circulation in the liver as well. It is ideally combined with plenty of fluids (drinking water, coffee, tea) between meals to eliminate metabolic waste products from the liver. This behavior is followed for all rules.

The consumption of regular but not too frequent meals provides building blocks and energy sources. The building blocks that are not used immediately are deposited until they are hungry, in which they are released from the depot. The distinction between warm and not regularly warm meals can be an expression of nutritional awareness, but also the result of time management (e.g., sandwich lunch). It does little to alleviate or increase the effects of alcohol consumption. The frequency of meals is important if a cantable (consuming) metabolic process predominates (e.g., a serious illness such as cancer, obstructive pulmonary disease, HIV, or others). In this case, food has to be fed frequently or continuously. In healthy people, eating twice a day can be much healthier than eating three times a day (interval diet). The consumption of vegetables is important for the supply of vitamins of the B group, which help the energy metabolism (citrate cycle and respiratory chain) to provide energy. This is important for endurance athletes. For sprinters and strength athletes, the short-term energy is mainly taken from the carbohydrate metabolism. Vegetable intake is an expression of a high level of health awareness. For athletes, vegetable intake can usefully be supplemented with supplements, especially for endurance sports.

8.2. Blood Pressure

High blood pressure is a chronic disease that is usually caused by a decline in cardiac output, declining kidney function (lack of filtration capacity) and the hardening of the blood vessels. Often one finds genetic predispositions combined with improper nutrition. Excess fats and cholesterol in particular lead to vascular problems in the fat metabolism. This can cause excessive foam cells to form and lead to diabetes. In this respect, high blood pressure, hyperlipidemia, hypercholesterolemia and diabetes (type 2) are often associated with one another. A healthy heart can fight against increased counter pressure for years and overcome it until a stroke suddenly occurs in the heart or brain. As a comparison, imagine a garden pump for irrigation with a hose and sprinkler. A clogged sprinkler and calcified hoses will sooner or later bring any engine to a standstill due to overheating if the power is no longer sufficient to transport water.

Rules 1, 2 and 3 contain some references to culinary lovers. This is not a problem as long as the weight is sufficiently controlled. In the case of high blood pressure, any weight reduction is helpful in reducing the symptoms.

Rules 4, 5 and 6 concern people who have normal blood pressure (normotonic). They eat irregularly warm meals more often, but they are nutritionally conscious in terms of vegetables and regular food intake.

The most important subset in this subject area are people with medically diagnosed high blood pressure. Weight reduction, exercise, smaller amounts of food and less frequent food intake help here.

In the case of high blood pressure, the warm food is probably an indication of health awareness. Salt should be reduced when the kidney is already inadequate. In terms of blood pressure rules, salt consumption is the same everywhere. Vegetables are also fed in all cases.

8.3. Cholesterol

Hypercholesterolemia is a disease of the fat metabolism. Cholesterol is produced by the body's own biosynthesis. If too much carbohydrate is ingested, it is broken down into glucose. The breakdown product acetyl-CoA can be linked to fatty acids or via steroid biosynthesis to hormones such as sex hormones, corticoids or mineralocorticoids. The internal excess cholesterol biosynthesis is genetically determined and more difficult to control than the dietary cholesterol. Hypercholesterolemia usually leads to vascular problems, atherosclerosis, diabetes or a stroke.

Rules 1 to 5 have identical physical activity. It is important for people to promote the breakdown of cholesterol.

Rules 1 to 3 show that, as with high blood pressure, health awareness is also well developed in diagnosed hypercholesterolemics. This is shown by the fact that the people do not smoke and in some cases have not smoked before. Smoking would seal the vessels even more. The behavioral patterns are also the same when it comes to the intake of salt and vegetables.

The difference between **Rules 1 to 3** and **Rules 4 to 5** is probably only quantitative in relation to fat intake. However, this was not collected and could not be considered in this research.

8.4. Diabetes

Diabetes is a carbohydrate metabolism disease. Diabetes is divided into type 1 diabetes and type 2 diabetes, which occurs with increasing age. As with cholesterol, genetics are also essential in glucose metabolism. The disease is found genetically more frequently in certain ethnic groups. As an example, people in the Indian subcontinent from countries like India, Pakistan, Tibet or Nepal can be named. In this area, most people die from cardiovascular problems at a young age. Gestational diabetes can often be diagnosed in women due to their genetic predisposition. In this case, the high progesterone level during pregnancy leads to gluconeogenesis and thus to increased blood glucose levels. It is for this reason that the term gestational diabetes is also used.

A permanently high blood sugar level leads to the exhaustion of the insulin secretion and the islet cells of Langerhans. While previously the insulin injection and irritation of the pancreas were used to release more insulin, there are now new antidiabetic drugs that slow down the digestion of carbohydrates in the gastrointestinal tract and thus lead to a massively higher efficiency of the pancreatic work. Researchers have discovered the mechanism in lizards, which ingest prey at very long intervals and digest it so slowly that the next food intake is not necessary for a long time.

Rules 1 to 3 show that a confirmed diagnosis of diabetes leads to significantly better health and nutritional behavior and understanding of nutrition. This is probably reflected in the great care taken in the preparation of meals (warm, with lots of vegetables and regularly in small quantities).

From **Rules 1 to 3** it can also be inferred that the intake of food supplements has little effect on the metabolic situation, unless movement is increased drastically and energy is generated with the micronutrients (B vitamins, iron, copper and the like) improved.

With **Rules 4 to 6**, people without a medical diagnosis of diabetes generally eat less regularly, in a lavish manner, with or without supplements.

9. CONCLUSION AND FUTURE WORK

In this paper, we apply a data mining method such as WARM Apriori algorithm to a big integrated database comprising of Swiss nutrition and health data to gain rules that show the effects of nutritional habits on some chronical diseases such as high alcohol consumption, high blood pressure, Diabetes, and high Cholesterol. For this purpose, we use an integrated database comprised of collected data from various Swiss national surveys as reported in a previous study (Lustenberger, 2021). Our database includes health data on the state of health and health-related habits of around 85,000 people over a period of 25 years (1992-2012) as well as data on food consumption, cooking, eating and physical activity habits of around 2,000 people in 2015 and 2016.

A deficiency in the data used for data mining is the deviating collection period for health and consumption data. While the health data was collected from the Swiss resident population every five years over a period of 25 years, data for food consumption comes from a national survey from 2014 to 2015. This affects the result insofar as consumer habits as a cause is not directly reflected in health as an effect. The habits of the population, such as physical activities or diet, changes over a longer period and in turn has an impact on health. Ideally, both surveys should be carried out repeatedly to gain even more informative value of the results and to ensure the causal
The algorithmic improvements as suggested in the study by Mewes et al. could be successfully implemented in the execution of the association analysis. The problem with removing sets of attribute values when using the a priori algorithm was solved with the WARM method. Using an expert weighting of characteristics enables the related dietary relevance to be better considered in the results than is the case with pure Apriori and ensures this relevance to not be pruned out in the mining process.

The interpretation of the derived rules reveals interesting aspects about the selected Swiss population subgroup. The study shows that the consumption of alcohol in small quantities does not have a negative impact on health. In addition to this, regular exercise in combination with an adequate increase in fluids in the form of water, coffee or tea between meals improves the circulation in the liver and helps to eliminate the metabolic waste products. Additionally, the consumption of vegetables is also important for the supply of vitamins of the B group, which help the energy metabolism to provide energy. These vitamins are particularly lacking in alcoholics and should then be taken with supplements.

Furthermore, for people with medically diagnosed high blood pressure, weight loss, regular exercise, smaller amounts of food and less frequent food intake help as measures. The analysis shows that the people in the sample with high blood pressure probably gained better health awareness because of the diagnosis. Smoking habit with diagnosed high blood pressure is either non-smokers or a status after smoking cessation. In addition, these people eat hot meals more often, which is also an indication of awareness. Notwithstanding, for an even more comprehensive assessment within the scope of this study, the body mass index (BMI) is missing as an important risk factor.

Regarding a high cholesterol value (hypercholesterolemia) as a disease of the fat metabolism, a low-cholesterol diet can be effective, but only if the genetic predisposition is not given and a fat intake that is much too high is reduced. If the person concerned already consumes little fat, an improvement in the diet can hardly be achieved. In addition, people diagnosed with high cholesterol levels show a pronounced health awareness. This is shown in the presented study by the fact that the people do not smoke. Smoking would, in addition to causing illness, block the blood vessels even more.

Furthermore, the rules and patterns in diabetes (type 2) show that the medical diagnosis of diabetes leads to significantly better health and nutritional habits and understanding of nutritional values of food intake. This was shown by the balanced preparation of meals (warm, vegetable-rich, and regular). Another finding is that dietary supplements do little when it comes to diabetes.

REFERENCES

[1] WHO, 2003. Diet, Nutrition, and the Prevention of Chronic Diseases. Report of a Joint WHO/FAO Ex-pert Consultation. In World Health Organization aper templates.

[2] Fardet, A., Boirie, Y. 2008. Associations between food and beverage groups and major diet-related chronic diseases: an exhaustive review of pooled/meta-analyses and systematic reviews, In Nutr Rev. 2014 Dec; 72(12):741-62. doi: 10.1111/nure.12153

[3] Fardet, A. Richonnet, C., Mazur, A., 2019, Association between consumption of fruit or processed fruit and chronic diseases and their risk factors: a systematic review of meta-analyses, Nutrition Reviews. In Nu-trition Reviews, Volume 77, Issue 6, Pages 376-387.

[4] Schneider, S., Heuterne, X., 2000, Moore, R., Lopes, J., 1999. Prediction Model for Health-Related Quality of Life of Elderly with Chronic Diseases using Ma-chine Learning Techniques. In Healthc Inform Res. 2014 Apr;20(2):125-134.

[5] Kee, S. K., Son, Y. J, Kim H.G., Lee J. Il., Cho, H.S., Lee, S., 2014, Associations between food and bever-age groups and major diet-related chronic diseases: an exhaustive review of pooled/meta-analyses and systematic reviews, In Nutr Rev. 2014 Dec; 72(12):741-62. doi: 10.1111/nure.12153

[6] Qudsi, D., Kartiwi, M., Saleh, N.B., 2017, Predictive data mining of chronic diseases using decision tree: A case study of health insurance company in Indonesia. In International Journal of Applied Engineer-ing Research 12(7):1334-1339

[7] Lei Z., Yang, S., Liu, H., Aslam, S., Liu, J., Bugingo, E., Zhang, D., 2018, Mining of Nutritional Ingredi-ents in Food for Disease Analysis, In IEEE Access 6(1):52766-52778

[8] McCabe, R.M, Adomavicius, G., Johnson P.E., Rund, E., Rush, A., Sperl-Hillen, A., 2008, Using Data Mining to Predict Errors in Chronic Disease Care, Advances in Patient Safety: In New Directions and Alternative Approaches in Vol. 3: Performance and Tools.

[9] Haslam, D.W., James, W.P.T., Obesity, In the Lancet, Volume 366, Issue 9492, Pages 1197-1209

[10] Lange, K.W., James W.P.T., Makulska-Gertruda E., Nakamura Y., Reissmann, A., 2008, A. Sperl-Hillen, Using Data Mining to Predict Errors in Chronic Disease Care, Advances in Patient Safety. In New Directions and Alternative Approaches (Vol. 3: Performance and Tools)

[11] Hearty, A.P., Gibney, M.J., 2008, A. Richonnet, C., Mazur, A., Analysis of meal patterns with the use of supervised data mining techniques—artificial neural networks and decision trees, In The American Journal of Clinical Nutrition, Volume 88, Issue 6, Pages 1632–1642.

[12] Von Ruesten, A., Feller, S., Bergmann, N.M., Boeing, H., 2013, S., Diet and risk of chronic diseases: results from the first 8 years of follow-up in the EPIC-Potsdam study, In European Journal of Clinical Nutrition volume 67, pages412–419.

[13] Yu E. Y. W., Wesselius A., Sinhart C., Wolk A., 2020, A data mining approach to investigate food groups related to incidence of bladder cancer, In the Bladder cancer Epidemiology and Nutritional Determinants International Study, Cambridge University Press

[14] Einsele, F., Sadeghi, L., Ingold, R., Jenzer, H., 2015, A Study about Discovery of Critical Food Consumption Patterns Linked with Lifestyle Diseases using Data Mining Methods, In HealthInf, BIOSTEC - International Joint Conference on Biomedical Engineering Systems and Technologies, Lisbon.

[15] Mewes I., Jenzer H., Einsele, F., 2020, building an integrated relational database from Swiss Nutrition's (menuCH) and Swiss Health datasets for Data Mining Purposes, submitted and accepted In ICAFNH 2021: International Conference on Agrilife, Food, Nutrition and Health

[16] BLV (2021). https://www.blv.admin.ch/blv/de/home/ lebensmittel-und-ernaehrung /ernaehrung/ menuch.html, date: 2/12/2021.

[17] BAG (2021). https://www.bag.admin.ch/bag/de/home/ zahlen-und-statistiken.html, date: 2/12/2021.

[18] Lustenberger, T., Jenzer, H., Einsele, F., 2021, Building an Integrated Relational Database from Swiss Nutrition's (menuCH) and multiple Swiss Health Datasets acquired from 1992 to 2012 for Data Mining Purposes, In Proceedings of the 10th International Conference on Data Science, Technology and Applications (DATA 2021), pages 150-15

[19] Agrawal R., Srikant, R., 1994, Fast algorithms for mining association rules. In IBM Research Report RJ9839, IBM Almaden Research Center, San Jose, California

[20] Cleve, J. & Lämmel, U. (Hg.). (2016). De Gruyter Studium. Data Mining (2. Aufl.). De Gruyter. https://doi.org/10.1515/9783110456776

[21] Cengiz, A. B., Birant, K. U. & Birant, D. (2019). Analysis of Pre-Weighted and Post-Weighted Association Rule Mining. In 2019 Innovations in Intelligent Systems and Applications Conference (ASYU).

AUTHORS

Farshideh Einsele, Prof. Dr., Lecturer and researcher in the Business section of Bern University of Applied Sciences.
She teaches business informatic subjects and her research work is dedicated to epidemiology, big data, and data mining.

Helena Jenzer, Dr., Chief Pharmacist FPH - PhD, university, Hospital PUK Zurich, formerly; Prof Dr Head of applied R&D Nutrition & Dietetics in Bern University of Applied Sciences.

Timo Lustenberger, Student, and member of research team Epidemiology Business Department of Bern University of Applied Sciences.

AUTHOR INDEX