Natarajan Meghanathan
Dhinaharan Nagamalai (Eds)

Computer Science & Information Technology

8th International Conference on Signal, Image Processing and
Pattern Recognition (SIPP 2020)
March 21~22, 2020, Vienna, Austria

AIRCC Publishing Corporation

Volume Editors

Natarajan Meghanathan,
Jackson State University, USA
E-mail: nmeghanathan@jsums.edu

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

ISSN: 2231 - 5403
ISBN: 978-1-925953-16-9
DOI: 10.5121/csit.2020.100201- 10.5121/csit.2020.100210

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

Preface

The 8th International Conference on Signal, Image Processing and Pattern Recognition (SIPP 2020) March 21~22, 2020, Vienna, Austria, International conference on Big Data, Machine learning and Applications (BIGML 2020), 5th International Conference on Data Mining & Knowledge Management (DaKM 2020), 5th International Conference on Software Engineering (SOEN 2020) and 8th International Conference on Artificial Intelligence, Soft Computing (AISC 2020), was collocated with 8th International Conference on Signal, Image Processing and Pattern Recognition (SIPP 2020). 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 SIPP 2020, BIGML 2020, DaKM 2020, SOEN 2020 and AISC 2020 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, SIPP 2020, BIGML 2020, DaKM 2020, SOEN 2020 and AISC 2020 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 SIPP 2020, BIGML 2020, DaKM 2020, SOEN 2020 and AISC 2020.

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.

Natarajan Meghanathan
Dhinaharan Nagamalai (Eds)

General Chair

Organization

Natarajan Meghanathan — Jackson State University, USA
Dhinaharan Nagamalai (Eds), — Wireilla Net Solutions, Australia

Program Committee Members

Aashish A.Bardekar, — Sipna College of Engineering & Technology, India
Abdel-Badeeh M. Salem, — Ain Shams University, Egypt
Abdulhamit Subasi, — Effat University, Saudi Arabia
Abhishek Shukla, — Singhania University, India
Addisson Salazar, — Universitat Politècnica de València, Spain
Ajune Wanis Ismail, — Universiti Teknologi Malaysia, Malaysia
Alejandro Regalado Mendez, — Universidad del Mar, Mexico
Alessio Ishizaka, — University of Portsmouth, England
Anandi Giridharan, — Indian Institute of Science, India
Anas M.R. AlSobeh, — Yarmouk University, Jordan
Asimi Ahmed, — Ibn Zohr University, Morocco
Ayman Kassem, — Cairo University, Egypt
Azah Kamilah Muda, — Universiti Teknikal Malaysia Melaka, Malaysia
Azian Azamimi Abdullah, — Universiti Malaysia Perlis, Malaysia
Azizollah Babakhani, — Babo Noshirvani University of Tecnology, Iran
Barbara Pekala, — University of Rzeszow, Poland
Barhoumi Walid, — SIIVA-LIMTIC Laboratory, Tunisia
Bichitra Kalita, — Assam Don Bosco University, India
Bilal H. Abed-alguni, — Yarmouk University, Jordan
Bouchra Marzak, — Hassan II University, Morocco
Changsoo Je, — Sogang University, South Korea
Dac-Nhuong Le, — Haiphong University, Vietnam
Deepak Garg, — Bennett University, India
Donatella Giuliani, — University of Bologna, Italy
Edmund Lai, — Auckland University of Technology, New Zealand
Fabio Gasparetti, — Roma Tre University, Italy
Farshad Safaei, — Shahid Beheshti University, Iran
Federico Tramarin, — University of Padova, Italy
Felix Yang Lou, — City University of Hong Kong, China
Fernando Tello Gamarra, — Federal University of Santa Maria, Brazil
Fernando Zacarias Flores, — Benemerita Universidad Autonoma de Puebla, Mexico
Guilong Liu, — Beijing Language and Culture University, China
H.Amca, — Eastern Mediterranean University, Turkey
Hacer Yalim Keles, — Ankara University, Turkey
Hamid Ali Abed AL-Asadi, — Basra University, Iraq
Hamlich Mohamed, — ENSAM Casablanca, Morroco
Hanming Fang, — Logistical Engineering University, China
Hao-En Chueh, — Yuanpei University, Taiwan
Hassan Ugail, — University of Bradford, UK
Henrique Joao Lopes Domingos, — New University of Lisbon, Portugal
Huang Lianfen, — Xiamen University, China

Vijayakumar V,	Multimedia University, Malaysia
Walid Barhoumi,	University of Carthage, Tunisia
Wee kuok kwee,	Multimedia University, Malaysia
Weili Zhang,	eBay Inc., USA
Wenwu Wang,	University of Surrey, UK
Wladyslaw Homenda,	Warsaw University of Technology, Poland
Yacef Fouad,	Division Productique et Robotique, Algeria
Yuan Tian,	King Saud Univerity, Saud Arabia
Yu-Chen Hu,	Providence University, Taiwan

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Artificial Intelligence Community (AIC)

Soft Computing Community (SCC)

Digital Signal & Image Processing Community (DSIPC)

Organized By

Academy & Industry Research Collaboration Center (AIRCC)

TABLE OF CONTENTS

8th International Conference on Signal, Image Processing and Pattern Recognition (SIPP 2020)

Privacy-Preserving Pattern Recognition with Image Compression….....01 - 12
Takayuki Nakachi and Hitoshi Kiya

Local Gray World Method for Single Image Dehazing..... 13 - 20
Vedran Stipetić and Sven Lončarić

International conference on Big Data, Machine learning and Applications (BIGML 2020)

MeramalNet: A Deep Learning Convolutional Neural Network for Bioactivity Prediction in Structure-Based Drug Discovery…….21 - 37
Hentabli Hamza, Naomie Salim, Maged Nasser and Faisal Saeed

Data Mining and Machine Learning In Earth Observation – An Application for Tracking Historical Algal Blooms……...39 - 58
Alexandria Dominique Farias and Gongling Sun

5th International Conference on Data Mining & Knowledge Management (DaKM 2020)

Two Approaches Toward Graphical Definitions of Knowledge and Wisdom.. 59 - 74
Mark Atkins

5th International Conference on Software Engineering (SOEN 2020)

From Quality Assurance to Quality Engineering for Digital Transformation……75 - 82
Kiran Kumaar CNK

Design of Software Trusted Tool Based on Semantic Analysis …….. 83 - 90
Guofengli

8th International Conference on Artificial Intelligence, Soft Computing (AISC 2020)

Building A Bi-Objective Quadratic Programming Model for the Support Vector Machine ..…….……..…….……..……. 91 - 98
Mohammed Zakaria Moustafa, Mohammed Rizk Mohammed, Hatem Awed Khater and Hager Ali Yahia

A Brief Survey of Data Pricing for Machine Learning.……..….. 99 - 110
Zuoqi Tang, Zheqi Lv, Chao Wu, Zhejiang University, China

Research on Farmland Pest Image Recognition Based on Target Detection Algorithm.…….....….. 111 - 117
Shi Wenxiu and Li Nianqiang, University of Jinan, China

PRIVACY-PRESERVING PATTERN RECOGNITION WITH IMAGE COMPRESSION

Takayuki Nakachi[1] and Hitoshi Kiya[2]

[1]Nippon Telegraph and Telephone Corporation,Kanagawa, Japan
[2]Tokyo Metropolitan University, Tokyo, Japan

ABSTRACT

In this paper, we propose a privacy-preserving pattern recognition scheme that well supports image com- pression. The proposed scheme is based on secure sparse coding using a random unitary transform. It offers the following two prominent features: 1) It is capable of pattern recognition in the encrypted image domain. Even if data leaks, privacy can be maintained because data remains encrypted. 2) It realizes Encryption-then-Compression (EtC) systems, where image encryption is conducted prior to compression. The pattern recognition can be carried out in the compressed signal domain using a few sparse coefficients. Based on the pattern recognition result, it can compress the selected images with high quality by estimat- ing sufficient number of sparse coefficients. We use the INRIA dataset to demonstrate its performance in detecting humans. The proposal is shown to realize human detection with encrypted images and efficiently compress the images selected in the image recognition stage.

KEYWORDS

Surveillance Camera, Pattern Recognition, Secure Computation, Sparse Coding, Random Unitary Trans- form

1. INTRODUCTION

With the increase in threats and criminal activity, security is seen as a major public concern. Image/video surveillance is one approach to addressing this issue. Many image/video surveillance systems are now widely deployed in many public spaces such as airports, banks, shopping streets, public streets, etc., and they are recording huge amounts of image/video every day. Fortunately, edge/cloud computing offers an efficient way of handling and analyzing the huge amounts of image/video data. However, edge/cloud computing poses some serious issues for end users, such as unauthorized use, data leaks, and privacy failures due to the unreliability of providers and accidents [1].

Many studies have examined the processing of encrypted data; most proposals use homomorphic encryption (HE) and secure multiparty computation (MPC) [2]. Even though service providers cannot directly access the native content of the encrypted signals, they can still apply HE and MPC. In particular, fully homomorphic encryption (FHE) allows arbitrary computation on en- crypted data [3]. However, these methods impose high communication costs, high computation complexity or large cipher text size, so further advances are needed for attractive applications such as big data analysis and advanced image/video processing. We take the random unitary transform approach as we focus on secure image processing [4]. Random unitary transform based encryp- tion methods have lower communication costs, lower computation complexity or small cipher text size. We continue to study secure sparse coding for pattern recognition [5]-[8],

Natarajan Meghanathan et al. (Eds): SIPP, BIGML, DaKM, SOEN, AISC - 2020
pp. 01-12, 2020. © CS & IT-CSCP 2020 DOI: 10.5121/csit.2020.100201

Encryption- then-Compression (EtC) systems [9]-[11]. Orthogonal Matching Pursuit (OMP), a sparse coding algorithm, is executed in the encrypted signal domain.

Early work on sparse coding was based on the efficient coding hypothesis, which states that the goal of visual coding is to faithfully reproduce the visual input while minimizing the neural effort [12]. It effectively represents observed signals as the linear combination of a small number of atoms. Sparse dictionary learning has been successfully applied to various image/video and audio processing applications [13]-[16]. The effectiveness of sparse coding has been reported for pattern recognition [15], image compression [16]. For example, the experiments of Ref. [16] show that rate-distortion based sparse coding outperforms JPEG and JPEG2000 by up to 6+ dB and 2+ dB, respectively.

In this paper, we propose a privacy-preserving pattern recognition scheme that extends previously proposed EtC methods [9]-[11]. The secure pattern recognition methods and EtC systems mentioned above were proposed separately. This current proposal offers not only image pattern recognition but also image compression. The integrated system is realized by performing pattern recognition in the secure compressed domain. 1) It is capable of efficient pattern recognition in the encrypted image domain. Even if data leaks, privacy is maintained because the data remains encrypted. 2) It works as an EtC system. Pattern recognition and image compression can be carried out seamlessly in the same compressed signal domain. This means that the proposed secure OMP algorithm chooses the atoms sequentially and then calculates the sparse coefficients. Pattern recognition employs the few sparse coefficients. Based on the pattern recognition result, additional atoms are chosen and used to compress the selected images. Finally, we employ the INRIA person dataset to evaluate the human detection performance of the proposed method [17]. Detecting humans in images is essential for not only image/video surveillance but also many applications such as automatic driver assistance, etc.

The organization of this paper is as follows. In Sec. 2, we explain related work. Section 3 describes sparse coding for image modeling. In Sec. 4, we propose secure sparse coding for secure sparse coding for pattern recognition with image compression. Section 5 shows simulation results. Conclusions and future work are given in Sec. 5.

2. RELATED WORK

In this section, we review the conventional secure pattern recognition methods and Encryption-then-Compression (EtC) systems.

2.1. Secure Pattern Recognition

We have proposed secure sparse coding for pattern recognition [5]-[8]. Feeding the encrypted images into the secure OMP computation yields the sparse coefficients used for pattern recognition. We verified that by adopting the random unitary transform, the pattern recognition performance is not degraded, which proves that the proposed framework operates securely with no performance degradation. Furthermore, compared with deep-learning based methods such as SPCANet [18], the sparse coding based method has several prominent advantages such as 1) low computational complexity and less data needed for training, 2) transparent machine learning: the algorithm is interpretable as the optimization problem is written in closed form. Refs. [6][7] detail the experiments and results.

2.2. Encryption-then-Compression (EtC) Systems

Encryption-then-Compression (EtC) systems [9]-[11] [19]-[21] have been proposed to securely transmit and compress images through an untrusted channel provider; the traditional technique is to use Compression-then-Encryption (CtE) systems. EtC systems allow us to close non-encrypted

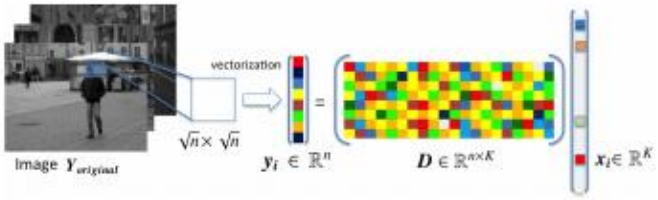


Figure 1: Sparse coding for image patches

images to SNS providers, because encrypted images can be directly compressed even when the images are multiply recompressed by SNS providers. Well-known EtC systems are block scrambling-based encryption schemes that are compatible with international standards, e.g. JPEG, JPEG2000, etc [19]-[21]. While the sparse coding based EtC systems [9]-[11] are not compatible with international compression standards, they do provide high coding performance because they form dictionaries that fit the observed signals.

3. SPARSE CODING FOR IMAGE MODELING

In this section, we overview sparse coding for image modeling which is the basis of secure pattern recognition and EtC systems.

3.1. Sparse Coding for Image Patches

We consider image patches of size $\sqrt{n} \times \sqrt{n}$ pixels that are ordered lexicographically as column vectors $\boldsymbol{y}_i = \{y_1, , ..., y_n\}^T \in \mathbb{R}^n$. The patches are extracted from image $\boldsymbol{Y}_{original}$ as shown in Fig. 1. We assume that every image patch $\boldsymbol{y}_i$ can be represented sparsely given the over-complete dictionary $\boldsymbol{D} = \{\boldsymbol{d}_1, ..., \boldsymbol{d}_K\} \in \mathbb{R}^{n\times K}$ whose columns contain K prototype atoms $\boldsymbol{d}_i$:

$$\boldsymbol{y}_i = \boldsymbol{D}\boldsymbol{x}_i, \tag{1}$$

where $\boldsymbol{x}_i = \{x_1, ..., x_K\}^T \in \mathbb{R}^K$ are sparse coefficients, $i = 1, \cdots, N$, and N is the total number of patches. In advance, dictionary $\boldsymbol{D}$ is designed for the images by training algorithms such as MOD [23] and K-SVD [24].

If n < K and D is a full-rank matrix, an infinite number of solutions to the representation problem are available. The solution with the fewest number of nonzero coefficients is certainly an appealing representation. This sparsest representation is the solution given by

$$(P_0) \quad \min_{\boldsymbol{x}_i} \|\boldsymbol{x}_i\|_0 \quad \text{subject to} \quad \boldsymbol{y}_i = \boldsymbol{D}\boldsymbol{x}_i, \tag{2}$$

where $\|\cdot\|_0$ is the l_0-norm, counting the nonzero entries of the vector. Extraction of the sparsest representation is, however, an NP-hard problem [25].

3.2. Selection of Dictionary Atoms

Dictionary atoms are typically estimated by a ”pursuit algorithm” that finds the following approximate solution:

$$\boldsymbol{x}_i = \arg\min_{\boldsymbol{x}_i} \left\|\boldsymbol{y}_i - \boldsymbol{D}\boldsymbol{x}_i\right\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{x}_i\|_0 < \epsilon_i. \tag{3}$$

We assume dictionary **D** is fixed. Well-known pursuit algorithms include Orthogonal Matching Pursuit (**OMP**) [22]. **OMP** is a greedy, step-wise regression algorithm. At each stage, **OMP** selects the dictionary atom having the maximal projection onto the residual signal. After each selection, the representation coefficients w.r.t. the atoms selected so far are found via least-squares search.

3.3. Dictionary Learning

An over-complete dictionary $\boldsymbol{D}$ is designed by adapting its content to fit a given set of images. Given the set $\boldsymbol{Y}=\{\boldsymbol{y}_i\}_{i=1}^N$, we assume that there exists a dictionary, $\boldsymbol{D}$, that can recreate the given images via sparse combinations. The overall mean square error of a representation is given by

$$E = \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_2^2 . \tag{4}$$

MOD (Method of Optimal Direction) [23] and K-SVD (K-Singular Value Decomposition) [24] are well-known dictionary learning algorithms. Assuming that $\boldsymbol{X}=\{\boldsymbol{x}_i\}_{i=1}^N$ is fixed, the MOD algorithm allows us to seek an update to $\boldsymbol{D}$ such that the above error is minimized. Taking the derivative of (4) with respect to $\boldsymbol{D}$, yields

$$\boldsymbol{D} = \arg\min_{\boldsymbol{D}} \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F^2 = \boldsymbol{Y}\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1}. \tag{5}$$

K-SVD is an iterative method that uses singular value decomposition; it alternates between sparse Coding based on the current dictionary and the process of updating the dictionary atoms to better Fit the data.

4. Secure Sparse Coding for Pattern Recognition With Image Compression

In this section, we propose a privacy-preserving pattern recognition system that offers image compression as an integrated component. The integrated system is realized by performing pattern recognition in the secure compressed domain.

4.1. Secure Computation Architecture

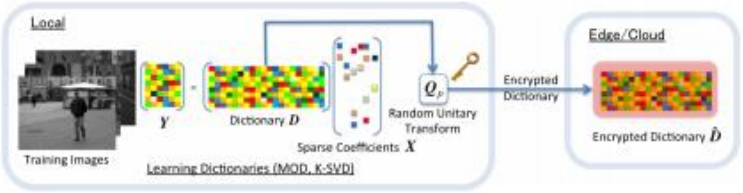
Figure 2 illustrates the architecture of privacy-preserving pattern recognition with image compression. It is based on the sparse coding of image patches. Figure 2(a) shows the training step. Dictionary $\boldsymbol{D}$ is designed by MOD or K-SVD algorithm at the local site. Feeding the training images to the learning algorithm yields dictionary $\boldsymbol{D}$. Next, we apply random unitary transform function $T(\cdot)$ to dictionary $\boldsymbol{D}$ to generate encrypted dictionary $\hat{\boldsymbol{D}}$. Encrypted dictionary $\hat{\boldsymbol{D}}$ is sent to the appropriate edge/cloud site and stored in a database.

Figure 2(b) shows the running step. The local site applies the same random unitary transform function $T(\cdot)$ to test image $\boldsymbol{Y}$ to generate encrypted image $\hat{\boldsymbol{Y}}$. Then encrypted image $\hat{\boldsymbol{Y}}$ is sent to the edge/cloud site. The edge/cloud site uses encrypted image $\hat{\boldsymbol{Y}}$ and encrypted dictionary $\hat{\boldsymbol{D}}$ to perform secure OMP computation. Secure OMP chooses the atoms sequentially and calculates the sparse coefficients $\boldsymbol{X}$ from the encrypted $\hat{\boldsymbol{Y}}$ and $\hat{\boldsymbol{D}}$. At first, pattern recognition is carried out in the compressed signal domain using a few sparse coefficients. $\hat{\boldsymbol{X}}^P$ is the set of the few sparse coefficients used for pattern recognition. Then the images selected by the pattern recognition stage are compressed. For this compression, additional atoms are chosen and calculates the sparse coefficients by secure OMP computation. $\hat{\boldsymbol{X}}^C$ is a set of sparse coefficients used for compression.

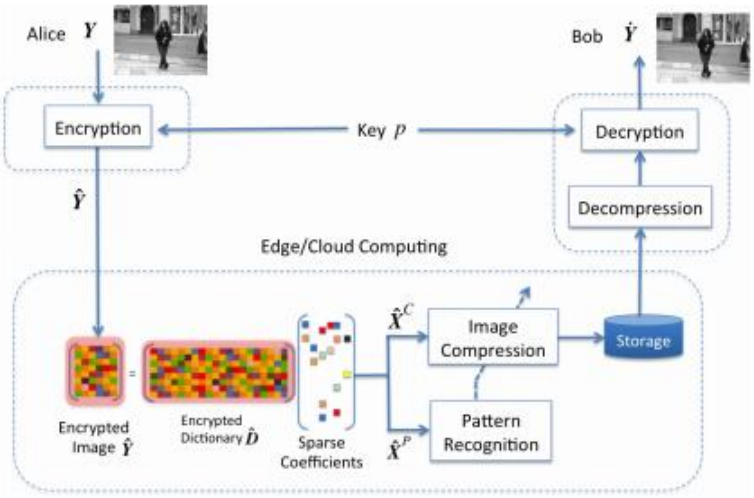
4.2. Random Unitary Transform

The encrypted images and dictionary are generated by using the random unitary transform approach. A vector $\boldsymbol{f}_i$ $(i = 1, \cdots, L) \in \mathbb{R}^N$ is encrypted by a random unitary matrix $\boldsymbol{Q}_p \in \mathbb{C}^{N \times N}$ with a private key p as follows

$$\hat{\boldsymbol{f}}_i = T(\boldsymbol{f}_i, p) = \boldsymbol{Q}_p \boldsymbol{f}_i, \tag{6}$$



(a) Training: generating encrypted dictionary



(b) Running: pattern recognition with image compression

Figure 2: Architecture of privacy-preserving pattern recognition with secure OMP computation.

where $\hat{\boldsymbol{f}}_i$ is an encrypted vector; L is the number of vectors. Note that the random unitary matrix $\boldsymbol{Q}_p$ satisfies

$$\boldsymbol{Q}_p^* \boldsymbol{Q}_p = \boldsymbol{I} \tag{7}$$

where $[\cdot]^*$ and I mean the Hermitian transpose operation and the identity matrix, respectively. In addition to unitarity, Q_p must have randomness for generating the encrypted signal. GramSchmidt orthogonalization is a typical method for generating Q_p. Furthermore, the encrypted vector has the following properties.

· Property 1: Conservation of Euclidean distances.

$$\left\| \boldsymbol{f}_i - \boldsymbol{f}_j \right\|_2^2 = \left\| \hat{\boldsymbol{f}}_i - \hat{\boldsymbol{f}}_j \right\|_2^2 \tag{8}$$

· Property 2: Norm isometry

$$\|\hat{\boldsymbol{f}}_i\|_2^2 = \|\boldsymbol{f}_i\|_2^2 \tag{9}$$

· Property 3: Conservation of inner products.

$$\boldsymbol{f}_i^* \boldsymbol{f}_j = \hat{\boldsymbol{f}}_i^* \hat{\boldsymbol{f}}_j \tag{10}$$

4.3. Secure OMP Computation

The proposed secure sparse coding computation generates encrypted signal $\hat{\boldsymbol{y}}_i$ and dictionary $\hat{\boldsymbol{D}}$ by the following transforms:

$$\hat{\boldsymbol{y}}_i = T(\boldsymbol{y}_i, p) = \boldsymbol{Q}_p \boldsymbol{y}_i \tag{11}$$

$$\hat{\boldsymbol{D}} = T(\boldsymbol{D}, p) = \boldsymbol{Q}_p \boldsymbol{D}. \tag{12}$$

The sparse coefficient $\hat{\boldsymbol{x}}_i$ is estimated for each image patch $\hat{\boldsymbol{y}}_i$. Instead of Eq. (3), we consider the following optimization problem in which $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{D}}$ are assumed to be given:

$$\hat{\boldsymbol{x}}_i = \underset{\boldsymbol{x}}{arg\ min} \| \hat{\boldsymbol{y}}_i - \hat{\boldsymbol{D}}\boldsymbol{x}_i \|_2^2 \quad subject\ to \quad \|\boldsymbol{x}_i\|_0 < \epsilon. \tag{13}$$

The sparse coefficient $\hat{\boldsymbol{x}}_i$ yielded by secure OMP computation is the same result as that created by the non-encrypted version [9]-[11]. The algorithm is shown below (prefix i of $\hat{\boldsymbol{x}}_i$ and $\hat{\boldsymbol{y}}_i$ is omitted for notation simplicity):

Secure OMP Computation Algorithm

Initialization: $k = 0$, and set

· The initial solution $\boldsymbol{x}^0 = \boldsymbol{0}$

· The initial residual $\hat{\boldsymbol{r}}^0 = \hat{\boldsymbol{y}} - \hat{\boldsymbol{D}}\mathrm{x}^0 = \hat{\boldsymbol{y}} = \boldsymbol{Q}_p \boldsymbol{y}$

· The initial solution supports $\boldsymbol{S}^0 = \emptyset$.

Main Iteration:

Increment k by 1 and perform the following steps:

· Sweep: Compute the errors

$$\hat{\epsilon}(i) = \left\|\boldsymbol{r}^{k-1}\right\|_2^2 - \frac{(\boldsymbol{d}_i \cdot \boldsymbol{r}^{k-1})^2}{\|\boldsymbol{d}_i\|_2^2}. \tag{14}$$

· Update Support: Find the minimizer

$$\begin{aligned} i_0 &= \underset{i \notin \boldsymbol{S}^{k-1}}{arg\ min} \{\hat{\epsilon}(i)\} \\ &= \underset{i \notin \boldsymbol{S}^{k-1}}{arg\ min} \{\epsilon(i)\}, \boldsymbol{S}^k = \boldsymbol{S}^{k-1} \cup \{i_0\}. \end{aligned} \tag{15}$$

· Update Provisional Solution: compute

$$\hat{\boldsymbol{x}}^k = \{(\hat{\boldsymbol{D}}_{\boldsymbol{S}^k})^T \hat{\boldsymbol{D}}_{\boldsymbol{S}^k}\}^{-1} \{(\hat{\boldsymbol{D}}_{\boldsymbol{S}^k})^T \hat{\boldsymbol{y}}\}. \tag{16}$$

· Update Residual: compute

$$\hat{\boldsymbol{r}}^k = \boldsymbol{Q}_p \boldsymbol{r}^k. \tag{17}$$

· Stopping Rule:
If $\|\hat{r}^k\|_2 < \epsilon$, stop. Until satisfaction is achieved, commence another iteration. Alternative stopping rule is given by

$$k = T_k, \tag{18}$$

where T_k is the number of specified atoms. Iteration is repeated until the number of selected atoms reaches T_k.

Output: The proposed solution $\hat{x}$ is obtained after k iterations.

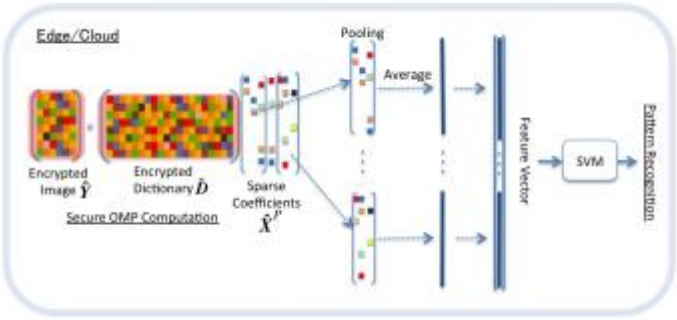


Figure 3: Feature extraction and classification.

4.4. Feature Extraction and Classification

The secure OMP algorithm select atoms sequentially and calculates the corresponding sparse coefficients for each image patch. We use just a few sparse coefficients (calculated using only $k = 1$ *or* 2 iterations) for pattern recognition. Figure 3 shows the procedure of feature extraction and classification. The sparse coefficients at each image patch are used for formatting the feature vector. In order to reduce the dimension of the feature, we take the statistics of the spatially local sparse coefficients of atoms as the feature, which corresponds to local spatial pooling. Multiple sparse coefficients $\boldsymbol{x}_i$, which correspond to local $B \times B$ image patch $\boldsymbol{y}_i$, are grouped into the averaged sparse coefficient $\bar{\boldsymbol{x}}_j (j =, 1, 2, \cdots, N/B^2)$, where B is block size. The averaged sparse coefficients $\bar{\boldsymbol{x}}_j$ are vectorized to produce feature vector $\vec{\boldsymbol{x}}$.

SVM is a supervised machine learning algorithm that can be used for both classification or regression tasks, but it is mostly used for the former. In SVM, we input a feature vector $\vec{\boldsymbol{x}}$ to the discriminant function as

$$(\vec{\boldsymbol{x}}) = sign(\omega^T \vec{\boldsymbol{x}} + b) \tag{19}$$

with

$$sign(u) = \begin{cases} 1(u > 1) \\ -1(u \leq 1), \end{cases} \tag{20}$$

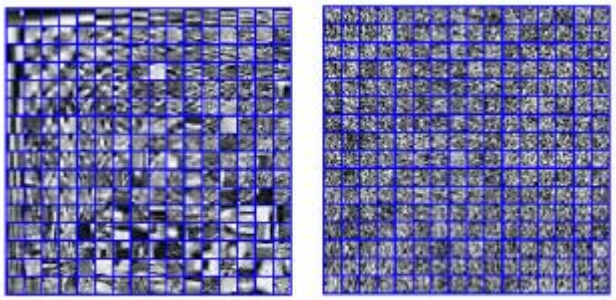
where ω is a weight parameter and b is a bias. SVM also has a technique called the kernel trick, which is a function that takes a low dimensional input space and transforms it into a higher dimensional space. This can be used for non-linear classification. For the pattern recognition task, classification is performed using a linear SVM. The SVM is trained using task data from training subjects.

4.5. Quality Control for Image Compression

Feeding the encrypted dictionary and the encrypted image into the secure OMP computation yields the sparse coefficients $\hat{\boldsymbol{x}}_i$ for each image patch $\boldsymbol{y}_i$. The decoded image $\dot{\boldsymbol{y}}_i$ can be obtained by $\dot{\boldsymbol{y}}_i = \boldsymbol{Q}_p^* \hat{\boldsymbol{D}} \hat{\boldsymbol{x}}_i$. This means that the proposed scheme can work as an EtC system. The image quality of the decoded image $\dot{\boldsymbol{y}}_i$ can be controlled by threshold ϵ_i, which determines the stopping condition of the secure OMP algorithm, i.e. $\left\| \boldsymbol{r}_i^k \right\|_2 < \epsilon_i$. In order to keep the image quality of each image patch, the same threshold is set: $\epsilon_i = \text{constant}$ $(i = 1, \cdots, N)$. An alternative stopping rule is $k = T_k$. In this case, the number of atoms in each patch is set to be the same.

5. EXPERIMENTAL RESULTS

We carried out experiments on detecting humans in images from the INRIA person dataset [17]. Here we assume that we compress only those that include human(s) captured by surveillance systems.

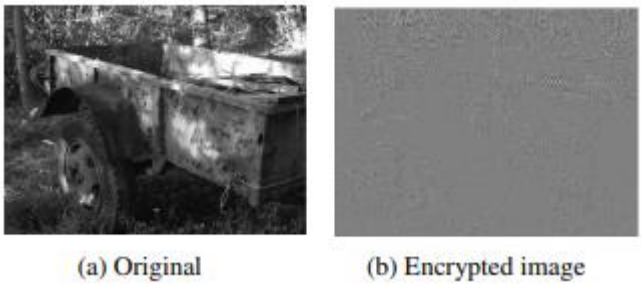


(a) Dictionary $\boldsymbol{D}$ (b) Encrypted dictionary $\hat{\boldsymbol{D}}$

Figure 4: A trained dictionary and corresponding encrypted dictionary for human images.

(a) Original (b) Encrypted image

Figure 5: A sample of original and encrypted human images.



(a) Original (b) Encrypted image

Figure 6: A sample of original and encrypted non-human images.

5.1. INRIA Person Dataset and Parameters

The INRIA person dataset is one of the most popular and widely used pedestrian detection benchmark datasets. The INRIA person dataset contains images of various sizes with and without humans. We evaluated the performance of the proposed method by challenging it with 480 × 640 pixels human and non-human images. The parameters settings are as follows:

1) Designing K-SVD: We applied K-SVD and trained a dictionary of size 64 × 256. The training data consisted of a set of image patches of size 8 × 8 pixels, randomly taken from 20 human images.
2) Creating the random unitary transform: We generated a 64 × 64 random unitary transform by the Gram-Schmidt orthogonalization method.
3) Designing and running the SVM: block size B=20 for local pooling of the sparse coefficients. For the human detection task, two-class classification is performed using a linear SVM. In the training step, the SVM is trained using 100 images (50 human images and 50 non-human images).

In the evaluation, we used 10-fold cross-validation. 100 images were partitioned into 10 subsamples (a single sub-sample contains 5 human and 5 non-human images). Of the10 sub-samples, a single sub-sample is retained as the validation data for testing, and the remaining 9 subsamples

Table 1: Detection Rate (DR) [%] of the proposed method.

(a) Number of atoms: $L = 1$

Test	1	2	3	4	5	6	7	8	9	10	Ave.
DR	100	70	80	70	90	90	80	60	90	70	80

(b) Number of atoms: $L = 5$

Test	1	2	3	4	5	6	7	8	9	10	Ave.
DR	90	60	90	70	90	90	80	50	100	70	79

Table 2: Detection Rate (DR) [%] of the non-encrypted method.

(a) Number of atoms: $L = 1$

Test	1	2	3	4	5	6	7	8	9	10	Ave.
DR	100	70	80	70	90	90	80	60	90	70	80

(b) Number of atoms: $L = 5$

Test	1	2	3	4	5	6	7	8	9	10	Ave.
DR	90	60	90	70	90	90	80	50	100	70	79

are used as training data. The cross-validation process is then repeated 10 times, with each of the 10 subsamples used exactly once as the validation data. The 10 results were then averaged to produce a single estimate.

5.2. Results

The trained dictionary and corresponding encrypted dictionary are shown in Fig. 4. Figures 5 and 6 show the original and corresponding encrypted images for a sample of human and non-human images, respectively. Feeding the encrypted dictionary and the encrypted images into the secure OMP computation yielded the sparse coefficients $\hat{x}_i$ for each image patch y_i.

Detection rate of the proposed privacy-preserving pattern recognition method is shown in Table 1. We evaluated two cases: the number of atoms L = 1 and L = 5. Detection rate is calculated by

$$Detection\ rate = \frac{Number\ of\ images\ correctly\ detected}{Total\ number\ of\ test\ images}. \quad (21)$$

Table 1 shows that the proposed method achieves a detection rate of around 80 [%]. Note that the results were obtained from encrypted images. Setting the number of atoms at L = 1 or L = 5 yielded almost the same performance. For comparison, we evaluated a pattern recognition method with the input being the non-encrypted version of OMP. Detection rate of the non-encrypted version is shown in Table 2. The 10-fold cross-validation used the same training and testing datasets for non-encrypted version of OMP and the secure OMP. The results show that the proposal has exactly the same detection performance as the non-encrypted version of the pattern recognition method.

Figure 7 plots coding efficiency (number of atoms vs. decoded image quality PSNR [dB]) for the selected human images. We controlled the image quality of the human images at each patch by setting number of atoms L = {1, 2, 3, 4, 5}. This figure shows that proposed method increases decoded image quality by adding the atoms sequentially. Note that there is no need to decompress and decrypt images when running the secure OMP algorithm.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed privacy-preserving pattern recognition with image compression. The pattern recognition can be carried out in the compressed signal domain. It can efficiently compress the images selected by the pattern recognition stage. We confirmed its performance by

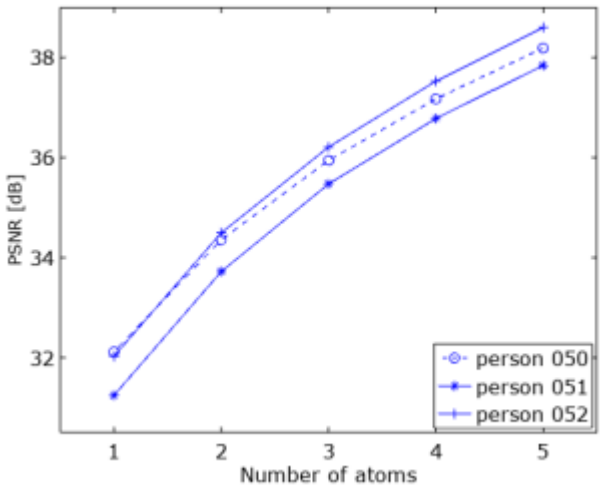


Figure 7: Coding efficiency (Number of atoms L vs. decoded image quality).

detecting humans in the INRIA dataset. In terms of estimation accuracy for pattern recognition, these experiments are merely the first step. Further study is required to enhance the proposal's performance.

REFERENCES

[1] C. T. Huang, L. Huang, Z. Qin, H. Yuan, L. Zhou, V. Varad-harajan, and C-C. J. Kuo,"Survey on securing data storage in the cloud,"APSIPA Transactions on Signal and Information Processing, vol. 3, e7, 2014.

[2] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," IEEE Signal Processing Magazine, vol. 30, no. 1, pp. 82-105, Jan. 2013.

[3] Z. Brakerski, "Fundamentals of fully homomorphic encryption - A survey," Electronic Colloquium on Computational Complexity, report no. 125, 2018.

[4] I. Nakamura, Y. Tonomura, and H. Kiya,"Unitary transform-based template protection and its application to l2-norm minimization problems,"IEICE Transactions on Information and Systems, vol. E99-D, no.1, pp. 60-68, Jan. 2016.

[5] Y. Wang, T. Nakachi, and H. Ishihara, "Edge and cloud-aided secure sparse representation for face recognition," 27th European Signal Processing Conference (EUSIPCO 2019), Sep. 2019.

[6] Y. Wang and T. Nakachi, "Towards secured and transparent AI technologies in hierarchical computing networks," NTT Technical Review, <https://www.nttreview.jp/archive/2019/201909.html>, vol. 9, 2019.

[7] Y. Wang and T. Nakachi, "Secure face recognition in edge and cloud networks: from the ensemble learning perspective," IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2020), to be presented.

[8] T. Nakachi, Y. Wang, and H. Kiya, "Privacy-preserving pattern recognition using encrypted sparse representations in L0 norm minimization," IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2020), to be presented.

[9] T. Nakachi and H. Kiya, "Practical secure OMP computation and its application to image modeling," Proceedings of the 2018 International Conference on Information Hiding and Image Processing (IHIP2018), Sep. 2018.

[10] T. Nakachi, Y. Bandoh, and H. Kiya, "Secure dictionary learning for sparse representation," 27th European Signal Processing Conference (EUSIPCO 2019), Sep. 2019.

[11] T. Nakachi and H. Kiya, "Secure sparse representations in L0 norm minimization and its application to EtC systems," 13th International Conference on Signal Processing and Communication Systems (ICSPCS2019), d13, pp. 61-67, Dec. 2019.

[12] H. B. Barlow, "Possible principles underlying the transformation of sensory messages," Sensory Communication, pp. 217-234,1961.

[13] M. Elad, "Sparse and redundant representations: from theory to applications in signal and image processing," Springer, 2010.

[14] M. Elad, " Sparse and redundant representation modeling - what next?, "IEEE Signal Processing Letters, vol. 19, no. 12, pp. 922-928, Dec. 2012.

[15] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: learning a discriminative dictionary for recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 11, pp. 2651-2664, Nov. 2013.

[16] X. Zhang, W. Lin, Y. Zhang, S. Wang, S. Ma, L. Duan, and W. Gao, ”Rate-distortion optimized sparse coding with ordered dictionary for image set compression,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 12, pp. 3387-3397, Dec. 2018.

[17] ”INRIA Person Dataset,” http://pascal.inrialpes.fr/data/ human/.

[18] L. Tian, C. Fan, Y. Ming, and Y. Jin,“ Stacked PCA network (SPCANet): an effective deep learning for face recognition,”2015 IEEE International Conference on Digital Signal Processing (DSP), pp. 1039-1043, Jul. 2015.

[19] K. Kurihara, M. Kikuchi, S. Imaizumi, S. Shiota, and H. Kiya, ”An encryption-thencompression system for JPEG/Motion JPEG standard,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, E98-A, no. 11, pp. 2238- 2245, 2015.

[20] W. Sirichotedumrong and H. Kiya,“ Grayscale-based block scrambling image encryption using YCbCr color space for Encryption-then-Compression systems, ”APSIPA Trans. Signal and Information Processing, vol. 8, no. E7, February 2019.

[21] T. Chuman, W. Sirichotedumrong, and H. Kiya, ”Encryption-then-compression systems using grayscale-based image encryption for JPEG images,” IEEE Transactions on Information Forensics and Security, vol. 14, no. 6,, pp. 1515-1525, June 2019.

[22] Y. C. Pati, R. Rezaiifar, Y. C. P. R. Rezaiifar, and P. S. Krishnaprasad,“Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition,” Proceedings of 27th Asilomar Conference on Signals, Systems and Computers, pp. 40-44, 1993.

[23] K. Engan, S. O. Aase, and J. Hakon Husoy, “ Method of optimal directions for frame design,”1999 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP1999), pp. 2443-2446, 1999.

[24] M. Aharon, M. Elad, and A. Bruckstein,“ K-SVD: An algorithm for designing overcomplete dictionary for sparse representation, ”IEEE Transactions on Signal Processing, vol. 54, no. 11, pp. 4311-4322, Nov. 2006.

[25] B. K. Natarajan,“Sparse approximate solutions to linear systems,”SIAM Journal on Computing, 24, 2, pp. 227-234, 1995.

AUTHORS

Takayuki Nakachi received the Ph.D. degree in electrical engineering from Keio University, Tokyo, Japan, in 1997. Since joining Nippon Telegraph and Telephone Corporation (NTT) in 1997, he has been engaged in research on super-high-definition image/video coding and media transport technologies. From 2006 to 2007, he was a visiting scientist at Stanford University. Dr. Nakachi is a member of the Institute of Electrical and Electronics Engineers the Institute of Electronics (IEEE) and the Information and Communication Engineers (IEICE) of Japan.

Hitoshi Kiya received his B.E and M.E. degrees from Nagaoka University of Technology, in 1980 and 1982, respectively, and his Dr. Eng. degree from Tokyo Metropolitan University in 1987. In 1982, he joined Tokyo Metropolitan University, where he became a Full Professor in 2000. From 1995 to 1996, he attended the University of Sydney, Australia as a Visiting Fellow. He is a Fellow of IEEE, IEICE and ITE. He currently serves as President-Elect of APSIPA, and he served as Inaugural Vice President (Technical Activities) of APSIPA from 2009 to 2013, and as Regional Director-at-Large for Region 10 of the IEEE Signal Processing Society from 2016 to 2017.

LOCAL GRAY WORLD METHOD FOR SINGLE IMAGE DEHAZING

Vedran Stipetić and Sven Lončarić

Faculty of Electrical Engineering and Computing,
University of Zagreb, Zagreb, Croatia

ABSTRACT

Images taken outdoors are often degraded by atmospheric conditions such as fog and haze. These degradations can reduce contrast, blur edges, and reduce saturation of images. In this paper we propose a new method for single image dehazing. The method is based on an idea from color constancy called the gray world assumption. This assumption states that the average values of each channel in a picture are the same. Using this assumption and a haze degradation model we can quickly and accurately estimate the haze thickness and recover a haze free image. The proposed method is validated on a synthetic and natural image dataset and compared to other methods. The experimental results have shown that the proposed method provides comparable results to other dehazing methods.

KEYWORDS

image restoration, image dehazing

1. INTRODUCTION

Images taken outdoors are often degraded by atmospheric conditions such as fog and haze. These degradations can reduce contrast, blur edges, and reduce saturation of images. As a result, many computer vision algorithms work worse in hazy conditions than in optimal weather conditions. For this reason the problem of image dehazing is an important one. Since in many real life applications it is only possible to have a single hazy image of an area, without haze free reference image, depth information, multiple images taken with different polarizations or similar additional information, there has been much effort in developing algorithms capable of removing haze from a single image.

2. PRIOR WORK

Most single image dehazing models are based on the physical description of how haze degrades an image. This description is derived for instance in [1]. It is given as

$$I^c = I_0^c t + (1 - t)A. \qquad (1)$$

Where c denotes the channel, so c is either R, G or B. I denotes the intensity in that channel in the observed image, I_0 denotes the intensity of the non-degraded image, A denotes the airlight, or the illumination dispersed by the haze and t is the transmission map that balances between scene radiance and airlight. Common assumptions made about the airlight A are that it is constant on the entire image and sometimes that it is the same in all three channels. The transmission map depends on the depth of the scene and haze density and is given by the formula

Natarajan Meghanathan et al. (Eds): SIPP, BIGML, DaKM, SOEN, AISC - 2020
pp. 13-20, 2020. © CS & IT-CSCP 2020 DOI: 10.5121/csit.2020.100202

$$t = \int e^{\beta d(x)} dx. \qquad (2)$$

Here the scene depth is denoted by d(x) and haze density by β. Mathematically this results in a system of 3 equations with 5 unknowns, the intensities of each channel, airlight and transmission, for each pixel. In order to solve this underdetermined system, it is necessary to introduce some prior knowledge. The most common prior used in dehazing is the *dark channel prior* first introduced by He et al in [2]. This prior assumes that in any local patch of a natural image, there exists a pixel with very low intensity in at least one of its channels. Then, in a hazy image the brightness of this dark channel in a patch must come from the haze, and so can be used to estimate transmission. The dark channel prior (sometimes abbreviated to DCP) is also widely used to estimate haze, by taking the most haze opaque pixels in the image. Many other papers built on the dark channel prior approach such as [3] where guided filter is used to smooth out the initial rough transmission estimate.

A different approach called *color attenuation prior* (sometimes abbreviated to CAP) is introduced in [4]. There, the authors use the fact that as depth increases, the saturation decreases and the intensity increases. So the difference between saturation and value components can be used to estimate the transmission. This is done using a linear model, with coefficients determined using machine learning. Once the transmission map is estimated this way, the pixels estimated to be the deepest in the image can be used to estimate the airlight A.

A third approach is the variational approach where parameters are determined by minimizing a functional derived from different priors. For example in [5] the generalized total variation prior is used, based in the assumption that the transmission map is piecewise smooth. In [6] the prior used was built upon ideas from computational color constancy, namely gray world assumption, which will also be used in this paper.

Although all of these methods have their advantages, they also have their shortcomings. The variational methods are usually slow, so they cannot be used in real time systems. Dark channel prior-based methods often exhibit halo effects near very bright objects in the image and color attenuation prior has trouble with scenes that are not naturally saturated, such as gray roads or other urban areas. Because of this we propose a new method, with the goal of overcoming these shortcomings.

3. PROPOSED METHOD

In this paper we propose a new method for single image dehazing. First, the airlight is estimated using the technique described in [2]. Then a coarse transmission map is estimated using a local version of the gray world prior. The transmission map is then refined using guided image filtering [3].

3.1. Gray World Assumption

The gray world assumption is a well-known prior in the area of computational color constancy. It is usually stated assuming that the average value in each of the channels is the same.

$$\frac{1}{N}\sum R_i = \frac{1}{N}\sum G_i = \frac{1}{N}\sum B_i \ . \qquad (3)$$

Another assumption that is sometimes made when using the gray world prior is that the average value is close to one half. This assumption is also used in [6].

3.2. Estimating Airlight

The method for estimating airlight is taken from the dark channel prior [2], but will also be described here for the sake of completeness. Since it is assumed that airlight has high brightness, it is larger than the dark channel in most pixels. Also, as depth increases, the transmission goes to zero, and so the pixels with large depth are very similar in color to airlight. Because of this, if we take a small number of brightest pixels in the dark channel, they will usually approximate airlight well, as they are the most haze opaque ones. In particular we take the 0.1% brightest pixels in the dark channel and then take the pixel with highest intensity. The intensity of that pixel is set to be the airlight.

3.3. Estimating Transmission

In order to get a coarse estimate of transmission, we will use a local gray world assumption. First we assume that transmission is constant in a small N*N patch around a pixel. Then, on that patch we have

$$\sum(I - A) = \left(\sum(I_0 - A)\right) t. \qquad (4)$$

So, after using the local gray world assumption that the average value of I_0 on that patch is 0.5, we get the expression for transmission

$$t = \frac{\frac{1}{N^2}\sum(I - A)}{\frac{1}{2} - A}. \qquad (5)$$

This can be done for all three channels. Empirically, we have determined that the best way to combine the information from all the different channels is to choose as the value of transmission in that pixel the value calculated from the channel with smallest average in the patch. After the transmission has been estimated in all pixels it is normalized by dividing the entire transmission map by the largest value that appears in the estimation. This is done to avoid physically unreasonable transmission estimates, such as the ones larger than 1.

This estimation is then refined using guided image filtering reduce the blockiness caused by the patch based approach to estimating transmission and avoid halo effects around edges in the restored image.

Once both airlight and transmission are estimated, the dehazed image is reconstructed from the equation:

$$I_0 = \frac{I - A}{t} + A. \qquad (6)$$

For purposes of numerical stability, when t is smaller than 0.05 it is set to 0.05.

4. EXPERIMENTAL RESULTS

In this section we present our experimental results and compare the proposed local gray world method to dark channel prior and color attenuation prior methods. The tests are done on both natural hazy images with unknown ground truth, and on synthetic indoor images with ground truth known. The synthetic images were taken from the RESIDE dataset [7]. All images were first

rescaled to 400*400 pixel format to ensure that patch sizes remain consistent across different images.

4.1. Qualitative Results

Qualitative results are ones judged visually on natural images with no known ground truth. These results are important because they judge the performance of dehazing on real haze degradation, which does not fully follow the simplified model from [1].

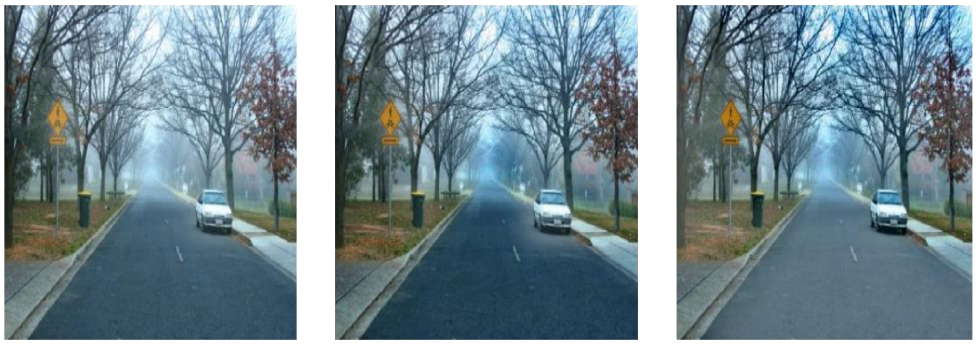


Figure 1. left - result of CAP, middle result of DCP, right result of our method.

In Figure 1 we can see the results of all three algorithms. The proposed local gray world method shows better properties of dehazing in the deeper parts of the image than color attenuation prior and has none of the halo effects caused by white objects in the dark channel prior. The color of the road is also better restored, as it is too dark in both of the other images. Figure 2 shows the original hazy version of the road image, and in Figure 3 we can see the estimated transmission maps of all three algorithms.

In Figure 4 we can see the results when applied to another outdoor image. Once again, the dark channel prior and our local gray world method give good results. The dark channel method again has some haloing issues, and the gray world method has too large contrast near the tops of mountains. This happens because of the sudden jump in color between mountains and the sky where the gray world assumption is not close to being valid.

In Figure 5 we see results applied on a hazy image of a field full of hay bales. In this image the oversaturation effect of our method is especially pronounced. The dark channel prior approach also seems to have removed the haze near the camera much better.

Figure 2. Original road image.

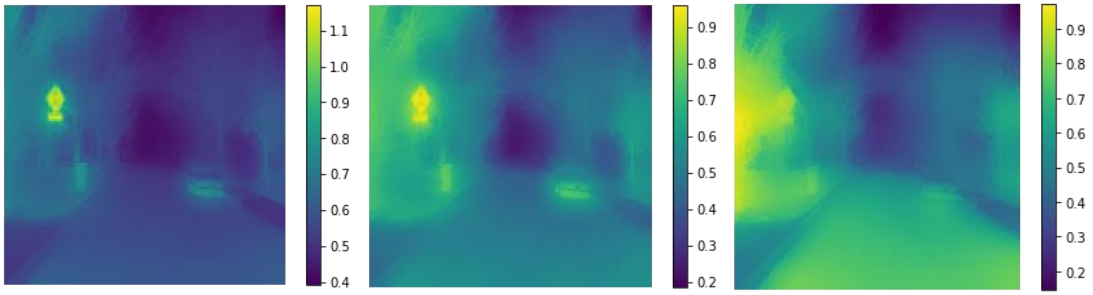


Figure 3. Estimated transmissions: left CAP, middle DCP, right proposed method.

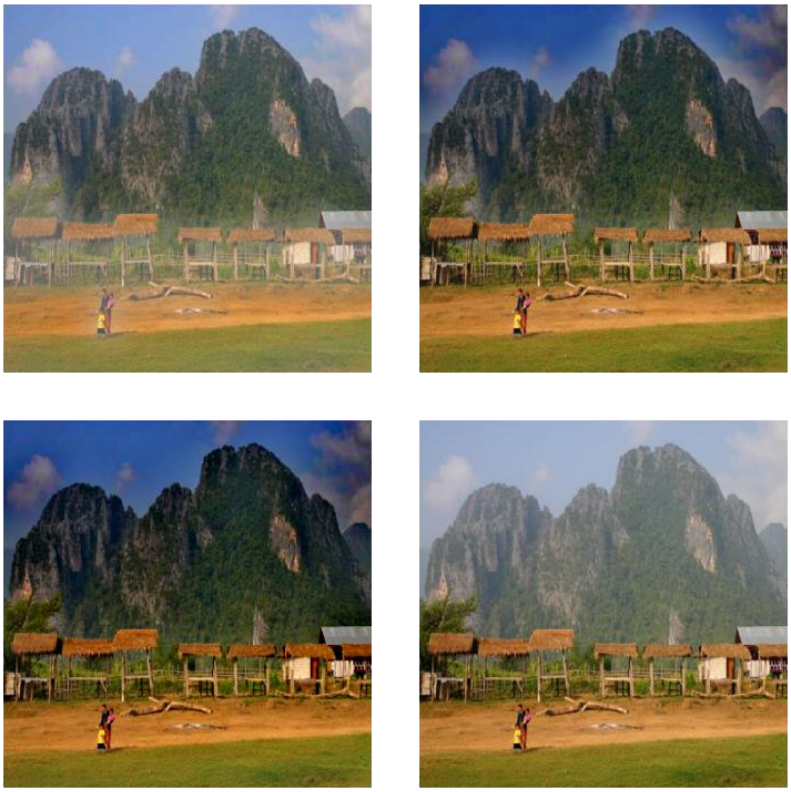


Figure 4. Top left, CAP result, top right DCP result, bottom left local gray world result bottom right original image.

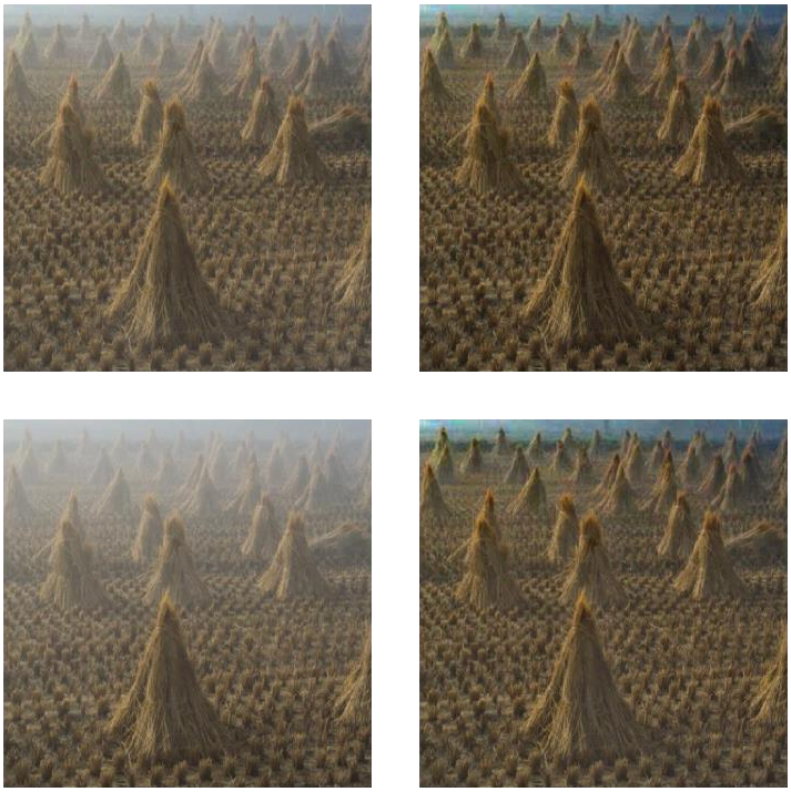


Figure 5. Top left CAP result, top right DCP result, bottom left original hazy image, bottom right local gray world result.

4.2. Quantitative Results

In this section we present quantitative results of dehazing using the proposed local gray world algorithm and take dark channel prior and color attenuation prior as benchmarks to compare against. The images were taken from RESIDE dataset [7] that is commonly used to evaluate dehazing algorithms. For each image and each algorithm we calculated structural similarity index, peak signal to noise ratio and mean square error compared to ground truth. The results are presented in tables 1, 2 and 3. SSIM stands for structural similarity index, PSNR stands for peak signal to noise ratio and MSE stands for mean squared error. We can see that according to all of the metrics on both images, color attenuation prior seems to be the best, while the proposed method seems to be worse than both others on the synthetic data. The actual dehazed images and originals can be seen on Figures 6 to 9. In Table 3 we present the averaged results on 100 randomly chosen images from the RESIDE dataset. We can see that on avarage the dark channel prior seems to be the best according to all metrics. We can see in the Figures that while the local gray world method seem to be good at removing the haze, it also oversaturates colors and increases contrast too much resulting in poor quantitative results.

Table 1. Quantitative evaluation on Hallway image. For MSE lower numbers are better, for SSIM and PSNR higher numbers are better.

Hallway	CAP	DCP	Local gray world
SSIM	0.92591	0.89616	0.88816
PSNR	20.05890	17.37561	16.13917
MSE	0.00986	0. 01829	0.02432

Table 2. Quantitative evaluation on NYU image. For MSE lower numbers are better, for SSIM and PSNR higher numbers are better.

NYU	CAP	DCP	Local gray world
SSIM	0.92528	0.92519	0.89771
PSNR	17.10202	17.68711	16.06112
MSE	0.01948	0.01703	0.02476

Figure 6. Hallway dehazing results. From left to right CAP, DCP and local gray world.

Figure 7. Ground truth (left) and artificially hazy hallway image.

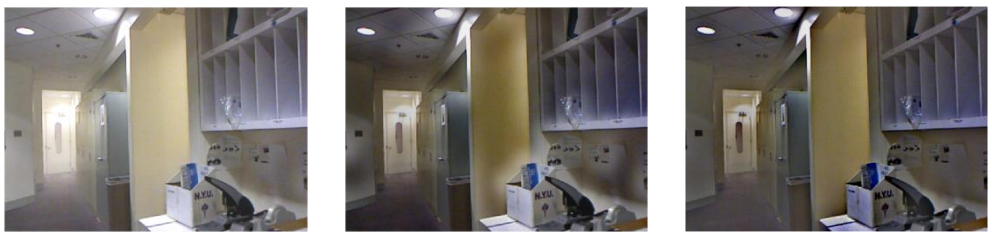


Figure 8. NYU dehazing results. From left to right: CAP, DCP local gray world.

Figure 9. Ground truth (left) and artificially hazy NYU image.

Table 3. Average results for 100 randomly chosen images from the RESIDE indoor dataset. For MSE lower numbers are better, for SSIM and PSNR higher numbers are better.

	CAP	DCP	LOCAL GRAY WORLD
SSIM	0.77925	0.85182	0.76476
PSNR	14.35013	17.61810	13.90976
MSE	0.06335	0.04891	0.09311

5. CONCLUSION AND FUTURE WORK

The proposed local gray world dehazing method presented in this paper performs comparably to benchmark methods in image dehazing on synthetic data and sometimes even better on natural images. As it only uses low level image statistics it is also a fast and computationally efficient method. Since the proposed method uses patches around each pixel, it has the same complexity as other methods discussed in the paper, however, it is slightly faster since it uses fewer operations per patch. The proposed method can be especially useful when an image contains large gray surfaces, such as roads or buildings, as we showed in the example in Figure 1. This leads us to conclude that this is an interesting direction for research. Another potential novelty that the local gray world method brings is the information about transmission in all three channels. In this paper we conformed to the norm and used the same transmission in all three channels, but this could be an interesting direction for future experiments.

REFERENCES

[1] S. K. Nayar and S. G. Narasimhan, "Vision in bad weather", in *Proc. IEEE International Conference on Computer Vision (ICCV), vol. 2. Sep. 1999, pp 820-827*

[2] K. He, J. Sun, X. Tang, "Single image haze removal using dark channel prior" *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 12, pp. 2341-2353, Dec. 2011.*

[3] K. He, J. Sun, X. Tang, "Guided image filtering", *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 6, pp 1397 - 1409, Jun. 2013*

[4] Q. Zhu, J. Mai, L. Shao "A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior", *IEEE Transactions on Image Processing, vol. 24, no. 11, pp. 3522-3533, Nov 2015*

[5] Y. Gu, X. Yang, Y Gao, "A Novel Total Generalized Variation Model for Image Dehazing", *Journal of Mathematical Imaging and Vision,* vol. 61 1329-1341, 2019

[6] A. Galdran, J. Vasquez-Corral, D. Pardo and M. Bertalmio, "Enhanced Variational Image Dehazing", *SIAM Journal of Imaging Sciences,* vol. 8, issue 3, pp 1519-1546, July 2015

[7] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, Z. Wang, "Benchmarking Single-Image Dehazing and Beyond", *IEEE Transactions on Image Processing,* vol. 28, no. 1, pp 492-505, Jan. 2019

MERAMALNET: A DEEP LEARNING CONVOLUTIONAL NEURAL NETWORK FOR BIOACTIVITY PREDICTION IN STRUCTURE-BASED DRUG DISCOVERY

Hentabli Hamza[1], Naomie Salim[1], Maged Nasser[1], Faisal Saeed[2]

[1]Faculty of Computing, Universiti Teknologi Malaysia, Malaysia
[2]College of Computer Science and Engineering,
Taibah University, Medina, Saudi Arabia

ABSTRACT

According to the principle of similar property, structurally similar compounds exhibit very similar properties and, also, similar biological activities. Many researchers have applied this principle to discovering novel drugs, which has led to the emergence of the chemical structure-based activity prediction. Using this technology, it becomes easier to predict the activities of unknown compounds (target) by comparing the unknown target compounds with a group of already known chemical compounds. Thereafter, the researcher assigns the activities of the similar and known compounds to the target compounds. Various Machine Learning (ML) techniques have been used for predicting the activity of the compounds. In this study, the researchers have introduced a novel predictive system, i.e., MaramalNet, which is a convolutional neural network that enables the prediction of molecular bioactivities using a different molecular matrix representation. MaramalNet is a deep learning system which also incorporates the substructure information with regards to the molecule for predicting its activity. The researchers have investigated this novel convolutional network for determining its accuracy during the prediction of the activities for the unknown compounds. This approach was applied to a popular dataset and the performance of this system was compared with three other classical ML algorithms. All experiments indicated that MaramalNet was able to provide an interesting prediction rate (where the highly diverse dataset showed 88.01% accuracy, while a low diversity dataset showed 99% accuracy). Also, MaramalNet was seen to be very effective for the homogeneous datasets but showed a lower performance in the case of the structurally heterogeneous datasets.

KEYWORDS

Bioactive Molecules, Activity prediction model, Convolutional neural network, Deep Learning, biological activities

1. INTRODUCTION

The biological systems function via the physical interactions occurring between the molecules. Hence, it is important to determine the molecular binding for understanding the biological system and discovering novel drugs [1]. The pharmaceutical industries have devoted a lot of effort towards discovering novel drugs. This discovery could improve our quality of life, however, could also lead to many adverse effects [2], [3]. Hence, the pharmaceutical companies must ensure drug safety during the research stage, as the observation of adverse effects during late clinical phases could lead to heavy financial losses. However, despite the development of

Natarajan Meghanathan et al. (Eds): SIPP, BIGML, DaKM, SOEN, AISC - 2020
pp. 21-37, 2020. © CS & IT-CSCP 2020 DOI: 10.5121/csit.2020.100203

computational systems for the past 30 years, they are inaccurate while predicting the molecular binding, and, physical experiments have to be conducted for determining the binding [3], [4].
The accurate molecular binding prediction could decrease the time required for discovering novel treatments, eliminating the toxic molecules in the initial developmental stages and for guiding studies towards medicinal chemistry [5]. Despite the requirement of powerful, but, versatile tools for determining the side effects of the novel drugs, none have been discovered till date. This problem can be solved by implementing computational models which have been obtained using the standard Quantitative Structure-Activity Relationships (QSAR) [6], [7].

In the similarity searching strategy, the activities of unknown compounds (target) are predicted by comparing them with the known chemical compounds. Thereafter, the researcher assigns the activities of similar compounds to the target compounds. Though many of the target prediction techniques have been successful, some problems still exist. Researchers have applied different techniques for predicting different target subsets for the same molecule [8]–[10]. One study [11] used the Multilevel Neighbourhoods of Atoms (MNA) structural descriptor system for activity prediction. MNA of the molecule is generated by the connection table and the table of atoms, representing every compound. Every descriptor possessed a specific integer number based on its dictionary. The molecular similarity was based on the Tanimoto coefficient, and, the compound activities were predicted using the activities of the most similar known compounds.

The popular ML algorithms, using the compound classification method for activity prediction (target), were Binary Kernel Discrimination (BKD) [12], Bayesian inference network for ligand-based virtual Screening [13], Naïve Bayesian Classifier (NBC) [14], Artificial Neural Networks (ANNs) [15] and Support Vector Machines (SVM) [16]. The Bayesian belief network classifier was used for predicting the ligand-based targets and their activities [1]. Here, the researchers introduced a novel approach, MaramalNet (Maramal means ‘predicting’ in Malay), which is a convolutional neural network that predicts the molecular bioactivity using a novel molecular matrix representation. Also, it is a deep learning system which incorporates the molecule’s substructural information for activity prediction.

2. DEFINITION AND RELATED WORK

2.1 Deep Learning

Deep learning is seen to dramatically improve the advanced artificial intelligent tasks such as speech recognition, object detection and machine translation [17]. The deep architectural nature of this technique is useful for solving the complex artificial intelligence-related problems [18]. Hence, researchers have used this technique in modern domains for several tasks like face recognition and object detection. This method has also been applied to many language models. For instance, [17] applied the recurrent neural networks for denoising the speech signals, [19] used the stacked autoencoders for determining the cluster pattern during gene expression. In another study,[20], the researchers used a neural model for generating images having differing styles. Also, [21] used the deep earning technology for a simultaneous analysis of sentiments from the multiple modalities.

The deep learning technology has undergone massive developments during the past few years. Empirical results showed that this technique was better than the other ML algorithms. This could due to the fact that this technique mimics the brain functioning and stacks multiple neural network layers one after another, like the brain model. According to [22], the Deep Learning machines show a better performance than the conventional ML tools as they also include the feature extraction method. However, till date, there exists no theoretical background for the deep learning technology. The deep learning techniques learn the feature hierarchies by using features from the higher hierarchical levels formed by the arrangement of the low-level features. The

learning features present at various abstraction levels allow the system to learn the complex functions which map the input and the resultant output from the data without depending on the human-developed features [22]. In the case of the image recognition systems, the conventional setup extracts the handcrafted features and feeds them to the SVM. However, the deep learning technology shows a better performance as it also optimises all the extracted features.

The biggest difference between the ML and deep learning technologies is their performance variations when the data volume increases. For a smaller dataset, the deep learning method performs inefficiently as it needs a huge data volume for proper understanding [21].

2.2 Convolutional Neural Network

The Convolutional Neural Network (CNN) is a type of deep feed-forward network which can be easily trained and generalised as compared to other networks having connectivity between the adjacent layers [23], [24]. CNN has been successfully used when other neural networks were unpopular, and currently, has been used in the computer vision community.

CNNs are designed for processing data which is in the form of multiple arrays, for instance, a grey-scale image made of 3×2D arrays with different pixel intensities. Various data modalities are presented as multiple arrays, like 1D for sequences and signals, including language; 2D for the audio or image spectrograms; and 3D for the volumetric or video images. The 4 major ideas which enable the CNNs to use the properties of the natural signals are shared weights, local connections, pooling and use of multiple layers [23], [24].

A classic CNN architecture (Figure 1) includes many stages. The initial stages are made of 2 types of layers: i.e., convolutional and pooling layers. The units within the convolutional layer can be organised in the feature maps, wherein every unit is linked to the local patches of the feature maps from the earlier layers through weights known as the filter bank. The output of the local weighted sum is passed through the non-linearity like the ReLU [25]. All the units within the feature map are seen to share one filter bank. The various feature maps within the layer use differing filter banks. This architecture is so composed to serve 2 purposes. Initially, in the case of array data like images, the local group of values are seen to be highly correlated and form distinctive and easily detectable local motifs. Secondly, the local statistics of the images or other signals are seen to be invariant to the location. Hence, if the motif is seen within one section of the image, it can also be present elsewhere. Thus, this network relies on the fact that the units at the different locations share the same weights and can be detected using the similar pattern from the other parts in the array. Mathematically, discrete convolution is the main filtering operation which is applied in the feature maps; hence, it is so named.

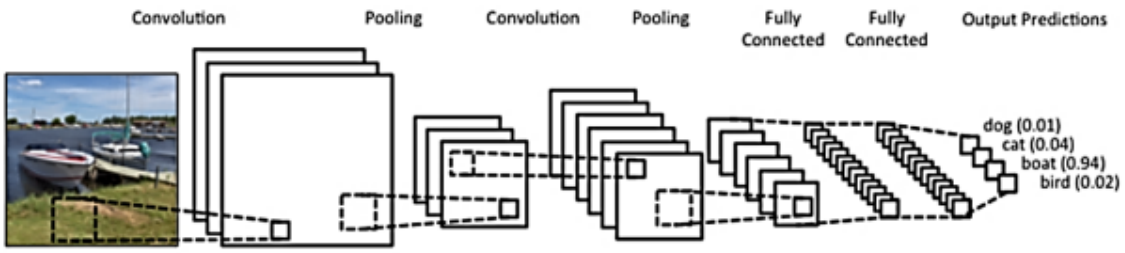


Fig. 1. Architecture of the CNN for Image Classification.

While the convolutional layer detects the local combination of the features based on the earlier layer, the pooling layer merges the semantically similar features into a single feature. Due to the relative position of these features, the motif formation can vary and the reliable detection of the motif is carried out by the coarse-graining of its position in every feature. The general pooling unit can compute a maximal number of the local patch of units into a single feature map.

As described in Figure 2, for the image classification, the CNN technique detects the edges from the raw pixels in Layer 1, and thereafter, uses the edges for detecting the simple shapes in Layer 2. Then, it uses these shapes for detecting the simpler shapes within the Layer 2 and also uses these shapes for determining the high level features, like the face shape in the higher layers. The final layer is the classifier which uses such high-level features [26].

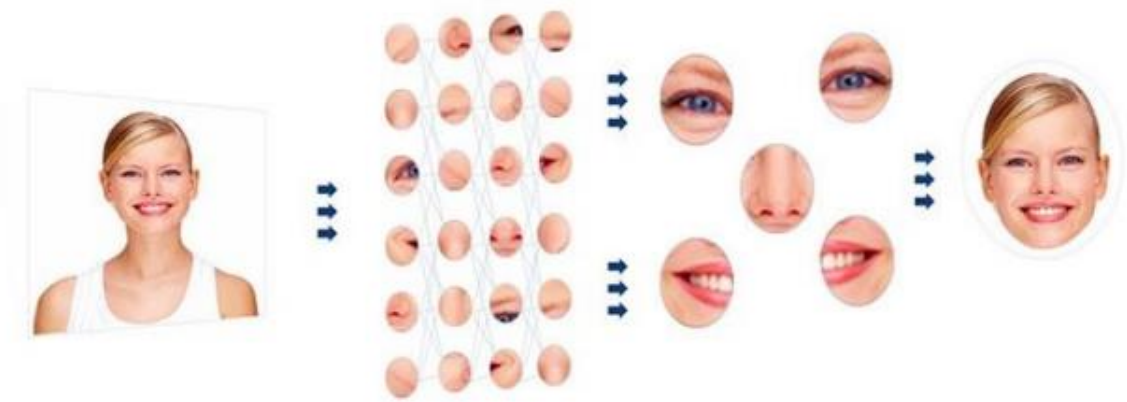


Fig. 2. Eyeris' Deep Learning-based facial feature extraction method based on CNN[26].

2.3 Convolutional Neural Network for the Prediction of the Biological Activities

In [27], the architecture of the Merck Molecular Activity Kaggle Challenge, based on the Multi-Task Deep Neural Network (MT-DNN) [28] showed the best performance. This architecture could train the neural network using multiple output neurons, wherein every neuron predicts the input molecule's activity using different assays. Also, [29]–[31] showed that MT-DNN could be scaled to include large biochemical databases like PubChem Bioassays [32] and ChEMBL [33].

However, there are many limitations associated with the ligand-based processes, like the MT-DNN. Firstly, these techniques are limited to those targets having a lot of prior available data, and hence, they are unable to make predictions for the novel targets. Secondly, the current deep neural networks designed for the ligand-based models also use some molecular fingerprints, like ECFP [34], as their input data. This type of input encoding restricts the feature discovery to the composition of the specific molecular structures that are defined by the fingerprinting procedure [29], [35], which eliminates its capacity to discover the arbitrary features. Thirdly, as these models are blind towards the target, they cannot elucidate the potential molecular interactions.

Another popular strategy used for library designing includes the application of the similarity principle [36], where the structurally similar compounds exhibit similar biological properties. But, researchers [37] have shown that such an empirical guideline is often unsuccessful, as the minor structural modifications could diminish the pharmacological activities of the ligand which is used for describing the molecular similarity within the substructures.

For addressing these limitations, the researchers in this study have proposed a novel matrix representation for the chemical compounds, i.e., mol2matrix, which is based on the molecular substructural similarities with a set of other molecules. Thereafter, the similarity values are

determined and the molecules arranged in the matrix. This technique can be used for many deep learning applications like prediction, virtual screening, molecular classification and molecular search. The following sections describe the design and the development of the MaramalNet approach. Also, the researchers have assessed its performance level by conducting several complex experiments which were based on the structure and bioactivity prediction.

3. MATERIALS AND METHODS

Initially, the researchers have described the construction of various experimental benchmarks used for testing the system. Thereafter, they have described the data encoding and Input representation system along with the design of the deep convolutional network.

3.1. Data Sets

Experiments were conducted over the most popular cheminformatics database: the MDL Drug Data Report (MDDR) [38]–[40] which has been used in our previous studies [1], [41]–[44]. This database consisted of 8294 molecules and contains 11 activity classes, which involve structurally homogeneous and heterogeneous actives, as shown in Table 1. Each row in the tables contains an activity class, the number of molecules belonging to the class, and the diversity of the class, which was computed as the mean pairwise Tanimoto similarity calculated across all pairs of molecules in the class with the ECFP4 (extended connectivity).

Table 1. MDDR Activity Classes Data Set

Activity Index	Activity class	Active molecules	Pairwise Similarity
31420	renin inhibitors	1130	0.290
71523	HIV protease inhibitors	750	0.198
37110	thrombin inhibitors	803	0.180
31432	angiotensin II AT1 antagonists	943	0.229
42731	substance P antagonists	1246	0.149
06233	substance P antagonists	752	0.140
06245	5HT reuptake inhibitors	359	0.122
07701	D2 antagonists	395	0.138
06235	5HT1A agonists	827	0.133
78374	protein kinase C inhibitors	453	0.120
78331	cyclooxygenase inhibitors	636	0.108

3.2. Input Representation

The feature extraction step is very important for analysing the data in the ML and NLP processes. This step helps in determining the interpretable data representation for the machines which could improve the performance of these learning algorithms. The application of inappropriate features could decrease the performance of even the best algorithms, whereas simple techniques perform very well if appropriate features are applied. The feature extraction is carried out unsupervised or even manually. In this study, the researchers have proposed an unsupervised distributed representation of the various chemical compounds.

A novel method was proposed called the mol2matrix (molecule to matrix). This technique could be used in cheminformatics for many problems related to the virtual screening, classification, biological activity prediction, similarity measurements and a substructure search of the

molecules. Here, every compound was embedded within an $n \times n$ matrix, which characterised the various molecular properties.

The distributed representation was seen to be a successful and popular ML approach [46], [47]. This approach involved encoding and storage of information within the system by interacting with the other compounds. The distributed representation technique was inspired by the human memory structure, wherein all memories are stored in a "content-addressable" manner. The content-based storage efficiently recalls all memories based on their partial description. Since these content-addressable thoughts and their properties are stored in a close proximity, the systems possess a viable infrastructure for generalising the features for any item.

The continuous vector representation, which acts like a distributed representation of words, was used in the Natural Language Processing (NLP) system for efficiently representing the semantic/syntactic units having multiple applications. In the model, every word was embedded with the vector in the *n-dimensional* space. The similar words had closer vectors, like "King, Queen" and "Woman, Man", wherein the similarity was based on the syntax and semantics. These vectors were trained based on the idea that the meaning behind the words was characterised by their context, i.e., neighbouring words. Hence, the various words along with their context were considered as the positive training samples [45]. They observed very interesting patterns by training the word vectors with the Skip-gram in the natural language. The words, having a similar vector representation, exhibit multiple similarity degrees. For example, Figure 3 shows that the words $\overrightarrow{King} - \overrightarrow{Man} + \overrightarrow{Woman}$ resemble their closest vector with the word $\overrightarrow{Queen}$ [46].

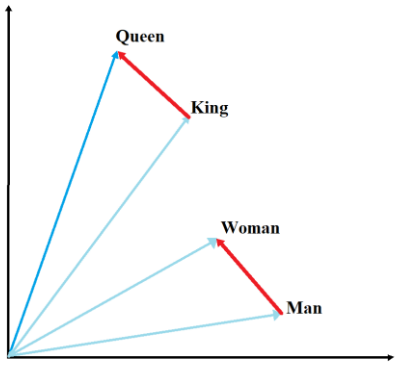


Fig. 3. Word2Vec wherein the words with similar vector representations display multiple similarity degrees.

Deep learning possesses the ability for constructing abstract features and this helps in predicting toxicity or biological activities. Here, the researchers have determined new chemical compound patterns for facilitating their biochemical and biophysical interpretation. Mol2matrix showed a similar molecular representation as the images represented by the Deep Learning technique.

The biological activity of a compound is an adverse property which affects its potential of becoming being marketed as a drug. The toxic or biological properties of a compound are based on their chemical structure, particularly, their substructures, which are identified as functional groups or toxicophores. Many toxicophores have been identified and described earlier [47]–[50].

In this study, the researchers predicted the biological activities using the molecules' distributed representation. This approach included the encoding and storage of information regarding the chemical compounds by establishing their interactions and similarities to the standard

toxicophores. With this in mind, the researchers assessed the similarities of every compound with the known 4096 toxicophore features, i.e., substructural patterns that represented the functional groups reported earlier [51].

The Kazius dataset (Figure 4) comprises of a group of 29 toxicophores, developed from the mutagenicity dataset after applying a novel toxicophore selection and validation criterion. It also consists of statistical, mechanistic and chemical information. These approved toxicophores are used for classifying and predicting the mutagenicity of various compounds in other datasets and display a high accuracy along with good sensitivity and specificity values. The researchers concluded that this set of toxicophores were very helpful in the biological activity prediction of any chemical compound [52]. The Kazius database consists of 4337 compounds, which are converted to ECFP4 by Pipeline Pilot. The first character in the name of the fingerprint, i.e., E, represents the atom abstraction process used for assigning the initial atom code that was based on the number of connections with an atom, the type of element, charge and the atomic mass fingerprints, and is thereafter folded to the final size of 1024 [53].

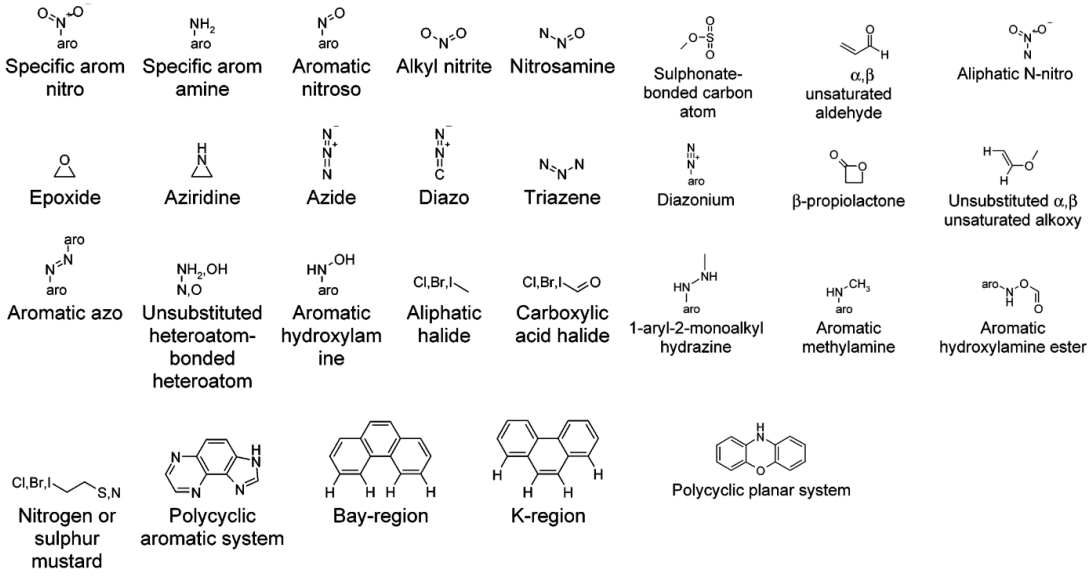


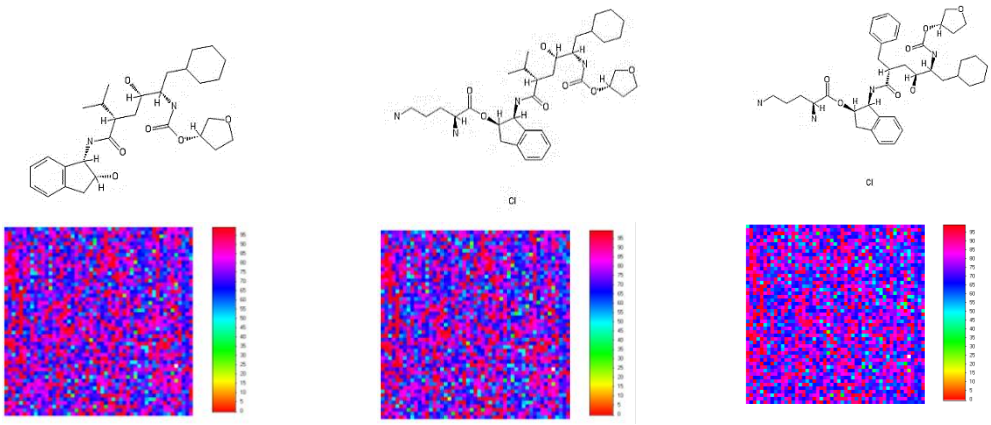
Fig. 4. A set of approved 29 toxicophores within the Kazius Dataset[51].

In this study, the researchers have proposed the mol2matrix for representing every molecule in the 64 × 64 matrix. This matrix comprises of the compounds displaying Tanimoto similarities to the 4096 toxicophore features presented in the Kazius dataset. After eliminating the 241 toxicophore features with the highest similarity, the Tanimoto-based Similarity Searching (TAN) [44] technique applied the binary form of the Tanimoto coefficient to the binary data. A similarity score, S_{xy}, was used for computing the similarities between the 2 molecular ECFP4 fingerprints, i.e., X and Y, with a length of 1024, wherein 'A' represented the number of bits present in the X and Y fingerprints, 'B' represented the number of bits that were present only in X, while 'C' represented the number of bits presents only in Y.

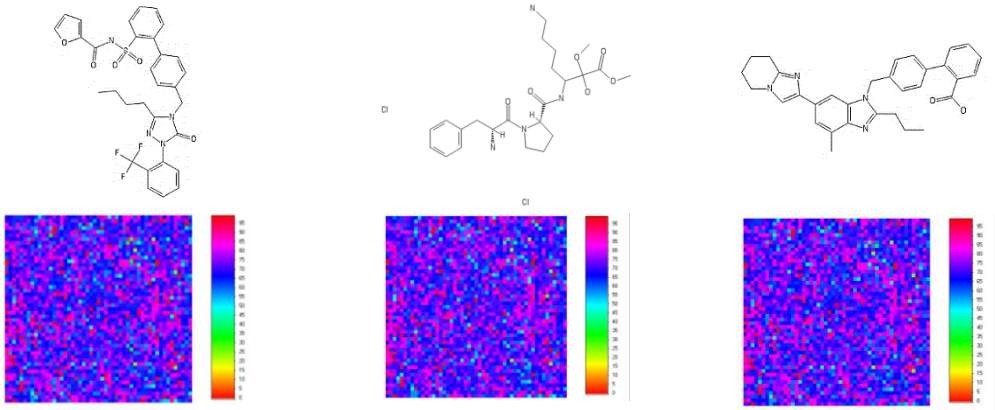
$$S_{xy} = \frac{A}{\mathrm{A} + \mathrm{B} + \mathrm{C}}$$

Every row in the matrix is filled with the molecules based on the order of their toxicophore features after comparing them to the Kazius dataset. This mol2matrix representation helps in visualising and characterising every molecule in the matrix based on their interactions and

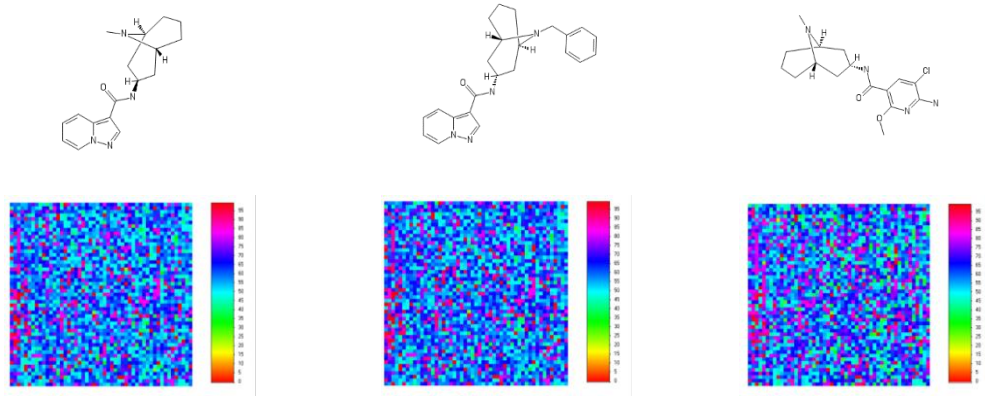
similarities with the functional groups. Thereafter, this matrix describes the toxic properties of the chemical compound. The mol2matrix is a good tool for describing the computational toxicology as it constructs the abstract chemical features. Figure 5 describes the molecules having different biological activities and classed in the MDDR dataset that were used in this study, along with their mol2matrix representation.



a)HIV protease inhibitors activity class



b) Angiotensin II AT1 antagonists activity class



c) 5HT3 antagonist activity class

Fig. 5. Examples describing 9 molecules that were categorised in 3 biological classes of the MDDR datasets and were used in this study along with their mol2matrix representation.

For studying the performance of the mol2matrix representation, the researchers also plotted the scatter graphs using the 8294 molecules which were categorised into 10 different biological

activity classes in the MDDR dataset (Figure 6). These scatter plots are used for determining the relationship between the different molecules within the same class, which was based on their individual representation that was reduced to a 3D structure using the Principal Component Analysis (PCA) technique for representing their features. As seen in the figure, the mol2matrix representation was not overlapping and could be observed easily and thereafter. Also, the biological activities of the molecules could be segregated. This shows that the proposed mol2matrix method can be successfully applied for the molecular representation and the biological activity prediction of different chemical compounds.

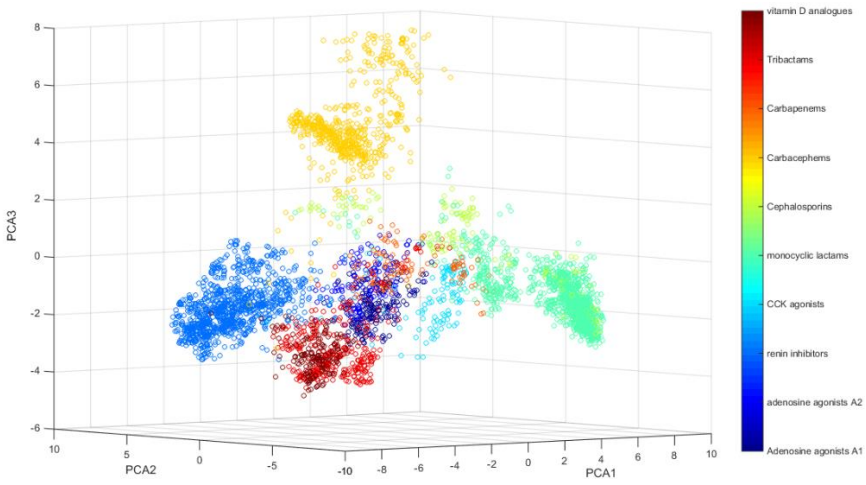


Fig. 6. 3D-scatter plots based on the mol2matrix representation of 8294 different molecules that were selected from the 10 biological activity classes of the MDDR dataset.

3.3. Network Architecture

After collecting all data, the researchers investigated the various model architectures. They considered the convolutional architecture with fully connected layers as the default architecture. Such architecture is appropriate for the multi- and high-dimensional data, like 2D images or genomic data, the researchers designed the MaramalNet layer configurations using the Krizhevsky principles[23] can view the source code through[54]. The configuration followed the generic design described earlier [23]. Fig.7 illustrates the proposed MaramalNetconfiguration.

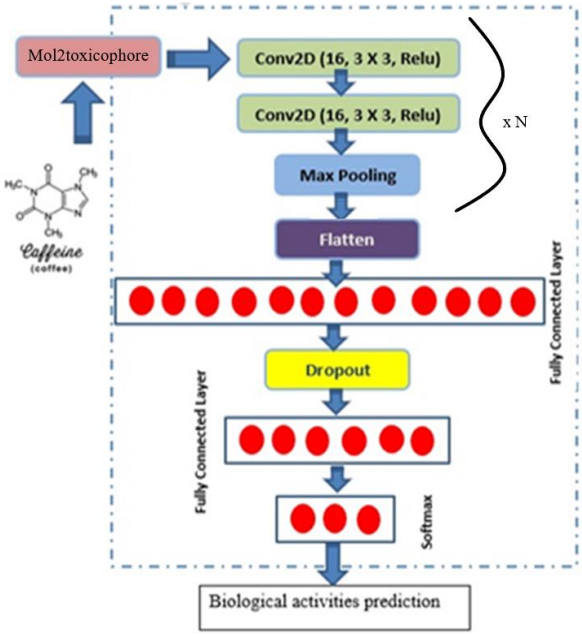


Figure 7 . The proposed MaramalNet configuration

Generally, the target prediction is carried out as follows:
The problem involves predicting if the given chemical compound, i, is active against the target, t. This information is encoded in the binary form, y_{it}, wherein $y_{it} = 1$, for an active compound, and is $y_{it} = 0$, if not. The problem also requires the compound behaviour prediction on *m* targets, simultaneously. During the training stage, a standard backpropagation algorithm is used for determining the CNN and minimising the cross-entropy of the targets and the output layer activation.

3.4. Machine Learning Algorithms

The researchers compared their proposed technique to 3 other available ML algorithms within the WEKA-Workbench [55], i.e., the Naive Bayesian classifier (NaiveB) [56], SVM classifier (known as the LibSVM, LSVM) [57] and a neural network classifier (RBFN) [58]. Determining the ideal classifier parameters is a very tiring process. However, the WEKA-Workbench helps in determining the best probable setup for the LSVM classifier. In this study, the LSVM has been applied to the linear kernel and the values of 0.1, 1.0, and 0.001 have been allocated to the Gamma, Cost, and Epsilon parameters, respectively. The researchers used a supervised discretisation technique for converting the numeric attributes to the nominal attributes in the NaiveB classifier and a minimal standard deviation limit of 0.01 was set in the RBFN classifier. All remaining parameters were kept default for every classifier in the WEKA-Workbench.

4. RESULTS AND DISCUSSION

The proposed code has been implemented in the Theano [59], which is a public deep learning software, based on the Keras [60]. The weights in the neural networks were initialised according to the Keras settings. All layers in the deep network were initialised simultaneously with the ADADELTA [61]. The complete network was trained using the Dell Precision T1700 CPU system with a 14GB memory and the professional-grade NVIDIA-Quadro discrete graphics. The deep network required 2 weeks for its training and testing.

4.1. Evaluation Measures

The researchers used a 10-fold cross-validation technique for validating all results of their proposed MaramalNet system. In this method, they divided the dataset into 10 sections, where 7 sections were used for training, while 3 were used for testing purposes. This procedure was repeated 10 times, hence, all compounds could be used within the test set at least once. Thus, every activity class could be tested against the other classes. Similar to other prediction techniques, the researchers determined the Area Under the receiver operating characteristic Curve (AUC) and used it as the quality criterion for assessing the performances of the various classification algorithms. AUC was estimated as follows:

$$\mathbf{AUC} = (\mathbf{sens} + \mathbf{spec})/\ 2 \quad (1)$$

Wherein sens and spec represent the sensitivity and specificity values, respectively, and are estimated as follows:

$$\mathbf{Sens} = \mathbf{tp}/\ (\mathbf{tp} + \mathbf{fn}) \quad (2)$$

$$\mathbf{Spec} = \mathbf{tn}/\ (\mathbf{tn} + \mathbf{fp}) \quad (3)$$

Wherein tp, tn, fp and fn are the no. of true positive, true negative, false positives, and false negatives, respectively. Where tp are the number of active molecules within the active set, while tn refers to the no. of inactive molecules that are selected in the inactive set. Meanwhile, fp and fn refer to the no. of active molecules present in the inactive set, and the no. of inactive molecules in the active set, respectively. In the model, a curve described the trade-off between the sensitivity and specificity, wherein sensitivity and specificity were defined as the efficiency of the model for identifying the positive and the negative labels, respectively. Furthermore, the Area Under the Curve (AUC) also assesses the model performance. When the AUC value of the prediction algorithm is nearer to 1, it is said to show a better performance.

4.2. Results

In this study, the researchers proposed MaramalNet, which was a novel ligand-based activity prediction or target fishing method for unknown chemical compounds. MaramalNet is a convolutional neural network, having a new molecular matrix representation, and is used for molecular bioactivity prediction. Furthermore, it is a deep learning system which incorporates the substructure information regarding the molecules for making predictions. Hence, the proposed MaramalNet technique was compared to 3 other ML algorithms present in the WEKA-Workbench, i.e., NaiveB, LSVM, and RBFN using optimal parameters.

Table 2 display the Sensitivity, Specificity and the AUC values for the MDDR dataset used in the study. Though a visual inspection of these tables could be used for comparing the prediction accuracy performance of the 4 algorithms, the researchers employed a quantitative technique of one-way ANOVA. This technique quantified the level of agreement observed between the multiple sets which ranked the same group of objects.

Table 2. Sensitivity, Specificity and AUC rates for the Prediction Models using the MDDR dataset.

activity index	MeramalNet			NaïveB			RBFN			LSVM		
	Sens	Spec	AUC	Sens	Spec	AUC	Sens	Spec	AUC	Sens	Spec	AUC
0.99	1	0.99	1	1	1	0.96	0.96	0.96	1	1	1	0.99
0.97	0.99	0.98	1	1	1	0.95	0.97	0.96	1	1	1	0.97
0.95	1	0.98	1	0.99	1	0.44	1	0.72	0.98	1	0.99	0.95
0.96	1	0.98	1	1	1	0.8	1	0.9	1	1	1	0.96
0.98	1	0.99	0.94	1	0.97	0.43	1	0.72	0.99	1	1	0.98
0.92	1	0.96	0.94	1	0.97	0.53	1	0.76	0.98	1	0.99	0.92
0.92	0.99	0.95	0.84	0.99	0.91	0.78	0.97	0.87	0.96	0.99	0.98	0.92
0.96	1	0.98	0.82	0.99	0.91	0.75	0.97	0.86	0.94	1	0.97	0.96
0.96	0.99	0.98	0.88	0.97	0.92	0.66	0.98	0.82	0.96	0.99	0.98	0.96
0.94	1	0.97	0.65	0.99	0.82	0.74	0.96	0.85	0.91	1	0.95	0.94
0.93	0.98	0.95	0.82	0.94	0.88	0.59	0.96	0.78	0.94	0.98	0.96	0.93

In this study, the researchers have applied the one-way ANOVA technique for evaluating the performance of all the 4 algorithms. Hence, in this case, the MDDR activity classes that were described earlier in Tables 1, were considered to be judges, while the parameters of Sensitivity, Specificity and AUC, which were measured for the different prediction algorithms, were considered to be objects. This test showed an output in the form of the *p-value*, median and the variance. In Figure 8, the researchers have presented the results of the one-way ANOVA test after comparing the sensitivity values for the MeramalNet, NaiveB, RBFN and the LSVM algorithms. A very small *p-value* of 1.16×10^{-3} was observed which clearly indicated the high significance of difference between the algorithms. Furthermore, it could be seen that the MeramalNet algorithm displayed a good sensitivity value of 0.94. A larger variance was noted between the NaiveB and the LSVM ML algorithms, i.e., 0.15 and 0.23, respectively, in comparison to the MeramalNet algorithm. This highlights the diversity in the sensitivity values noted in the algorithms with a variance of 0.049. Meanwhile, those models exhibited an average sensitivity value of 0.90 and 0.74, respectively.

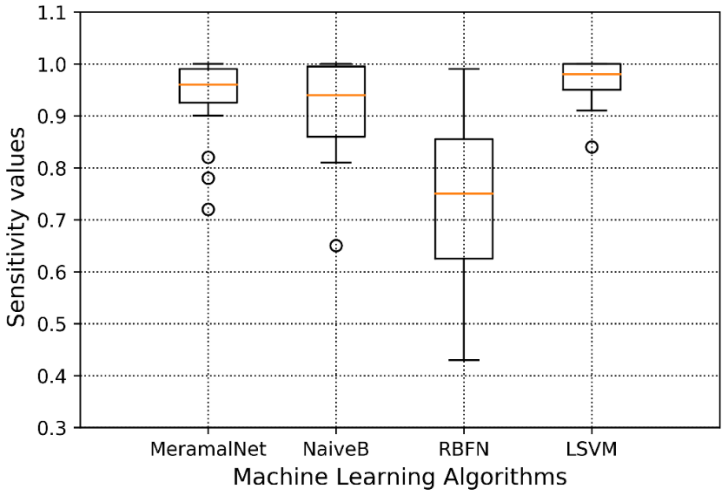


Fig. 8. Comparison of the sensitivity values for the MeramalNet, NaiveB, RBFN and LSVM algorithms using ANOVA

In Figure 9, the researchers have presented the results of the one-way ANOVA test after comparing the specificity values for the MeramalNet, NaiveB, RBFN and the LSVM algorithms. A larger variance was noted between the NaiveB and the RBFN ML algorithms, i.e., 0.01 and 0.04, respectively, in comparison to the MeramalNet algorithm. This highlights the diversity in the specificity noted in the algorithms with a variance of 0.0062. Also, the MeramalNet algorithm displayed a good specificity value of 1.0, while the NaiveB and RBFN algorithms showed a mean specificity value of 0.99 and 0.98, respectively. A small *p-value* of 6.25×10^{-5} was seen which indicated the significance of difference between the algorithms.

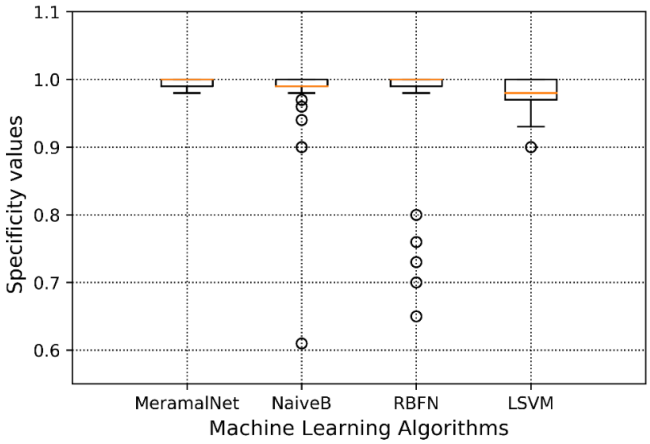


Fig. 9. Comparison of specificity values for the MeramalNet, NaiveB, RBFN and LSVM algorithms using ANOVA

In Figure 10, the researchers have presented the results of the one-way ANOVA test after comparing the AUC values for the MeramalNet, NaiveB, RBFN and the LSVM algorithms. A larger variance was noted between the LSVM, NaiveB and RBFN ML algorithms, i.e., 0.125, 0.083 and 0.033, respectively, in comparison to the MeramalNet algorithm. This highlights the diversity in the AUC values noted in the algorithms with a variance of 0.02. The MeramalNet algorithm displayed a good AUC value of 0.98, while the LSVM, NaiveB and RBFN algorithms showed a mean AUC value of 0.96, 0.99 and 0.85, respectively. A very small *p-value* of 1.6×10^{-14} was seen which indicated the significance of difference between the algorithms.

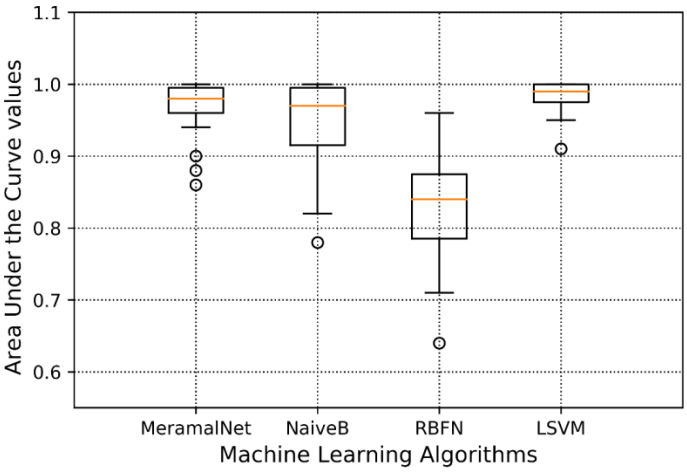


Fig. 10. Comparison of AUC values for the MeramalNet, NaiveB, RBFN and LSVM algorithms using ANOVA

A visual inspection of the one-way ANOVA results (Figures 16-18) indicated that the MaramalNet activity prediction technique was more applicable, convenient and exhibited less severe outliers compared to the NaiveB, RBFN and LSVM ML algorithms, thus, proving the efficacy of the novel prediction approach.

Furthermore, the results presented in Tables 2 for MDDR dataset indicated that this MaramalNet activity prediction technique showed the least variance for the Sensitivity, Specificity and the AUC values for all the activities classes in comparison to the classic NaiveB, RBFN and LSVM ML algorithms, indicating that the deep learning process should be considered as a novel, promising and interesting method for predicting the activities of chemical compounds.

5. CONCLUSION

In this study, the researchers investigated the deep convolutional networks (having up to 9 weight layers) for predicting the activities and for the ligand-based targets. They demonstrated that there was a lower representation depth for the prediction accuracy. They also proposed a novel mol2matrix technique, which was less overlapped and could segregate the biological activities of the molecules. Thereafter, they applied the new MaramalNet technique on the popular datasets and compared their performance with 3 standard ML algorithms. All experiments indicated that the MaramalNet algorithm exhibited interesting prediction rates (where the highly diverse dataset showed 88.01% accuracy, while a low diversity dataset showed 98% accuracy). Furthermore, the experiments also indicated that this novel MaramalNet algorithm showed an effective performance for the homogeneous datasetsbut showed a lower performance against the structurally heterogeneous datasets. Hence, the researchers have presented MaramalNet as a stable and convenient activity prediction approach for the unknown target chemical compounds. However, this area still needs to be explored further and better accuracy prediction techniques have to be developed for the highly diverse activity classes.

ACKNOWLEDGMENT

This work is supported by the Ministry of Higher Education (MOHE) and the Research Management Centre (RMC) at the Universiti Teknologi Malaysia (UTM) under the Research University Grant Category (VOT Q.J130000.2528.16H74 and R.J130000.7828.4F985). Also, we would like to thank Prof Sigeru Omatu for his great feedback during the early discussions about conducting this research.

REFERENCES

[1] A. Ammar, L. Valérie, J. Philippe, S. Naomie, and P. Maude, “Prediction of new bioactive molecules using a Bayesian belief network,” J. Chem. Inf. Model., vol. 54, no. 1, pp. 30–36, 2014.

[2] K. Barakat, “Computer-Aided Drug Design,” J. Pharm. Care Heal. Syst., vol. 1, no. 4, pp. 1–2, 2014.

[3] D. de la Iglesia, M. Garcia-Remesal, G. de la Calle, C. Kulikowski, F. Sanz, and V. Maojo, “The impact of computer science in molecular medicine: Enabling high-throughput research,” Curr. Top. Med. Chem., vol. 13, no. 5, pp. 526–575, 2013.

[4] S. Kothiwale, C. Borza, A. Pozzi, and J. Meiler, “Quantitative structure–activity relationship modeling of kinase selectivity profiles,” Molecules, vol. 22, no. 9, pp. 1–11, 2017.

[5] L. Wang et al., “Accurate and reliable prediction of relative ligand binding potency in prospective drug discovery by way of a modern free-energy calculation protocol and force field,” J. Am. Chem. Soc., vol. 137, no. 7, pp. 2695–2703, 2015.

[6] A. Vaidya, S. Jain, S. Jain, A. K. Jain, and R. K. Agrawal, "Quantitative Structure-Activity Relationships: A Novel Approach of Drug Design and Discovery," J. Pharm. Sci. Pharmacol., vol. 1, no. 3, pp. 219–232, 2014.

[7] C. H. Andrade, K. F. M. Pasqualoto, E. I. Ferreira, and A. J. Hopfinger, "4D-QSAR: Perspectives in drug design," Molecules, vol. 15, no. 5, pp. 3281–3294, 2010.

[8] H. Ding, I. Takigawa, H. Mamitsuka, and S. Zhu, "Similarity-based machine learning methods for predicting drug-target interactions: a brief review.," Brief. Bioinform., vol. 15, no. 5, p. bbt056-, 2013.

[9] F. Luan, T. Wang, L. Tang, S. Zhang, and M. Natália Dias Soeiro Cordeiro, "Estimation of the toxicity of different substituted aromatic compounds to the aquatic ciliate tetrahymena pyriformis by QSAR approach," Molecules, vol. 23, no. 5, 2018.

[10] C. F. Lagos, G. F. Segovia, N. Nu ez-Navarro, M. A. Faúndez, and F. C. Zacconi, "Novel FXa inhibitor identification through integration of ligand- and structure-based approaches," Molecules, vol. 22, no. 10, 2017.

[11] D. Filimonov, V. Poroikov, Y. Borodina, and T. Gloriozova, "Chemical Similarity Assessment through Multilevel Neighborhoods of Atoms: Definition and Comparison with the Other Descriptors," J. Chem. Inf. Comput. Sci., vol. 39, no. 4, pp. 666–670, 1999.

[12] P. Willett, D. Wilton, B. Hartzoulakis, R. Tang, J. Ford, and D. Madge, "Prediction of ion channel activity using binary kernel discrimination," J. Chem. Inf. Model., vol. 47, no. 5, pp. 1961–1966, 2007.

[13] B. Chen, C. Mueller, and P. Willett, "Evaluation of a Bayesian inference network for ligand-based virtual screening," J. Cheminform., vol. 1, no. 1, pp. 1–10, 2009.

[14] X. Xia, E. G. Maliski, P. Gallant, and D. Rogers, "Classification of kinase inhibitors using a Bayesian model," J.Med.Chem., vol. 47, pp. 4463–4470, 2004.

[15] D. a Winkler and F. R. Burden, "Application of neural networks to large dataset QSAR, virtual screening, and library design.," Methods Mol. Biol., vol. 201, pp. 325–367, 2002.

[16] K. Kawai, S. Fujishima, and Y. Takahashi, "Predictive Activity Profiling of Drugs by Topological-Fragment-Spectra-Based Support Vector Machines," J. Chem. Inf. Model., vol. 48, no. 6, pp. 1152–1160, 2008.

[17] Y. LeCun, B. Yoshua, and H. Geoffrey, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[18] Y. Bengio, Learning Deep Architectures for AI, vol. 2, no. 1. 2009.

[19] A. Gupta, H. Wang, and M. Ganapathiraju, "Learning structure in gene expression data using deep architectures, with an application to gene clustering," 2015 IEEE Int. Conf. Bioinforma. Biomed., pp. 1328–1335, 2015.

[20] L. a Gatys, A. S. Ecker, and M. Bethge, "A Neural Algorithm of Artistic Style," arXiv Prepr., pp. 1–16, 2015.

[21] H. Wang, A. Meghawat, L.-P. Morency, and E. P. Xing, "Select-Additive Learning: Improving Cross-individual Generalization in Multimodal Sentiment Analysis," vol. 1, 2016.

[22] H. Wang and B. Raj, "On the Origin of Deep Learning," Arxiv, pp. 1–72, 2017.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Adv. Neural Inf. Process. Syst., pp. 1–9, 2012.

[24] T. Sainath, A. R. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," Proc. IEEE Int. Conf. Acoust. Speech Signal Process., pp. 8614–8618, 2013.

[25] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," Proc. 27th Int. Conf. Mach. Learn., no. 3, pp. 807–814, 2010.

[26] Chen, Yu-Hsin and Krishna, Tushar and Emer, Joel and Sze, Vivienne, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," in IEEE International Solid-State Circuits Conference, ISSCC 2016, Digest of Technical Papers, 2016, pp. 262–263.

[27] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structure-activity relationships," J. Chem. Inf. Model., vol. 55, no. 2, pp. 263–274, 2015.

[28] G. E. Dahl, N. Jaitly, and R. Salakhutdinov, "Multi-task Neural Networks for QSAR Predictions," pp. 1–21, 2014.

[29] T. Unterthiner, A. Mayr, G. Klambauer, and S. Hochreiter, "Toxicity Prediction using Deep Learning," 2015.

[30] T. Unterthiner, A. Mayr, G. Klambauer, M. Steijaert, J. K. Wegner, and H. Ceulemans, "Deep Learning as an Opportunity in Virtual Screening," Deep Learn. Represent. Learn. Work. NIPS 2014, pp. 1–9, 2014.

[31] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande, "Massively Multitask Networks for Drug Discovery," no. Icml, 2015.

[32] Y. Wang et al., "PubChem's BioAssay database," Nucleic Acids Res., vol. 40, no. D1, pp. D400--D412, 2011.

[33] A. P. Bento et al., "The ChEMBL bioactivity database: an update," Nucleic Acids Res., vol. 42, no. D1, pp. D1083--D1090, 2014.

[34] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," J. Chem. Inf. Model., vol. 50, no. 5, pp. 742–754, 2010.

[35] D. Dana et al., "Deep Learning in Drug Discovery and Medicine; Scratching the Surface," Molecules, vol. 23, pp. 1–15, 2018.

[36] J. MLA and G. M. Maggiora, "Concepts and Application of Molecular Similarity," Wiley Interdiscip. Rev. Mol. Sci., vol. 50, pp. 376–377, 1990.

[37] Y. C. Martin, J. L. Kofron, and L. M. Traphagen, "Do structurally similar molecules have similar biological activity?," J. Med. Chem., vol. 45, no. 19, pp. 4350–4358, 2002.

[38] "Sci Tegic Accelrys Inc." [Online]. Available: http://accelrys.com/products/collaborative-science/databases/bioactivity-databases/mddr.html.

[39] J. J. Sutherland, L. a. O'Brien, and D. F. Weaver, "Spline-Fitting with a Genetic Algorithm: A Method for Developing Classification Structure-Activity Relationships," J. Chem. Inf. Comput. Sci., vol. 43, no. 6, pp. 1906–1915, 2003.

[40] "Sutherland dataset." [Online]. Available: http://cdb.ics.uci.edu/cgibin/LearningDatasetsWeb.py.

[41] H. Hentabli, S. Naomie, and F. Saeed, "AN ACTIVITY PREDICTION MODEL USING SHAPE-BASED DESCRIPTOR METHOD," J. Teknol., vol. 1, pp. 1–8, 2016.

[42] H. Hentabli, N. Salim, A. Abdo, and F. Saeed, "LINGO-DOSM : LINGO for Descriptors of Outline," Intell. Inf. Database Syst. Springer Berlin Heidelb., pp. 315–324, 2013.

[43] H. Hentabli, N. Salim, A. Abdo, and F. Saeed, "LWDOSM : Language for Writing Descriptors," Adv. Mach. Learn. Technol. Appl. Springer Berlin Heidelb., pp. 247–256, 2012.

[44] H. Hentabli, F. Saeed, A. Abdo, and N. Salim, "A new graph-based molecular descriptor using the canonical representation of the molecule," Sci. World J., vol. 2014, 2014.

[45] T. Mikolove, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," pp. 1–9, 2013.

[46] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," pp. 1–12, 2013.

[47] R. Benigni, A. Giuliani, R. Franke, and A. Gruska, "Quantitative structure-activity relationships of mutagenic and carcinogenic aromatic amines," Chem. Rev., vol. 100, no. 10, pp. 3697–3714, 2000.

[48] P. Ertl, "An algorithm to identify functional groups in organic molecules," J. Cheminform., vol. 9, no. 1, pp. 1–7, 2017.

[49] E. L. Schymanski et al., "Critical Assessment of Small Molecule Identification 2016: automated methods," J. Cheminform., vol. 9, no. 1, pp. 1–21, 2017.

[50] M. He, Q. Yang, A. Norvil, D. Sherris, and H. Gowher, "Characterization of Small Molecules Inhibiting the Pro-Angiogenic Activity of the Zinc Finger Transcription Factor Vezf1," Molecules, vol. 23, no. 7, p. 15, 2018.

[51] J. Kazius, R. McGuire, and R. Bursi, "Derivation and validation of toxicophores for mutagenicity prediction," J. Med. Chem, vol. 48, pp. 312–320, 2005.

[52] K. Hansen et al., "Benchmark data set for in silico prediction of Ames mutagenicity," J. Chem. Inf. Model., vol. 49, no. 9, pp. 2077–2081, 2009.

[53] F. Saeed and N. Salim, "Using soft consensus clustering for combining multiple clusterings of chemical structures," J. Teknol. (Sciences Eng., vol. 63, no. 1, pp. 9–11, 2013.

[54] V. GUPTA, "Image Classification using Convolutional Neural Networks in Keras," 2017. [Online]. Available: https://www.learnopencv.com/image-classification-using-convolutional-neural-networks-in-keras/.

[55] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.

[56] G. H. John and P. Langley, "Estimating Continuous Distributions in Bayesian Classifiers," pp. 338–345, 2013.

[57] C. CHIH-CHUNG, "LIBSVM : A library for support vector machines," ACM Trans. Intell. Syst. Technol., vol. 2, p. 27:1-27:27, 2011.

[58] G. Bugmann, "Normalized Gaussian radial basis function networks," Neurocomputing, vol. 20, no. 1–3, pp. 97–110, 1998.

[59] F. Bastien et al., "Theano: new features and speed improvements," pp. 1–10, 2012.

[60] F. Chollet, "Keras Documentation," Keras.Io, 2015.

[61] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," 2012.

DATA MINING AND MACHINE LEARNING IN EARTH OBSERVATION – AN APPLICATION FOR TRACKING HISTORICAL ALGAL BLOOMS

Alexandria Dominique Farias and Gongling Sun

International Space University, Strasbourg, France

ABSTRACT

The data produced from Earth Observation (EO) satellites has recently become so abundant that manual processing is sometimes no longer an option for analysis. The main challenges for studying this data are its size, its complex nature, a high barrier to entry, and the availability of datasets used for training data. Because of this, there has been a prominent trend in techniques used to automate this process and host the processing in massive online cloud servers. These processes include data mining (DM) and machine learning (ML). The techniques that will be discussed include: clustering, regression, neural networks, and convolutional neural networks (CNN).

This paper will show how some of these techniques are currently being used in the field of earth observation as well as discuss some of the challenges that are currently being faced. Google Earth Engine (GEE) has been chosen as the tool for this study. GEE is currently able to display 40 years of historical satellite imagery, including publicly available datasets such as Landsat, and Sentinel data from Copernicus.

Using EO data from Landsat and GEE as a processing tool, it is possible to classify and discover historical algal blooms over the period of ten years in the Baltic Sea surrounding the Swedish island of Gotland. This paper will show how these technical advancements including the use of a cloud platform enable the processing and analysis of this data in minutes.

KEYWORDS

Earth Observation, Remote Sensing, Satellite Data, Data Mining, Machine Learning, Google Earth Engine, Algal Blooms, Phytoplankton Bloom, Cyanobacteria

1. INTRODUCTION

Earth observation (EO) has become more prominent in the last decade with more satellites in orbit that are capable of observing the Earth every year. The miniaturization of component parts has also enabled a new generation of CubeSats that are also adding to the data gained from remote sensing (RS). As RS and EO are often used interchangeably, it is worth defining RS as the act of viewing, observing and analysing an object from a given distance. This paper will only be addressing RS data that is observing the Earth, specifically satellite imagery.

The technology providing satellite imagery has improved significantly, with output types ranging from simple traditional photographic images to complex spectral graphs. These developments increase the amount of data that is collected on a daily basis. The data in most cases is so abundant that manual processing is not an option for analysis of all results. As such,

Natarajan Meghanathan et al. (Eds): SIPP, BIGML, DaKM, SOEN, AISC - 2020
pp. 39-58, 2020. © CS & IT-CSCP 2020 DOI: 10.5121/csit.2020.100204

there has been a prominent trend in techniques used to automate this process and host the processing in massive online cloud servers.

These processes include data mining (DM) and machine learning (ML) which will be discussed in this individual report. ML has been emphasized in this study as most methods currently being used in earth observation fall under the heading of 'machine learning'. The types techniques that will be discussed include clustering, regression, neural networks, and convolutional neural networks (CNNs).

This paper will show how some of these techniques are currently being used in the field of earth observation. Some of the challenges of the tools and environments that are currently used will also be discussed. As a practical exploration of these techniques using historical earth observation data, Google Earth Engine (GEE) has been chosen to process and run our scripts on publicly available Landsat RS data catalogues. GEE currently is able to display 40 years of historical satellite imagery and has a built in JavaScript application programming interface (API) that makes geospatial analysis across petabytes of data possible in the Google Cloud [1]. Using this RS data, it is possible to use various DM and ML techniques to classify and discover historical algal blooms in the Baltic Sea surrounding the Swedish island of Gotland.

The organisation of this paper is as follows: Section 2 will discuss data mining patterns and techniques used in EO. Section 3 will look at some of the challenges that are specific to analysing data for EO including discussing training data. In section 4, a 'real-world' example is introduced by looking at historical data in the Baltic Sea and using GEE to identify algal blooms over a period of ten years. Discussions and recommendations are found in section 5 followed by conclusion of the study in section 6.

2. Literature Review of Data Mining Patterns and Techniques in Earth Observation

There are a number of methods that can be used in creating data mining patterns specific to Earth Observation. Amongst these are clustering, regression, neural networks and CNNs. These will be briefly summarized in the following section.

2.1. Clustering

Clustering, according to Berkhin [2], is "a division of data into groups of similar objects." The objects within the cluster groups are similar to each other and not similar to objects from other groups. There are a number of clustering algorithms, amongst these are hierarchical, partitioning and grid-based methods, constraint-based clustering, scalable clustering, and high dimensional data algorithms. Figure 1 below, is an example of clustering.

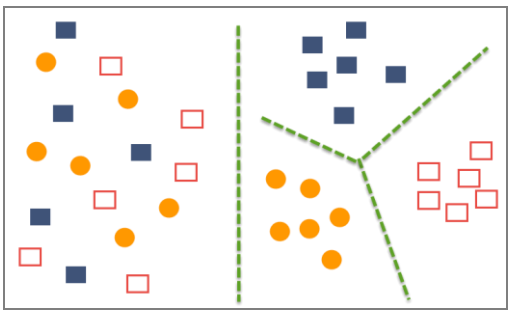


Figure 1. An example of clustering

2.2. Regression

In data mining, as defined by Oracle [3], "Regression is a data mining function that predicts a number." In a regression model, the data set that is used to predict the outcome, has values that are known. The attributes in the data set are called predictors and the outcome is a value which is known as the target. One example of this can be housing cost estimation. The value of the house is the target and the predictors could be attributes as number of rooms, age, location, previous sale costs, etc [3].

Regression can be linear or non-linear. A linear regression is based on the ability to approximate the relationship between the target and the predictors with a straight line. In non-linear regression a relationship is unable to be approximated by a straight line, so a more complex equation has to be defined. In figure 2, a graph with a single predictor is shown for linear regression. The y axis is the target and x is the predictor. The error, also known as the residual, is a measure of the difference between the predicted and the expected value [3].

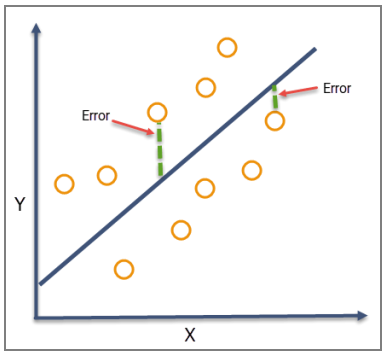


Figure 2. An example of linear regression

2.3. Neural Networks

Neural networks were originally designed to be an analogue of the computation of biological neurons. It has been described by Han, Kamber, and Pei [4], as, "…a set of connected input/output units in which each connection has a weight associated with it." As the network learns, it adjusts the weights of inputs and adjusts accordingly in order to predict classes. This generally involves a long training time and there are criticisms that it is very hard to interpret where the weights come from and what the hidden units are in the network. Neural networks can be described as a 'black box' in that inputs go in and an output is given, but it is unknown what exactly happens inside the box to get to the output. The advantage of neural networks is that they are very tolerant of data that is 'noisy'. They are also very good for classifying patterns that have not been trained and where there is little known about the attributes and classes and the relationships between them [4].

Below in figure 3 can be seen a basic node of a neuron for a neural network. The inputs are given and weights are attached to each. The inputs are then summed and then go through an activation function to determine if the signal should progress further. If the neuron is activated, it is able to contribute to the overall outcome. The outcome can be the function of adding the object to class or participating in a classification.

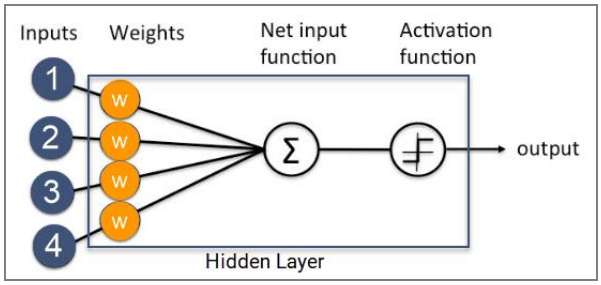


Figure 3: Diagram of a neural network node [5]

Multiple nodes together make a layer and the layer subsequently contributes to the next layer. In this model, there are at least three layers, an input layer, a hidden layer, and an output layer [6]. Neural networks can be classified into three categories, a supervised neural network, an unsupervised neural network and a reinforcement neural network. In a supervised neural network, the network relies on training data. In an unsupervised neural network, there is no training data provided, the network tries to creates the correlations on its own and uses these to classify new data. In a reinforcement neural network, the network learns by means of penalties and rewards for right and wrong decisions. [7]

A basic three-layer network can be seen on the left side of the diagram in figure 4.

2.4. Convolutional Neural Networks

A subset of neural networks that has gained much attention in visual recognition is the convolutional neural network. The structure of CNNs allows for the learning of abstract feature detectors and allows mapping of these features into representations. These representations enhance performance of future classifiers [8].

CNNs have an architecture with multiple stages that each contain three layers. These layers include a convolutional layer, a pooling layer and an output layer (fully-connected layer). The convolutional layer is where the primary processing is done by having a spatially small filter slide, or convolve, over the full volume. This in turn produces an activation map that is two dimensional. The pooling layers main functions are to reduce the spatial size, number of parameters and control overfitting. The output or fully-connected layer contains the final scores of the classification [6]. CNNs have become specifically useful with RS data for scene understanding, target recognition and pixel classification [8].

Convolutional networks deal with tensors, which are in essence nested arrays. The layers of CNNs are arranged in three dimensions that also include depth. Images are defined as three dimensional objects which include colour encoding. In figure 4, is a CNN (right side) compared to a regular three-layer neural network (left side). The image input in the CNN is shown as the red block with the dimensions of the image being the height and width. The Red-Green-Blue (RGB) channels make up the depth.

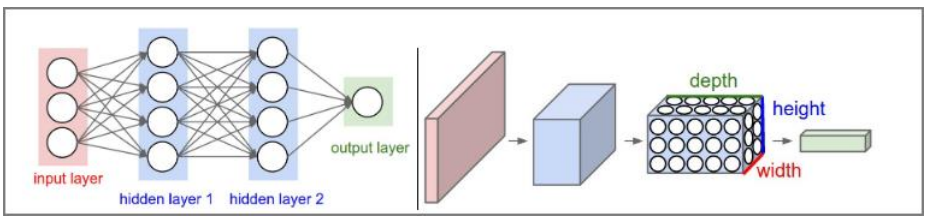


Figure 4: Comparative diagram of a three-layer neural network (left) vs a CNN (right) [6]

3. CHALLENGES INHERENT TO EARTH OBSERVATION AND REMOTE SENSING

According to Kanevski et al. [9], geospatial data in general, has very specific characteristics or 'particularities' that complicate analysis and prevent modelling via traditional geostatistical models. Amongst these are nonlinearity, spatial and temporal non-stationarity, multi-scale variability, presence of noise and extremes/outliers, and a multivariate nature [9].

Additionally, Ball, Anderson, and Chan [10] after reviewing 57 survey papers in DL and RS and 205 RS applications papers compiled a list of nine challenges for DL in RS. The issues they identified are below in table 1.

One of the issues listed below by Ball, Anderson, and Chan [10], is a "high barrier to entry" as a challenge for the RS community. Until a few years ago, most ML tools were made by software developers for software developers. These tools also use a variety of programming languages and proficiency in the language is typically a requirement to use them. It is only in the last few years that there has been a focus on making tools easier to use for non-developers.

The size of the data is also restrictive for most independent or student researchers to process on personal computers as processing generally requires systems with multiple graphic processing units (GPUs).

Table 1. Challenges for deep learning in remote sensing [10]

	Challenge	**Details**
1	Limited data sets/limited training data	In all the papers surveyed, there were five commonly used data sets. They showed that overall accuracies can not necessarily be trusted based on the number of training samples for each paper. They also showed that the data sets are saturated. They recommend that new data sets are required. In RS, there is only a small set of imagery with samples labelled for training.
2	Models for RS applications are often very complicated	RS models can have very intricate relationships and can be inaccurate if the input data does not take this into account. Low temporal resolution can be a challenge as well. The recommendation is to focus on more complex features instead of pixel-level and spatial patterns
3	Big data	Algorithms need to be streamlined and there needs to be better processing power. There is a focus on being able to combine different types of data.
4	Non-traditional data sources	Using sources such as social media photos and videos or tweets with geo-location data for real-time analysis.
5	DL architectures	Complex RS problems may not be solvable with current DL architectures.
6	Transfer learning	Current challenges include, transfer when endmembers are not the same, transfer of low to midlevel features, especially those from different domains, transfers for imagery collected in different atmospheric conditions and times.
7	An improved understanding of DL systems	New DL methods, both practical and theoretical need to be explored to go deep. There should be improved training and generalization capabilities.
8	High barriers to entry	Hardware restrictions and multiple software development requirements can create a steep learning curve for DL. Many RS tasks are not included in standard libraries.
9	Training and optimizing the DL	There are many ways to train a DL system and it can be difficult. DL systems can also have millions of parameters.

Below in table 2, is a summary of some of the RS applications from twenty additional papers that were surveyed by Ball, Anderson, and Chan [11]. This paper's aim was to, "showcase what has been done, what is being done, and what big questions remain and need to be tackled by the community." [11]. They identified CNNs as being one of the primary algorithms used for remote sensing data, often deep neural networks. They also found it was common to use non-remote sensing pre-trained data as well as transfer data to assist in classification.

Table 2. Challenges and contributions in RS applications [11]

RS Application	Challenges	Example Contributions
Synthetic Aperture Radar (SAR) processing	Traditional SAR processing methods use features crafted by hand	• CNNs for feature extraction allow for change detection and classification • Algorithms that require no prior processing or segmentation
Ocean processing	Ships are very small, cloud interference, wave interference	• Deep CNN to extract features and detect ships • Provided bounding boxes for recognition of ocean fronts
Classification and labelling	Large size of imagery, multiple resolutions, image matching	• CNNs for the recognition of dust, smoke, hurricanes, etc. • Deep CNNs for detection buildings from orthoimages
Multimodal (mixed techniques)	Combining multiple technologies	• CNN detector for golf courses, augmented with temporal data • Hyperspectral and visible images combined with CNN for feature extraction later also combined with spectral, statistical and spatial data
Spectral-spatial processing	Anomalies in hyperspectral data	• Stacked denoising autoencoder for hyperspectral anomaly detection • Deep stacked sparse autoencoder for feature learning in hyperspectral images
Object tracking and recognition	Large spatial areas, drifting in long term tracking	• Dual correlative deep networks for aircraft recognition • CNN spatial clustering and chip detection for the identification of surface-to-air missile sites
Architectural studies	Determining if shallow CNNs are sufficient for feature recognition	• For remote scenes, greater CNN depth is essential for identifying features • Over time, CNNs have become deeper

3.1. Training data

The size of the data is also a major factor in being able to train a system effectively. Ball, Anderson, and Chan [10], identified the challenges associated with training for DL systems which deal with vast amounts of RS data.

The training issues they identified include:

- DL systems can have millions of parameters
- RS data may not be labelled
- Hyperspectral data is a very large data cube with many layers, while DL algorithms are typically trained from very small RGB images
- Light detection and light detection and ranging (LiDAR) have insufficient literature as the data is not an image, but a point cloud
- Gridded searches or random methods are required for optimization which can be very time consuming [10]

Helber et al. [12], have also emphasized the importance of having a high-quality dataset for training and classification. They state that one of the challenges to creating these training sets has been access to ground truth datasets that are reliably labelled. In response to this challenge, they created a dataset called EuroSAT from Sentinel-2 images. Their dataset consists of 27,000 images consisting of 10 different classes which can be seen in figure 12. For instance, in the river, and sea and lake classes, they have tried to accommodate the various colours, locations and bodies of water. Each of the 10 classes contain 2000 to 3000 images. Each image measures 64x64 pixels and is at a 10 m per pixel resolution.

Approximately 1.6TB of data in the form of compressed images comes from the Sentinel-2 constellation every day. Each of these 27,000 images had to be manually checked and sorted multiple times in order to get an acceptable accuracy for their training set [12].

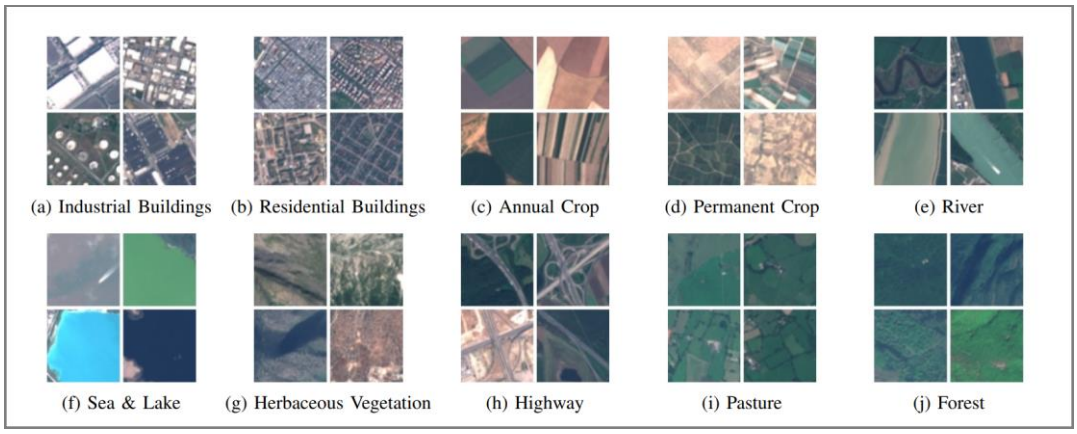


Figure 5: EuroSat training dataset from Sentinel-2 images [12]

4. REAL WORLD EXAMPLE: DISCOVERING ALGAL BLOOMS WITH DATA MINING AND MACHINE LEARNING IN EARTH OBSERVATION

4.1. Phytoplankton, cyanobacterial and algal blooms, eutrophication

The photo below in figure 6, which is of Gotland, a Swedish island in the Baltic Sea, is called 'Van Gogh from Space'. It is a part of the United States Geological Survey (USGS) 'Earth as Art' image gallery [13]. Aside from being a beautiful image, the image also has the significance of showing a 'phytoplankton bloom'. This image was taken on the 13th of July, 2005 from Landsat-7. The phytoplankton biomass is caused by an increase in nutrients, generally associated with rising nitrogen concentrations. These occurrences are often called algal blooms and are also associated with cyanobacteria blooms, which have the potential to be toxic or harmful. They can be natural and seasonal or caused by pollution originating from densely populated areas or industrial runoffs [14]. The red arrows in figure 6 show these algal blooms. The image below is a colour enhanced image and the blooms are seen in bright green. Landsat-7 bands and the appearance of colours in feature properties will be discussed in section 4.2.

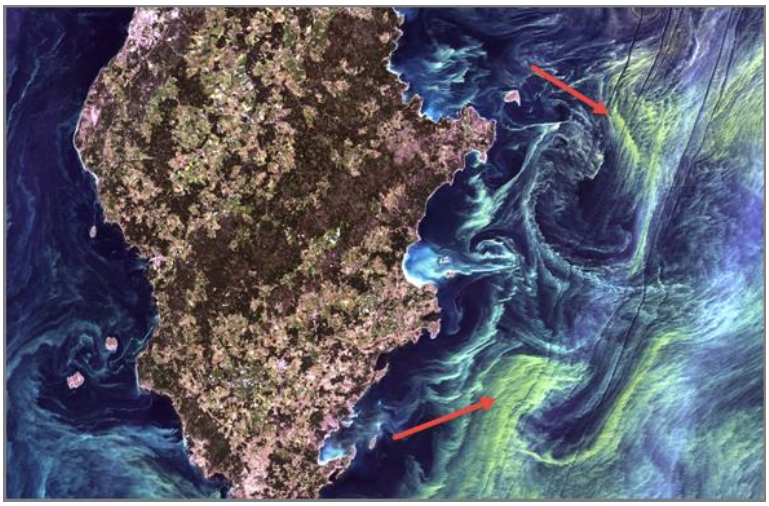


Figure 6: Gotland Island on 13 July, 2015. Red arrows indicate algal blooms [13]

A bloom occurring because of related pollution is known as eutrophication. The blooms are also correlated with an increase in chlorophyll a which contributes to the green color of the blooms [14]. The seasonal variations of these blooms in the Baltic Sea are seen generally during July and August when the water is warmer [15]. In the Gotland Sea, these seasonal blooms have also been known to occur in the autumn months from October through December [14]. Signs of blooms outside the seasonal windows can be possible indications of eutrophication.

Up until the 1960s, most blooms which were the result of eutrophication were recorded only in coastal waters. After the 1960's it became common to see these blooms occurring in the open areas of the Baltic Sea [16].

4.2. Tools and methodology: Google Earth Engine

The Google Earth Engine has been chosen as the tool for this study. GEE currently is able to display 40 years of historical satellite imagery, including publicly available Landsat-7 RS data catalogues, and has a built in JavaScript API that makes geospatial analysis across petabytes of data possible in the Google Cloud Platform [17]. This means that a local installation is not necessary and all computations can be executed in the cloud. As these are extremely large datasets, this is one of the major advantages and reasons for choosing GEE. GEE also has a Python API that is equipped with Cloud Datalab which allows it to be run with Jupyter notebooks. The JavaScript API is available in the GEE coding editor and is an online Integrated Development Environment (IDE) which allows for immediate visualization and rapid prototyping [17]. The IDE can be seen in figure 7.

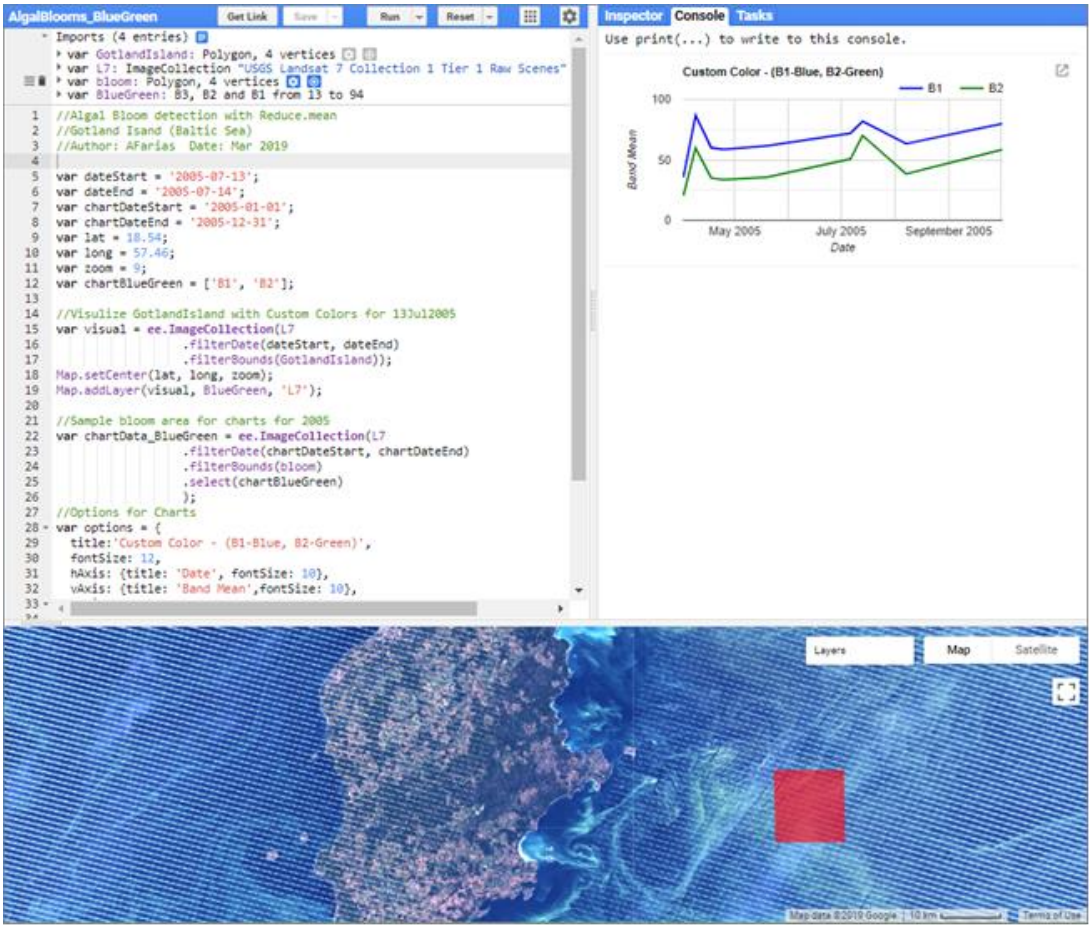


Figure 7: Earth Engine IDE. Shown is the JavaScript editor (top left), console with a chart in the console output (top right), and map visualization layer (bottom)

Earth Engine, although relatively new, has already seen hundreds of scientific papers published making use of the tool for a variety of applications such as medical studies, vegetation and forestry, wetlands and hydrology, agriculture, urban studies and disaster management [18].

4.3. Landsat-7 data catalogue in Earth Engine

Landsat-7 was launched on the 15th of April, 1999 and is currently still operating today. At the time of submission of this report, Landsat-7 will be celebrating its 20th year anniversary in orbit, but is expected to be replaced by Landsat-9 late 2020 [19]. Landsat-7's system capabilities include high volume, high resolution and multispectral resolution while averaging 250 scenes per day. It was designed for a 705 km, sun synchronous orbit with a 16-day mapping cycle. It also has an internal cloud cover prediction mechanism and only captures sunlit areas which prevent it from collecting data that is unusable [20].

ETM+ has a swath of 185 km, with six spectral bands (1-5, 7), a panchromatic band (8) and a thermal band (6). It has a spatial resolution of 30 m for the spectral bands, the panchromatic band has a resolution of 15 m and the thermal band has a 60 m resolution. All bands have two gain settings of high or low [19],[20].

4.3.1. Earth Engine's implementation of Landsat-7 data

In |GEE, the Landsat-7 data catalogue, which has been acquired from USGS, has designated 10 bands for use in analysis. This can be seen below in table 3. The bands are selectable in layers of up to three bands to create a composite image in GEE. In GEE, the gain settings are only separated in thermal band 6 as B6_VCID_1 and B6_VCID_2. Band 6 has also been resampled from the original 60 m resolution to 30 m.

The dataset used for this study is the USGS Landsat 7 Collection 1 Tier 1 Raw Scenes collection (LANDSAT/LE07/C01/T1). Tier 1 describes scenes with the highest available data quality. These scenes are appropriate for time-series analysis as they have been calibrated across the various Landsat sensors and they have Level-1 Precision Terrain (L1TP) processed data [17].

Table 3. Landsat-7 Band Details for GEE [19],[20], [21]

Band Name	Resolution	Wavelength	Description
B1	30 m	0.45 - 0.52 µm	Blue
B2	30 m	0.52 - 0.60 µm	Green
B3	30 m	0.63 - 0.69 µm	Red
B4	30 m	0.77 - 0.90 µm	Near infrared
B5	30 m	1.55 - 1.75 µm	Shortwave infrared 1
B6_VCID_1	Resampled from 60 m to 30 m	10.40 - 12.50 µm	Low-gain Thermal Infrared 1
B6_VCID_2	Resampled from 60 m to 30 m	10.40 - 12.50 µm	High-gain Thermal Infrared 2
B7	30 m	2.08 - 2.35 µm	Shortwave infrared 2
B8	15 m	0.52 - 0.90 µm	Panchromatic
BQA			Landsat Collection 1 QA Bitmask

4.4. Recognizing algal blooms with remote sensing data from Landsat-7

The measures for computing a body of water as eutrophic include secci-disk transparency (SDT), total phosphorus (TP) and chlorophyll-a (Chl-a). Chl-a measurements are not influenced by sediment or acids and correlate with the volume of phytoplankton concentration in a body of water. The increase in Chl-a is a good indicator for detecting blooms specifically with RS data [22].

Fuller, Aichele, and Minnerick [22], determined in Landsat 7 images, that to detect Chl-a, "…the combination of band 2 (Green), band 3 (Red), and band 7 (short wave infrared) produced the highest R2 values." R2 is the coefficient of determination and gives a statement about the error or the residual, as shown in chapter 2.2.3.6, between the predicted and the expected value. The higher the value is the better the prediction is.

Therefore, the recognizing of chlorophyll-a (variable Chl-a) from Landsat-7, for Fuller, Aichele, and Minnerick [22], can be seen in equation 1, where the variables a, b, c and d are the derived coefficients from the regression equation. For the purposes of identifying this configuration of bands, it has been labelled it as 'FAMChl'.

$$\ln(Chl - a) = a(band2) + b(band3) + c(band7) + d \quad (1)$$

Weber [23], has specified that in order to identify cyanobacteria specifically, and not green algae, a 620 µm band is necessary. This falls in between band 2 and band 3 and is not covered by the instrumentation on Landsat-7. Therefore, any detections with Landsat-7 are assumed to only be algal blooms and make no assumptions about corresponding cyanobacteria levels.

4.5. Visualization of band configurations

A number of other configurations were looked at for band optimization. Images from the visualization of the following band composites can be seen in figure 8. These include the following:

- RGB 'True Color' – (B3,B2,B1)
- False Color – (B4,B3,B2)
- Short-Wave Infrared (SWIR) – (B7,B4,B2)
- FAMChl – (B3,B2,B7)
- Custom Color – (B3, B4, B7)

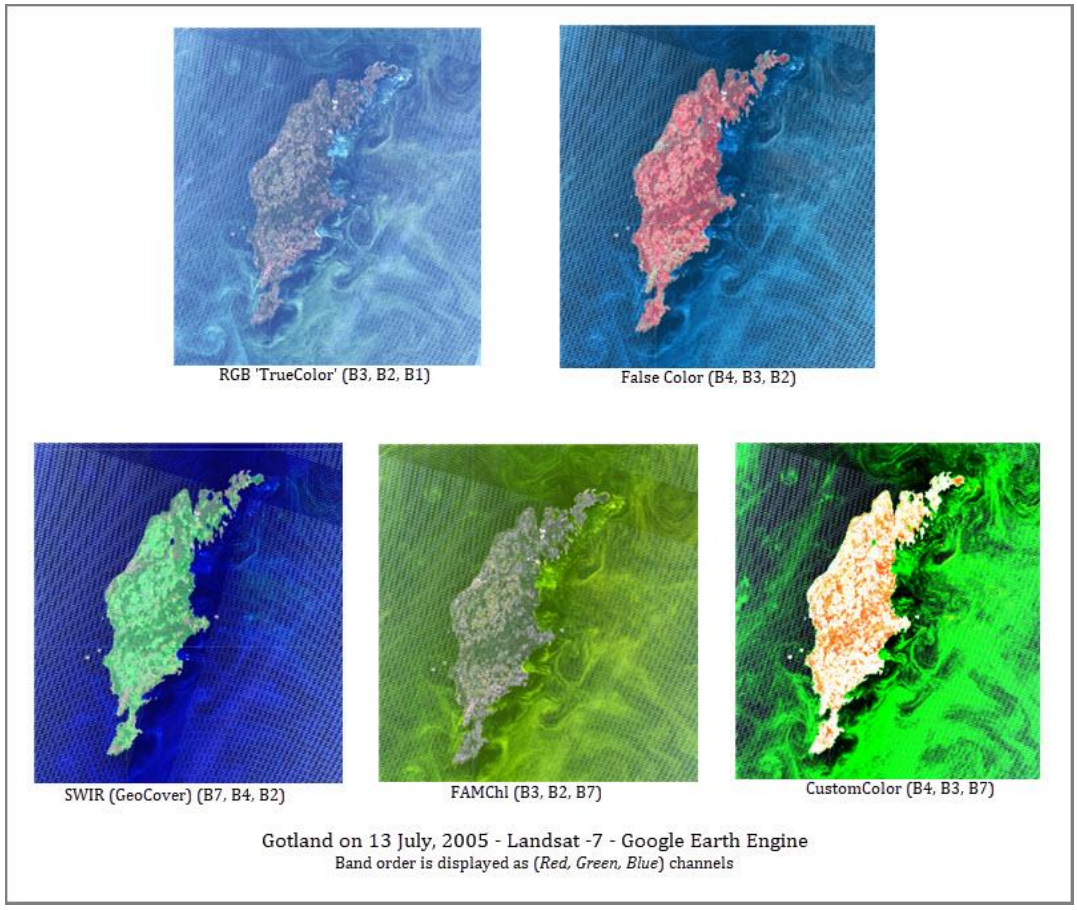


Figure 8: Visualizations of band combinations tested for detecting algal blooms

The artefact lines that can be seen in the images figure 9 are part of a Scan Line Corrector (SLC) failure that happened on Landsat-7 on May 31st, 2003. Despite the SLC fault, a USGS report found that the data was still excellent quality for at least 86% of the pixels when augmented with interpolation [24]. All Landsat-7 images have been processed with the SLC in 'off' mode since the fault.

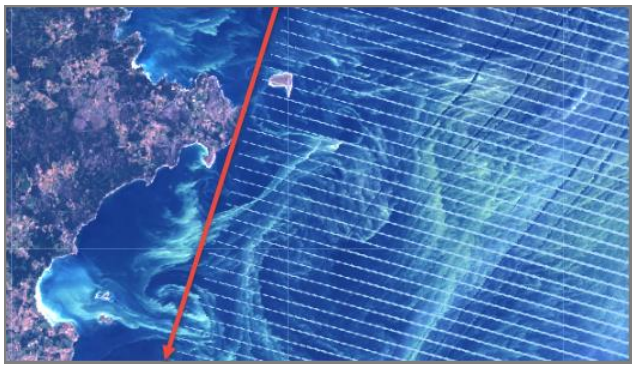


Figure 9: Visualizations of band combinations tested for detecting algal blooms

The colour of primary features such as vegetation or water changes significantly depending on the combination of bands chosen. This is visually significant, but also significant when sampling data at the pixel level for scene analysis. Feature colours can be seen in the table in figure 10 for true colour, false colour and SWIR.

	True Color **Red: Band 3** **Green: Band 2** **Blue: Band 1**	**False Color** **Red: Band 4** **Green: Band 3** **Blue: Band 2**	**SWIR (GeoCover)** **Red: Band 7** **Green: Band 4** **Blue: Band 2**
Trees and bushes	Olive Green	Red	Shades of green
Crops	Medium to light green	Pink to red	Shades of green
Wetland Vegetation	Dark green to black	Dark red	Shades of green
Water	Shades of blue and green	Shades of blue	Black to dark blue
Urban areas	White to light blue	Blue to gray	Lavender
Bare soil	White to light gray	Blue to gray	Magenta, Lavender, or pale pink

Figure 10: The colour of feature display in composite images [25]

4.6. Historical remote sensing and ground truth comparisons

Since 2002, the Swedish Meteorological and Hydrological Institute (SMHI) has been monitoring algae levels in the Baltic Sea. They are now, since 2009, supplementing traditional water sampling methods with satellite data from ENVIronment SATellite (ENVISAT) and Earth Observing System Aqua (EOS-AQUA) using the MEdium Resolution Imaging Spectrometer (MERIS) and Moderate Resolution Imaging Spectroradiometer (MODIS) sensors respectively [26]. The sensors are only able to detect surface level algae which have a high reflectance. They are not able to see algal blooms currently through clouds or at night [26].
Hansson and Hakansson [27] described the 'Baltic Algae Watch System' which monitored cyanobacterial blooms from 1997-2006. They processed images from the National Oceanic and Atmospheric Administration (NOAA) - Advanced Very High-Resolution Radiometer (AVHRR) (NOAA-AVHHR) based on a supervised classification algorithm in near infrared and thermal channels. The NOAA-AVHHR data, however, has a poor resolution at ~1 km^2 which made costal detection difficult [27].

In 2005, there were 13 Cholorphyll-a measurements taken at the 'BY15 GOTLANDSDJ' (Gotland Station). The station is marked by the red x in figure 11. This number of samples varies from year to year. For instance, there were 11 samples taken in 2016 [28].

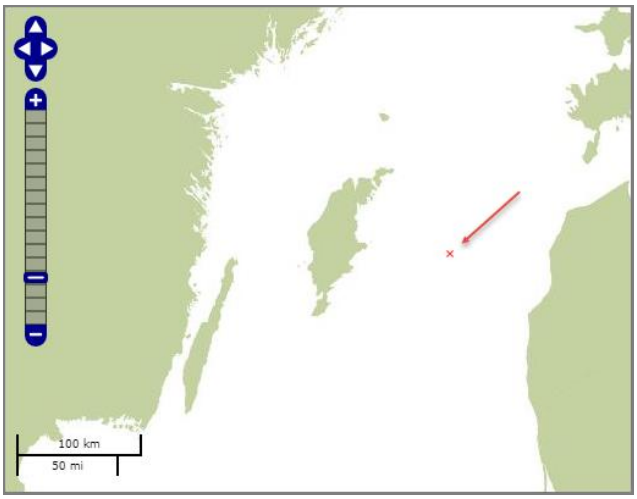


Figure 11: Location of SMHI Chlorophyll-a sampling (SMHI, 2019b)

The data for these measurements is publicly available on the SMHI Swedish Sea Archives/Svenskt HavsARKiv (SHARK). The SHARKweb has data collected by SMHI that goes back to 1893 for numerous marine biological, chemical, and physical parameters [29].
The 13 samples for Chl-a that were taken in 2005 can be seen in figure 12. It should be noted that the dates of sampling do not necessarily correspond with the dates of blooms or the dates that Landsat-7 was acquiring data.

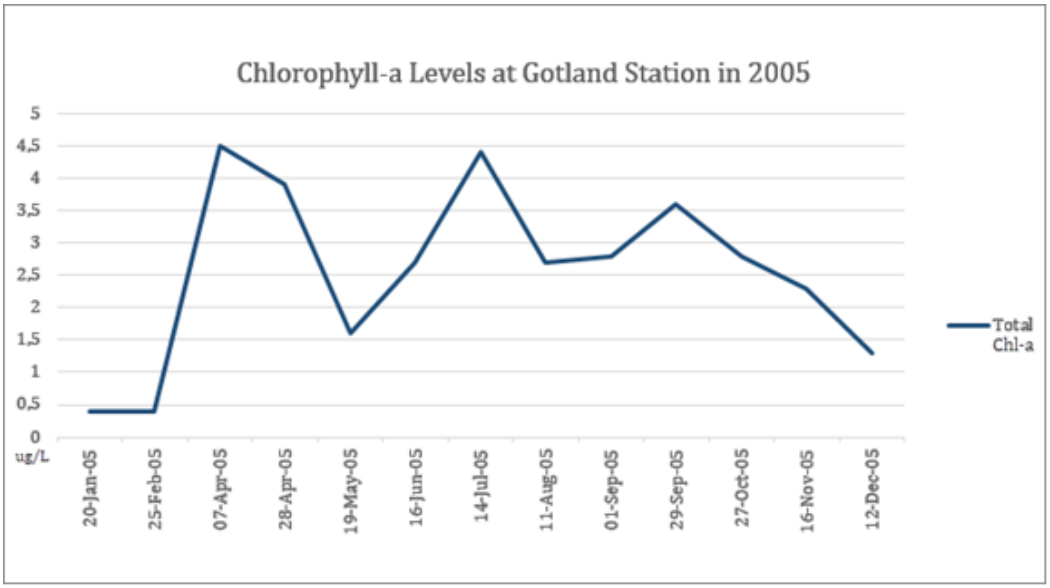


Figure 12: Chlorophyll-a levels sampled from Gotland station in 2005, from the SMHI database for use in the ground-truth comparison

4.7. Mean reduction by geometry region in Earth Engine

GEE has the ability to define specific geometry objects to areas on the map that are to be used for analysis. For the purposes of this study, a rectangle around Gotland Island was defined. This can be seen in light blue in figure 20. The darker blue around it is the result of returned Landsat-7 data for that day. Multiple swaths can be in the image depending on the angle that the image was acquired. In figure 20, there is only one swath of 185 km in width. The area of the Gotland Island geometry is ~32,000 km, the area of the bloom geometry is ~100 km.
The sample size of ocean was approximately 10km^2 as seen in red in figure 13. A test was also done by increasing the number of sample blocks to three, however the accuracy decreased significantly. All further tests were done using one sample block.

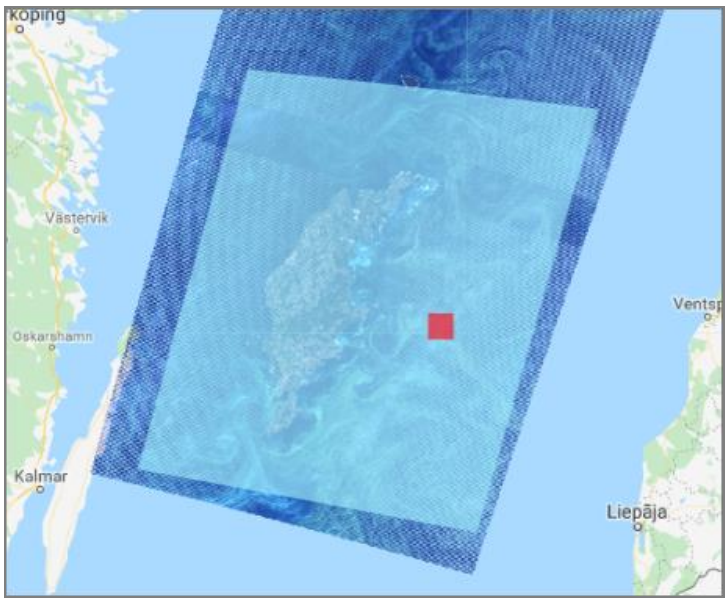


Figure 13: Geometry objects used for analysis in GEE. (Light blue) Gotland Island (Red) bloom sampling location

The ee.Reducer is a way to get pixel statistics of a geometry or region. The reducer will take an area in the image and compute a value for each of the bands. If it is a true colour image with RGB bands, it will return three numbers, one for each band. In each of the band combinations above, there are three bands, so what has been done is to take the mean of the pixels and between the bands. All bands must be specified to be sampled at the same resolution, in this case, the sampling was done at 30 m as the bands all have this spatial resolution. This is especially useful when comparing spectral values over a time series. It is worth noting that the maximum pixels that the reducer can compute is 10 million, so large areas, such as the whole island cannot be computed. This will work with smaller areas such as the bloom area. There is also a function to only use the maximum number of pixels. The result from this will be a random selection of pixels up to 10 million. A diagram of how ee.Reducer is used with the workflow can be seen in figure 14.

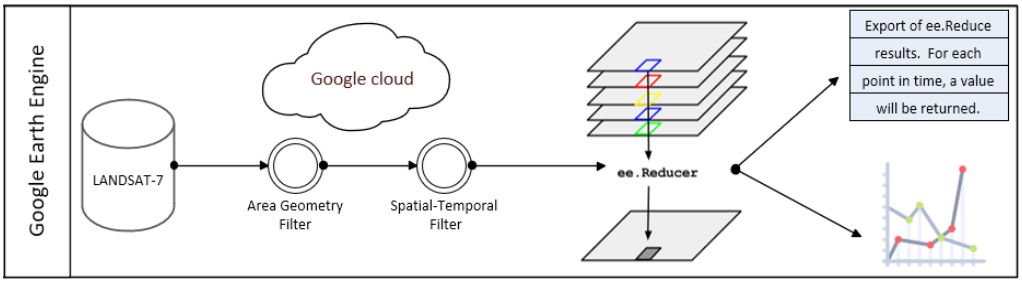


Figure 14: Workflow of ee.Reducer being applied to Landsat-7 with an area geometry filter and a spatial-temporal filter. [29]

The band mean spectral values for CustomColor, TrueColor, FalseColor, FAMChl and SWIR were calculated by using ee.Reducer.mean. The values for Chl-a levels as well as the band mean spectral value were plotted seen in figure 15. The results of this exercise also showed that the sampling dates for Chl-a and the collection date for the Landsat-7 images were non concurrent. There were 13 sample dates for Chl-a and 9 dates for Landsat-7 data. Of these, only one date perfectly matched. In the end, only 19 sample dates were used as there was no Landsat-7 data for Jan, Nov or Dec. There are many reasons for why this data was missing. Landsat-7 is meant to have a 16-day return cycle, but it is also meant to discard any scenes with cloud cover or any night time scenes. It is also possible for scenes to be discarded where there are system faults or calibration errors.

In the case of algal blooms in the Baltic Sea, they are often seasonal and can last an extended period of time, so an assumption was made that the last sampling or collection value would remain until the next one was received. This could lead to an uncertainty for a detailed analysis of time periods with an offset to a certain data point. For instance, it would be incorrect if a Chl-a sample was taken when there were no blooms and a bloom only appeared a few days after when the satellite passed the region. This method did, however, allow for a rough trend to be seen when plotted and a preferred band was chosen as a result. A more accurate regression model/curve would be possible with more row data and more data points.

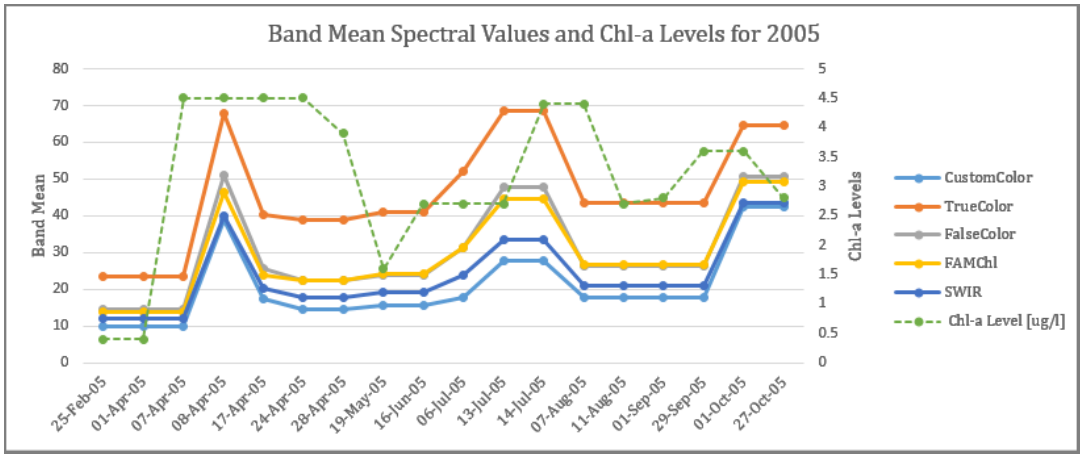


Figure 15: The lack of concurrent sampling points from SMHI and Landsat-7 image data means that data over a larger period is needed

As a backup to the method above, all bands were also plotted individually. By doing this, it was shown that bands 1 and 2 were preferred over band 3. As such, for all future analysis only band 1 (blue) and band 2 (green) were used for analysis. This can be seen in figure 16.

According to Pitarch et al., [31] there are two types of algorithms presently being used to measure Chl-a via remote sensing. This includes an empirical method using a blue/green reflectance ratio, as we have identified above. The other method is a semi-analytical method, where the water type is more complex in the possible colorations found. Case I waters are generally found in open waters where as Case II waters can be anywhere the water is discoloured. They have considered the Baltic Sea to be a "challenging test bed for remote sensing" with a high concentration of coloured dissolved organic matter (CDOM). The standard algorithm that is provided for Chl-a detection, generally by the space agencies, is one that is specific to global applications and does not take into account these discolorations specific to certain regions [31].

4.8. Results: Analysis with blue and green bands

Since Landsat-7 launched in 1999, it was decided to do a ten-year initial analysis through to 2009. After 2009, the SMHI changed the way they observed Chl-a via remote sensing, so a comparative analysis by year is challenging beyond this period. For a more recent analysis, the years of 2010-2019 can be also studied together, however, that is out of scope for this paper. Nevertheless, the results in this paper can be used for further analysis and studies of the earlier period.

The numbers for band 1 and 2 returned from the initial test are shown in figure 16. The chart is a sample of the charts included in the GEE IDE. As seen in this plot, there are a number of spikes above 100. These were cross checked to bloom dates with SMHI and did not correspond. Also, in all the tests with the 2005 data, there were never any points above 90.

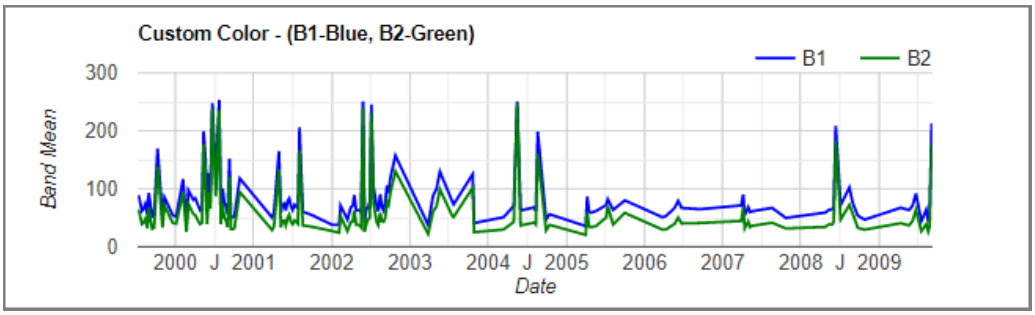


Figure 16: Landsat-7 band 1 and 2 spectral values for the bloom geometry with abnormal spikes above 100 (1999-2009)

The individual spectral values were exported and evaluated against visual representations of the plotted days in question. Anything over 100 was considered an anomaly and in most cases, when cross checked visually, was due to cloud cover or what appeared to be instrument errors. There was also a very basic preliminary comparison of the number of days of known cyanobacterial blooms for the period of 1999-2009. For this, the spectral values have been compared with heat maps from SMHI with the number of days with cyanobacterial blooms [26].

The next evaluation that was done was to attempt to match days with data points for both Chl-a sampling and satellite imagery. For the period of 1999-2009, there were 137 days with Chl-a data samples. For Landsat-7, there were 145 days. These were matched with an interval of +/-7 days. This left 96 days over the ten-year period. Because there were some unexplained discrepancies in data around 2006, possibly due to recalibration, it was decided to focus on the years of 2001-2005 (minus SLC fault data). The final analysis contained 45 days and can be seen in figure 27.

The SCL fault occurred on the 31st of May, 2003 and data from the satellite became temporarily unavailable. The data became available again in July, but a second product with the gaps filled in via interpolation was not available until May 10, 2004 [20]. All data points from this time were also removed as seen on the x-axis in figure 17.

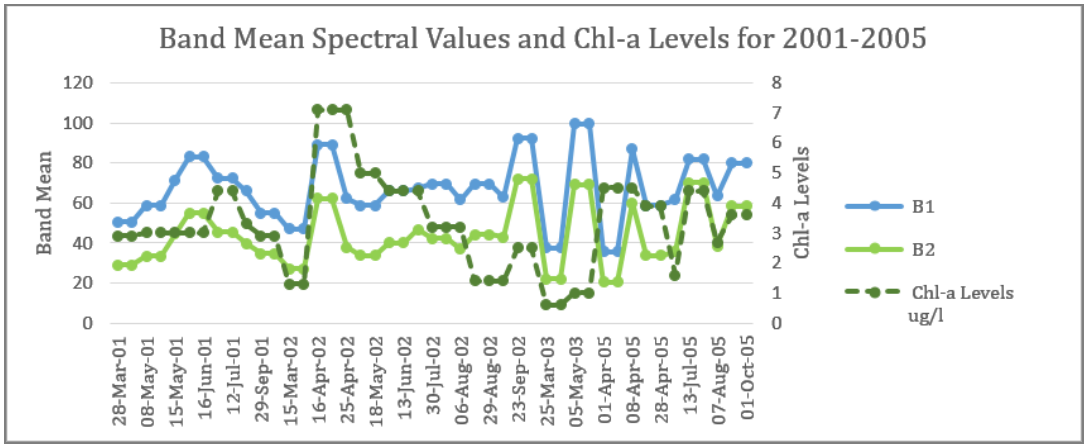


Figure 17: Bands 1 and 2 plotted as spectral values by date shown with Chl-a sampled levels

While there are a number of correlations between the collected samples and spectral values in the blue and green spectrum, more detailed analysis of individual points is necessary. While we have seen that cloud cover should be filtered from the collection, certain outliers were still found, and it is possible that a partially clouded scene can be skewing results. There are a number of cloud filtering algorithms that can be explored in future studies.

5. Discussion and Recommendations for Future Study

The study of detecting algal blooms via earth observation was chosen because it has been identified as having seasonal events that are regularly monitored via satellite. As an added benefit, it is also monitored via ground truth methods such as water sampling. This made it possible to test various detection techniques and cross check them with historical results. Gotland Island was chosen as there was a very large bloom that extended over a longer period in 2005. The algal blooms in this area also have a very high reflectance which can be picked up with remote sensing.

There were a number of discoveries in the analysis of the earth observation data. One of the most important ones is that ground sample dates do not necessarily correspond with Landsat-7 pass over dates. There is also a chance that the data collected is not useable, such as with heavy cloud cover. Because of this, it is recommended that a future study be done with a satellite or a constellation of satellites that have a more frequent revisit time. Also, in order to study cyanobacterial blooms as well as algal blooms, it is recommended to use a 620 µm band which Landsat-7 does not have [22].

The area that was sampled for this study was ~10 km^2, which is a very large area. There was also only one sample location as the initial tests with multiple locations showed a decrease in accuracy. This should be further studied with various sample areas and sample methods. Also, other sample methods might be necessary for different bodies of water with other algal types. The tested areas should also include coastal areas which can have vastly different colorations depending on runoffs and surrounding terrain.
This exploration was done specifically using the result of single spectral value which was calculated for a sample location. There are also other random sampling methods that could be

used and tested for better results. The algorithms that have been previously packaged in ocean observation software are fairly complex and have the need for customization to specific areas [31]. One future potential that has arrived from this study could be building a training set of historical algal blooms from the data points identified above.

The GEE JavaScript API was excellent for fast prototyping and testing ideas, however, for more powerful algorithms, extended libraries should be considered for in depth studies. The Python API is a good candidate for this, and Google is also working on a number of solutions for unsupervised learning based on the Weka platform [31].

It is worth mentioning the computer processing time involved in preforming the analysis over multiple years. In previously used Geographic Information Systems (GIS), such as ArcGIS, this type of analysis would have taken hours or even days [23]. Because it is hosted in the Google cloud, the processing for ten years of data took a matter of seconds. This is the biggest take away from this exploratory study with Google Earth Engine. The ability to rapidly prototype and visualise results across vast datasets, within seconds, has the potential to dramatically change earth observation data mining techniques.

6. CONCLUSION

The areas being studied, data mining and machine learning, are wide-ranging fields of study. In EO, there is an increasing focus on machine learning to handle the massive amounts of data that are being collected on a daily basis by higher resolution instruments. There are many challenges in EO, including the size of data, its variable and complex nature, a high barrier to entry, and the datasets used for training the data. However, as the field grows these challenges are being addressed.

Software companies are attempting to combat the barrier to entry by providing easy to use IDEs which enable fast prototyping and almost instant visualizations. A new training data set which is focused on improving training for deep learning systems, EuroSAT, has been implemented
In this paper as a real-world example, the island of Gotland in the Baltic sea was studied by using GEE. Ten years of Landsat-7 data, along with a dataset from SMHI, for ground truth, was analysed to discover historical algal blooms in the Baltic Sea.

One of the biggest take-aways from this study is the speed of processing in GEE, which is hosted in the Google cloud. Where previously, this type of analysis would have taken hours or days to process, it is now available in a few seconds. While GEE is still very new, there have already been hundreds of scientific papers published using it.

Although primarily an exploratory study, this paper has shown the increasing potential for new tools and techniques to enhance the analysis of earth observation data for scientific research.

REFERENCES

[1] N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore, “Google Earth Engine: Planetary-scale geospatial analysis for everyone,” Remote Sens. Environ., 2017.
[2] P. Berkhin, Survey of Clustering Data Mining Techniques. 2002.
[3] Oracle, “Oracle® Data Mining Concepts 11g Release 1 (11.1),” 2008.
[4] J. Han, M. Kamber, and J. Pei, “Data Mining. Concepts and Techniques, 3rd Edition (The Morgan Kaufmann Series in Data Management Systems),” 2011.
[5] Skymind Inc., “A Beginner’s Guide to Neural Networks and Deep Learning | Skymind,” 2019. [Online]. Available: https://skymind.ai/wiki/neural-network. [Accessed: 29-Mar-2019].

[6] A. Karpathy, "Convolutional Neural Networks for Visual Recognition," 2018. [Online]. Available: https://cs231n.github.io/. [Accessed: 29-Mar-2019].
[7] J. Alzubi, A. Nayyar, and A. Kumar, "Machine Learning from Theory to Algorithms: An Overview," in Journal of Physics: Conference Series, 2018, vol. 1142, no. 1.
[8] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," IEEE Geosci. Remote Sens. Mag., vol. 4, no. 2, pp. 22–40, 2016.
[9] M. Kanevski, A. Pozdnukhov, V. Timonin, and A. Pozdnoukhov, "Machine Learning Algorithms for GeoSpatial Data. Applications and Software Tools," 2008, vol. 12.
[10] J. E. Ball, D. T. Anderson, and C. S. Chan, "Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community," J. Appl. Remote Sens., vol. 11, no. 04, p. 1, Sep. 2017.
[11] J. E. Ball, D. T. Anderson, and C. S. Chan, "Special Section Guest Editorial: Feature and Deep Learning in Remote Sensing Applications," J. Appl. Remote Sens., vol. 11, no. 04, p. 1, Jan. 2018.
[12] P. Helber, B. Bischke, A. Dengel, and D. Borth, "Introducing Eurosat: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification," in IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, 2018.
[13] USGS, 2018. Earth Resources Observation and Science (EROS) Center. [Online] Available: https://eros.usgs.gov/image-gallery/earth-art-3/van-gogh-space [Accessed 22 Nov. 2018].
[14] N. Wasmund and S. Uhlig, "Phytoplankton trends in the Baltic Sea," ICES J. Mar. Sci., vol. 60, no. 2, pp. 177–186, Apr. 2003.
[15] S. Anttila et al., "A novel earth observation based ecological indicator for cyanobacterial blooms," Int. J. Appl. Earth Obs. Geoinf., vol. 64, pp. 145–155, Feb. 2018.
[16] T. Finni, K. Kononen, R. Olsonen, and K. Wallström, "The History of Cyanobacterial Blooms in the Baltic Sea," AMBIO A J. Hum. Environ., vol. 30, no. 4, pp. 172–178, Aug. 2001.
[17] N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore, "Google Earth Engine: Planetary-scale geospatial analysis for everyone," Remote Sens. Environ., 2017.
[18] L. Kumar and O. Mutanga, "Google Earth Engine Applications Since Inception: Usage, Trends, and Potential," Remote Sens., vol. 10, no. 10, p. 1509, Sep. 2018.
[19] U.S. Department of the Interior | U.S. Geological Survey, "What are the band designations for the Landsat satellites?," USGS, 2014. [Online]. Available: https://www.usgs.gov/faqs/what-are-band-designations-landsat-satellites-0?qt-news_science_products=7#qt-news_science_products. [Accessed: 20-Nov-2018].
[20] NASA, "Landsat 7 Science Data Users Handbook," 2011.
[21] Google Developers, "Landsat Collections in Earth Engine | Earth Engine Data Catalog | Google Developers," 2019. [Online]. Available: https://developers.google.com/earth-engine/datasets/catalog/landsat/. [Accessed: 01-Mar-2019].
[22] L. M. Fuller, S. S. Aichele, and R. J. Minnerick, "Predicting Water Quality by Relating Secchi-Disk Transparency and Chlorophyll a Measurements to Satellite Imagery for Michigan Inland Lakes, August 2002. Scientific Investigations Report 2004-5086," USGS. 2007.
[23] S. J. Weber, "Utilizing Geospatial Cloud Computing and Data Analytics for Cyanobacteria Harmful Algal Bloom Risk Mapping in Georgia Piedmont Waterbodies," UGA. 2017.
[24] S. Andrefouet et al., "Preliminary Assessment of the Value of Landsat-7 ETM+ Data Following Scan Line Corrector Malfunction," US Geol. Surv. EROS Data Cent. Sioux Falls, SD, USA, 2003.
[25] El Centro Informático Científico de Andalucía (CICA), "An Introductory Landsat Tutorial," 2019. [Online]. Available: https://huespedes.cica.es/geo/agr/Landsat Tutorial-V1.html. [Accessed: 01-Mar-2019].
[26] SMHI, "Monitoring algae from satellite | SMHI," 2010. [Online]. Available: https://www.smhi.se/en/theme/monitoring-algae-from-satellite-1.11923. [Accessed: 01-Apr-2019].
[27] M. Hansson and B. Hakansson, "The Baltic Algae Watch System - a remote sensing application for monitoring cyanobacterial blooms in the Baltic Sea," J. Appl. Remote Sens., vol. 1, no. 1, p. 011507, Dec. 2007.
[28] L. Wesslander, K; Viktorsson, "Summary of the Swedish National Marine Monitoring 2016-Hydrography, nutrients and phytoplankton," Rep. Oceanogr., vol. 60, p. 92, 2017.
[29] SMHI, "Marine Environment Data | SMHI," 2019. [Online]. Available: http://www.smhi.se/klimatdata/oceanografi/havsmiljodata. [Accessed: 25-Mar-2019].
[30] Google Developers, "Statistics of an Image Region | Google Earth Engine API | Google Developers," 2019. [Online]. Available: https://developers.google.com/earth-engine/reducers_reduce_region. [Accessed: 03-Apr-2019].

[31] J. Pitarch, G. Volpe, S. Colella, H. Krasemann, and R. Santoleri, “Remote sensing of chlorophyll in the Baltic Sea at basin scale from 1997 to 2012 using merged multi-sensor data,” Ocean Sci, vol. 12, pp. 379–389, 2016.
[32] Google Developers, “Unsupervised Classification (clustering) | Google Earth Engine API | Google Developers,” 2019. [Online]. Available: https://developers.google.com/earth-engine/clustering. [Accessed: 01-Apr-2019].

AUTHORS

Alexandria Dominque Farias is currently an IT Application Specialist at OHB InfoSys in Bremen, Germany. She is originally from El Paso, Texas, but spent 12 years living in Cape Town, South Africa. She holds an MSc in Space Studies from the International Space University in Strasbourg, France and Masters in Information Technology from the University of Cape Town. She did her undergraduate work at the University of New Mexico and Flinders University of South Australia. She was an IT Officer for Crystal Cruises and held various software development roles at Allan Gray Pty in Cape Town. She started her career at InfoGenesis in Santa Barbara, California where she worked as a Systems Engineer.

Mr. Sun is a professor of Space System Engineering at the International Space University (ISU) located at Strasbourg, France.

Mr. Sun held several senior executive positions both in China and Europe. He started his career as a system engineer of launch vehicle design and followed as project manager of international satellite launching service in China Academy of Launch Vehicle Technology (CALT). He was a founding member of China Manned Space Agency (CMSA) in 1993 and worked as General Designer Assistant for China Manned Space Program (CMSP) for 8 years, he was in charge of launcher system specification definition, system coordination and interface control among spaceship/launcher/spaceport, and conceptual study for RDV as well. After the maiden flight of Shenzhou spaceship, he moved to Munich and worked as Managing Director of EurasSpace GmbH to promote the Sino-European cooperation in space for 8 years. He founded CASC (China Aerospace Science and Technology Corporation) European Office based in Paris in 2010 and served as premier Chief Presentative in the office for 7 years. He was deeply involved in most of the joint space programs between China and Europe in this period.

TWO APPROACHES TOWARD GRAPHICAL DEFINITIONS OF KNOWLEDGE AND WISDOM

Mark Atkins

Florida Institute of Technology, USA

ABSTRACT

Two approaches are taken here in an endeavor to discover natural definitions of knowledge and wisdom that are justifiable with respect to both theory and practice, using graph theory: (1) The metrics approach is to produce graphs that force an increase in various graph metrics, whereas (2) the dimensions approach is based on the observation that the graphical representation of aggelia in the DIKW hierarchy seems to increase in dimension with each step up the hierarchy. The dimensions method produces far more cogent definitions than the metrics method, so that is the set of definitions proposed, especially for use in artificial intelligence.

KEYWORDS

Knowledge Representation, Artificial Intelligence, Graph Theory, DAG, DIKW

1. INTRODUCTION

Despite the emphasis on deriving and using knowledge in our modern era of data mining and applied artificial intelligence (AI), "knowledge" itself has never been well defined. While undefined fundamental terms are fairly common in math and the sciences (e.g., "point," "life," "intelligence"), lack of such definitions often impedes progress. Consider that if one cannot even answer the question ”What does it mean for a system to understand something?” then it would be difficult to design a system that understands anything at all, which in fact seems to be the current situation with all known types of computer hardware and software [1] [2], which is clearly impeding progress in applied AI. In addition to immediately practical benefits, such a definition of "knowledge" (and of its more abstract cousin "wisdom") could be of great benefit toward answering much heavier questions, especially in artificial general intelligence (AGI), since such an insight might provide clues about which Knowledge Representation Method (KRM) to use in AGI systems, especially since the KRM of human brains is not known and Edward Feigenbaum once considered this the most important problem in all of AGI [1].

One impediment to establishing such a definition has been the longstanding assumption that knowledge must be true and consistent, even though humans often hold false and inconsistent beliefs despite humans being the paragon of intelligence on this planet. The well-known definition from Plato that "knowledge is justified true beliefs" has recently been shown to be faulty, based on the hypothetical Gettier cases[3]. An underlying reason for the difficult of determining a useful definition is that knowledge is an intangible, abstract, theoretical construct [3]. A number of interesting metaphors to knowledge have been made [4], especially objects [5],

Natarajan Meghanathan et al. (Eds): SIPP, BIGML, DaKM, SOEN, AISC - 2020
pp. 59-74, 2020. © CS & IT-CSCP 2020 DOI: 10.5121/csit.2020.100205

energy, waves [4], and fluid flows, but these have lacked detailed descriptions or formulas. In general, modern research seems to have concentrated on truth values and metaphors rather than the data structures that might hold those truth values.

This lack of key definitions is somewhat surprising in computer subfields since it is generally understood that directed, acyclic graphs (DAGs) have great representational power, to the extent that many concepts from AI (e.g., expert systems, semantic networks, neural networks, bayesian networks) use DAGs to represent them, as do many concepts from traditional computer science (e.g., flow charts, various automata graphs, Entity Relationship Diagrams, fork-join diagrams, Petri nets, computer network diagrams, precedence diagrams, DeMarco data flow diagrams, logic circuit diagrams), as do the recurring key problems themselves that are common across many scientific fields (e.g., the travelling salesman problem, the subgraph isomorphism problem, the Hamiltonian path problem, the vertex cover problem, the clique problem, the graph coloring problem) that exemplify the P versus NP-problem, which is one of the most important unsolved problems in computer science. This situation is a major clue that if precise, useful definitions of knowledge and wisdom do exist, then those definitions are likely representable by some variation of DAGs. Another clue is that drawing pictures is a well-known problemsolving heuristic in mathematics [6]. These were the two main guiding heuristics that steered this study toward graph theory for a definition.

2. MOTIVATION

This study began in 2008 to provide a practical method to measure the amount of knowledge in knowledge-based systems versus databases for comparison purposes in the commercial software world. Such a method had not existed because practical definitions of the concepts of knowledge and wisdom have not existed (e.g., [7] [8] [9]), a preliminary problem that required solution before any such metric could be developed. The company in Westminster, California that motivated this study folded the same year, whereupon this potentially valuable study became shelved for over a decade.

3. BACKGROUND DEFINITIONS

DIKW hierarchy. The DIKW hierarchy describes the hierarchical relationship of the four main levels of data abstraction: Data, Information, Knowledge, and Wisdom, where Data is at the bottom level of this hierarchy, and Wisdom at the top. Usually this hierarchy is represented as a pyramid called the DIKW pyramid (Figure 1). Although the bottom two levels (Data and Information) of the pyramid are well understood, the top two levels (Knowledge and Wisdom) are not. Since this hierarchy sheds some light on the relationships of the elements in question, it is used as the starting point for this study.

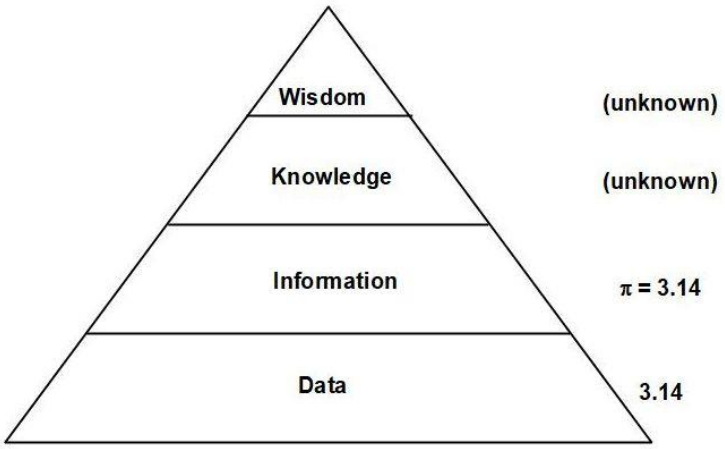


Figure 1. The DIKW pyramid

Aggelia. Let us define ”aggelia” (pronounced like ”ag-el-EE-ah”) as the general type of content in the varied levels of the DIKW hierarchy. In other words, aggelia denotes the information-like content the DIKW hierarchy holds, in a generic sense. Aggelia is an ancient Greek word that means ”message” or ”announcement.”

Dumbbells. Let us define ”dumbbell” generically as two vertices of a graph connected by an edge (Figure 2). The edge may or may not be directed, depending on need, and may or may not be labeled or weighted. A single dumbbell with a non-directed edge is equivalent to the simple path P_2 (e.g., [10]), K2 (e.g., [11]), or S2 (e.g., [10]), all well-known from graph theory. Dumbbells are often used as building blocks in networks of any kind, especially in networks of knowledge.

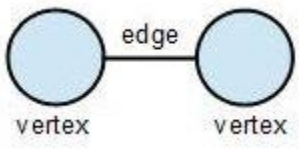


Figure 2. A single dumbbell with an undirected edge

WDAGs. Despite the general lack of understanding of aggelia, software developers in applied AI have pushed forward with temporary conventions about what knowledge and wisdom might look like to a digital computer in order to get practical results immediately. From both rule-based expert systems (RBESs) and neural networks standard data structures have arisen that contain a head, tail, and a numerical weight on the directed edge (Figure 3). In MYCIN this weight is interpreted as the amount of evidence for the assertion described via text at the tail (consequent) [12], and in neural networks this weight is interpreted as the strength of signal transmission to effect the desired mapping from head to tail. In graph theory such a network is called a weighted directed graph. Since applied AI data structures almost always remove or control cycles in networks (e.g., recurrent neural networks, the backpropagation algorithm) we can assume that all such graphs considered here contain no cycles, which then qualifies them as weighted, directed

acyclic graphs (DAGs), most commonly called ”edgeweighted DAGs,” but sometimes ”wDAGs,” or ”WDAGs”. For brevity we use the term WDAG here.

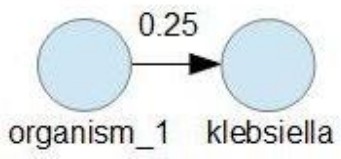


Figure 3. A WDAG is composed of dumbbells such as this.

LDAGs. In semantic nets a similar data structure exists, with the same textual labeling on head and tail that is common in RBESs and neural networks, but with textual labeling instead of numerical values on the edges (Figure 4), which in this case denotes the relationship between the two connected vertices. Let us abbreviate ”labeled DAG” as ”LDAG.”

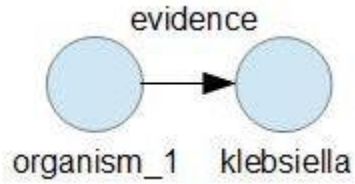


Figure 4. An LDAG is composed of dumbbells such as this.

LWDAGs. If a WDAG is combined with an LDAG then the result could be called an ”LWDAG,” which is the term used here. Since such a graph is remarkably general, LWDAGs will be the starting point for the modifications discussed here. Modern RBES languages such as CLIPS and JESS use the equivalent of such LWDAGs, although often enhanced by optional calculations in the head (= antecedent). Per graph theory the double edge containing both text and a number prevents the graph from being classified as a simple graph (e.g., [13]) but this can be overlooked in various ways, such as considering the number to be part of the text.

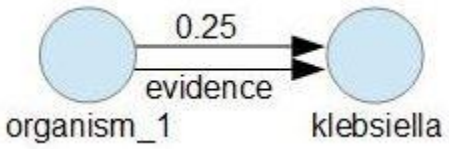


Figure 5. An LWDAG is composed of dumbbells such as this

4. DEVELOPMENT OF THE GRAPHICAL REPRESENTATION METHOD

4.1. Overall Strategy

In general we seek some attribute that would cause a graphical representation of one level of aggelia to qualitatively ”transcend” that layer naturally so that the result would be considered the next higher level of abstraction. This will be called ”qualitative transcendence” here. To gain

sufficient insights into the problem to initiate a promising research path we will use the following overall strategy of investigation:

1. make observations from data, information, and (to some extent) knowledge represented as LWDAGs, for initial insights and direction
2. take consideration of the two possible approaches to defining knowledge and wisdom

(a) to cause an increase in the values of metrics

(b) to increase dimensions

3. if any of those approaches are promising, propose the implied formal definitions

The reasoning for the non-consideration here of other known graph extensions is as follows. **K colored graphs** (e.g., [14]) are those with labels that are consecutive integers, which is a special case of textual labels, which were already selected above in labeled graphs, so the colored graph extension is redundant. **K-partite graphs** (e.g., [14]) have a constrained structure and in Petri nets can be used with added edges for special transition functions that ”fire,” but a KRM graph would likely have its descriptive power harmed by additional structural constraints, and its transitions have no obvious need to fire. **Marked graphs** are special purpose graphs such as Petri nets with tokens for modeling processes and states, which are a different purpose than representing aggelia as done here, so marked graphs do not extend the power of a graph as a KRM in a useful enough way here. Graphs with inputs, such as graphs representing discrete finite state automata (DFAs) or Turing machines are not needed since the graph is already representing possible input; **graphs with inputs** are mainly used to represent Turing machines, which are special purpose machines to represent states rather than aggelia itself. **Graphs with outputs**, such as Turing machines represented as graphs, specify some type of desired behavior such as printing, but behavior is merely an effect on the outside world that is already being sensed continually, and if there does exist an important adjunct behavior it should be explicitly described by itself.

4.2. Observations from Data and Information Represented as LWDAGs

Data. The nature of data is already well-known. Data typically consist of numbers or text that is unorganized or has no context. For example, the value 3 is data that could be a measurement of some kind, identification of some kind, computer memory address, an order within a list, or something else. Since data itself is typically of more interest than their organization or context, the latter of which are already understood by the implied organization or positions within a data structure, the convention here will be to represent data by vertices of a graph, and to represent relationships by the edges of that graph. In graph theory a single vertex is called an ”isolated vertex” (e.g., [14]) or ”trivial graph” (e.g., [11]), so an isolated vertex will represent a single datum here. There exists only one obvious method of measuring isolated vertices, and that is to count them, so both diagrams and measurement of pure data are trivial, other than possibly making a decision as to whether duplicated data should affect the count. (See Figure 6.)

Figure 6. amount of data = count of unique isolated vertices = 2

Information. The nature of information is also well-known. As soon as a datum like 3 is given a context such as ”x = 3”, or designated location of potential importance like ”memory[515] = 3”, the resulting expression is considered information and no longer data. The obvious resulting representation of information would be a labeled edge connecting the two related data vertices, all of which together become the LDAG defined earlier. (See Figure 7.) Note that even if two or more dumbbells share the same vertex, the counting method for information is the same: only dumbbells are counted, never individual vertices or edges.

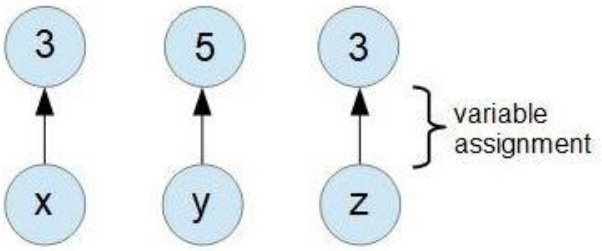


Figure 7. amount of information = count of dumbbells = 3

Knowledge. Clearly, knowledge would tend to be represented by collections of dumbbells. In an RBES these dumbbells may or may not all be connected while the RBES' corresponding rules wait for combinations of states that will fire them (detected by their head), though in a DFA where there exists a small set of all critically important states represented by vertices, all vertices would be connected head-to-tail in a network.

Nature has taught us that epiphenomena commonly arise when a high enough quantity has been reached of the underlying components, such as a high enough quantity of oxygen molecules giving rise to sound, or a high enough film frame rate giving rise to the perception of a moving picture, or a high enough quantity of fissile material giving rise to critical mass that produces a sustained nuclear chain reaction. Thus it is entirely possible that, per popular terminology, a vertex with a high enough count of adjacent vertices might be considered to have ”wisdom” rather than merely ”knowledge.” (See Figures 8 and 9.) More generally, any graph component—such as vertex, edge, dumbbell, group, or border—can have any measurable attribute—such as count, degree, length, distance, or nesting depth—and any applicable summary function can be used— count, average, maximum, minimum, and various ratios of those values. This will be Approach #1.

On the other hand, note that as soon as a few dumbbells have been connected to each other to form a tree structure instead of a list structure, the dimension of the resulting structure usually increases from 1-D to 2-D (except if the resulting graph happens to be a linear chain of nodes like P_n). This suggests that it may not be necessarily the specific number of dumbbells that have been connected that made the essential difference, but rather than they have been connected in such a way that they can no longer fit into their former dimension, since increasing a dimension could certainly be considered a form of qualitative transcendence. This will be Approach #2.

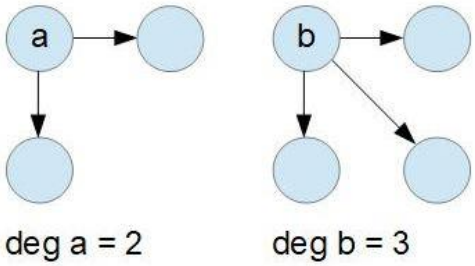


Figure 8. Does the graph on the right contain more "knowledge" than the graph on the left? Any "wisdom"?

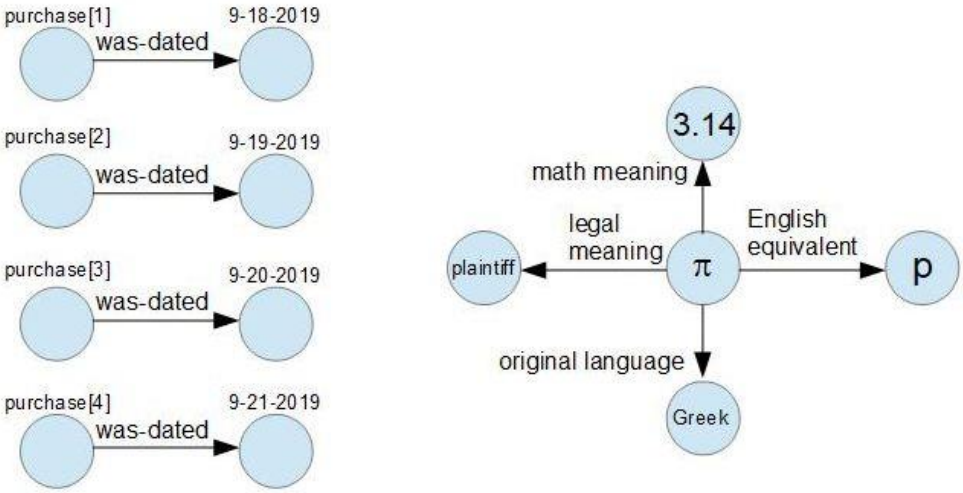


Figure 9. Does the graph on the right contain more "knowledge" than the graph o the left? Any "wisdom"?

5. CONSIDERATION OF THE TWO POSSIBLE APPROACHES

5.1. Approach #1: Increase in the Values of Metrics

If this approach were consistently applied, the following situations would ensue:

1. Enough data, represented as isolated vertices, could be considered information.

2. Enough information, represented as dumbbells, could be considered knowledge.

3. Enough knowledge, represented as a larger graph of dumbbells, could be considered wisdom.

Unfortunately, Situations #1 and #3 outright fail: (Situation #1) Since by definition we know data vertices must be isolated in order to lack associations, and since by definition information must be connected in order to contain associations, then more data can still be only data, not information; one form of aggelia cannot transform to the other via a change in quantity, in this case. (Situation #3) From common usage of the term ”wisdom” it becomes clear that likelihoods are normally

involved, even if those values are 100%, and even if the values are not stated. Some examples of folk wisdom are:

• Everyone must pay taxes. (Except the wealthy, or residents of The Bahamas or Bermuda.)

• Money can't buy happiness. (Unless your income is at the low end of the scale, especially in a poorer country [15].)

• (in chess) A knight on the rim is grim. (Except in Petrov's Defense Italian Variation or the Caro-Kann Defense Exchange Variation.)

A piece of wisdom is most commonly a generalization in a very complicated field such as life, business, love, psychology, biology, mathematics, natural language, music, art, or chess, where some heuristic is desirable to make headway among the myriad of possibilities, but where that heuristic is very rarely guaranteed true 100% of the time (else it would likely be a scientific law). This observation strongly suggests that the notion of wisdom equates to heuristics, each of which requires at least a likelihood value, and usually requires a large group of possibilities from which to obtain sufficiently supportive statistics. Since knowledge graphs as discussed so far contain neither groups of dumbbells nor weights on such groups, the common meaning of wisdom cannot apply to knowledge graphs without additional modification, therefore Approach #1 fails on creating wisdom from knowledge.

Although the above knowledge-is-much-information definition (Situation #2) could be forced to work, the fact that two of three such qualitative transcendence patterns do not hold means these definitions are not natural and consistent. Collectively the evidence says that Approach #1 is flawed for usage in natural definitions of transcendence to the next level of aggelia. However, the metrics used in Approach #1 will still always be useful for measuring the amount of aggelia in any single level of the DIKW hierarchy, if a definition of that given level can be supplied.

5.2. Approach #2: Increase Dimensions

Unlike Approach #1, it turns out that Approach #2 has an absolutely consistent pattern of qualitative transcendence across all levels of aggelia. The only trick is to find a way to group an entire section of a graph using a type of mechanism that does not already exist in the existing knowledge graphs, then to connect that group to an outside node via an arc. One solution is to group parts of graphs (or all of each graph) with a border, and then treat the border itself as a head to which a new edge can be attached. Such a border could be informally described as a "swollen head" or "swollen vertex" (Figure 10).

A graph "border" is a concept introduced here that is similar to the concepts of hypergraph and colored graph, but is subtly different from each. A hypergraph contains (hyper)edges that connect more than one vertex, but each hyperedge can connect only to a vertex, not to another hyperedge [16]. A colored graph also groups vertices, and it does so by giving each vertex in the group a color, but this color is not represented by a new entity within the graph to which a connection may be made [17], therefore a group cannot function as a type of vertex. Therefore a new mechanism is needed that not only groups vertices, but creates an edge from that group to another vertex. The border concept is therefore used for this purpose in this article. A "border" will be defined as a boundary line drawn around a set of vertices in a connected graph.

As originally suggested for Approach #2, each step up the DIKW pyramid does in fact now imply an increase of one dimension in aggelia representation when the border construct is allowed; all transitions now show qualitative transcendence as originally desired (Figure 11). In retrospect these dimension-based definitions make perfect sense because of their consistent pattern of achieving qualitative transcendence by creating connected groups of each previous data structure:

• information/dumbbells are connected groups of isolated vertices

• knowledge/graphs are connected groups of dumbbells

• wisdom/groups are connected groups of graphs

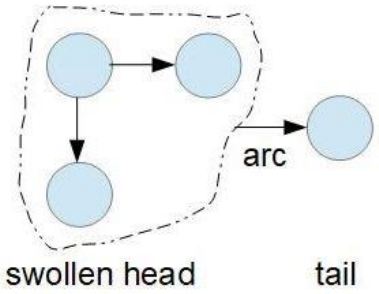


Figure 10. A group could be considered a "swollen head."

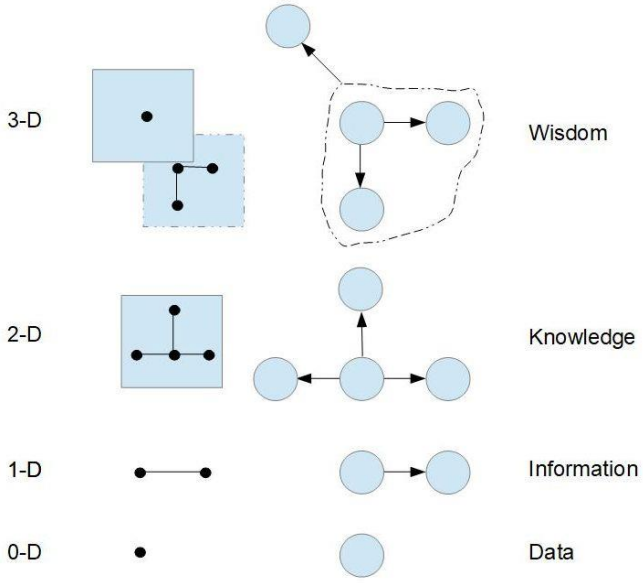


Figure 11. Each step up the DIKW hierarchy suggests an increase in one dimension in aggelia representation.

It appears that over the ages humans have abstractly sensed something qualitatively different about each of the four concepts in the four aggelia levels, viz. the different implied dimensions, to

the extent that humans gave each of those concepts a unique name, even though humans could not define two of the names well. This realization provides appealing psychological support for the proposed definitions. Since graph theory is considered a branch of mathematics (viz., discrete mathematics [18]), the proposed definitions are on solid mathematical ground, which makes the proposed definitions very fortunate for academics, as well.

This completes the two different approaches to defining the four levels of aggelia. A trick is used below for the proposed definitions: two sets of terms are used, one set that states mechanically that aggelia level x is defined as an ”x graphical unit,” where x is an element of the set data, information, knowledge, wisdom, then the more second set of definitions states that ”x graphical unit” is defined as the graphical unit at the dimension level associated with the proposed graph structure of x. In this way only the second half of the definitions need to be changed if later consensus decided that the proposed graphical associations were unsuitable as a foundation of definitions.

Note that certain fine points are left unresolved in the proposed definitions of this article: (1) whether the pathological path graph P3 or longer should be considered knowledge because it contains more than two vertices, or whether it should be considered not knowledge because it does need to occupy three dimensions, and (2) whether a non-planar graph should be considered

a 3-D graph only because it cannot be drawn in 2-D without crossing lines (probably it should not be considered 3-D, since its essential character remains unchanged). If the recommended definitions are actually used then practice would best determine the decisions for these questions.

6. IMPLIED FORMAL DEFINITIONS

6.1.General Definitions

Aggelia is the non-physical part of any description. The description can be numerical, textual, pictorial, a data stream, sensor input, a formula, or other. The generality of the description can be at any level. Aggelia is the general type of content of the DIKW hierarchy, which includes data, information, knowledge, and wisdom.

A **border** is a single closed loop drawn around selected items within a (usually larger) group to denote that all enclosed items are members of the desired group.

The **Aggelia Definitions Proposal** is a name of the set of proposed definitions in this article, for convenient reference in any later articles

6.2. DIKW-RELATED GRAPHICAL DEFINITIONS

A data graphical unit (DGU) is an isolated vertex.

An **information graphical** unit (IGU) is two units of data that have been associated with an edge, which may be directed, labeled, weighted, or any combination of those modifications. An information graphical unit is also called a dumbbell.

A **knowledge graphical** unit (KGU) is a graph that uses the vertices and edges of dumbbells, where the vertices of any connected dumbbells are shared.

A **wisdom graphical** unit (WGU) is an information graphical unit plus a set of data grouped by border where that group is associated with the data graphical unit via an edge. A wisdom graphical

unit is also called a swollen dumbbell, and the swollen end of it is called a swollen vertex.

6.3.DIKW-Related Aggelia Definitions

Data is aggelia whose structure is that of a data graphical unit.

Information is aggelia whose structure is that of an information graphical unit.

Knowledge is aggelia whose structure is that of a knowledge graphical unit.

Wisdom is aggelia whose structure is that of a wisdom graphical unit.

6.4.Some Useful Proposed Graph Metrics

Assume that G is the graph in question, B is a border in G, and L is a specified label. When measuring the amount of various levels of aggelia in a graph, especially when using ratios and percentages, the following metrics are predicted to work well with the above definitions.

depb(B) = the depth of border B = the count of border crossings for nested borders for B

avgdepb(G) = the average depth of borders in G

bor(G) = the count of borders in G

alab(G) = the count of all unique labels in G

slab(L, G) = the count of a specific label L in G

7. IMPLICATIONS OF THE DEFINITIONS

If the proposed definitions, or some variation of them, became accepted then the theoretical and practical implications would be extensive and important. Some such possible implications are listed below.

Note that one non-implication is influence upon Shannon's information theory, or vice versa. Information theory is based on probabilities and real-life situations (e.g., [19]), whereas the graphical representations here are assumed to be 100% certain. In the current context, to apply probability to the components of a knowledge graph would make as little sense as applying probability theory to a human-designed DFA or to the symbols of a mathematical formula.

7.1. Historical

This would be the first time in over 2,000 years that these concepts, which were debated by Plato and Socrates, were mathematically formalized.

7.2. Terminological

Many frequently used computer terms would be perceived as technically erroneous. For example:

• ”databases” would be more accurately called ”information bases”

• ”data structures” (of computer science) would be more accurately called ”information structures”

• ”knowledge bases” (of applied AI) with heuristics would be more accurately called ”wisdom bases”

7.3.Practical

Objective, numerical comparisons could be made between knowledge base versus knowledge base, or even between more disparate entities such as knowledge base versus database. The proposed metrics, or variations of them, would be used to determine which repository contained more data, information, knowledge, and wisdom, and even the quality of the wisdom.

For example, consider a single table database with 2 records and 3 fields, where the records were instances of people, and the fields were name, phone number, and e-mail address. This database would be equivalent to the following dumbbells with non-directed edges, each dumbbell of which is represented here as an object-attribute-value (OAV) triplet:

(person1 name name1)

(person1 emailaddress emailaddress1)

(person1 phonenumber phonenumber1)

(person2 name name2)

(person2 emailaddress emailaddress2)

(person2 phonenumber phonenumber2)

This database is therefore equivalent to 2 x 3 = 6 dumbbells, or 6 IGUs. In general, disregarding compound keys and non-visible indices, a single table database of R records and F fields contains R x F information graphical units.

As a practical example of how databases can be compared with knowledge bases using the methods of this article, consider Figure 9. In answer to the question posted in the caption of

Figure9 the unconnected graph on the left that represents a database contains 4 information graphicalunits, but contains no graphs that require 2-D space, therefore it contains zero knowledge, onlyinformation. This is exactly what one might expect from a database. In contrast, although thegraph on the right also contains 4 information graphical units, the graph as a whole represents

knowledge–a higher level of abstraction–which is exactly what one might expect from a knowledge base. Neither Figure 8 nor Figure 9 contain any wisdom since neither has any groups.

7.4.An Architecture that ”Understands”

The meaning of ”understanding” would possibly become completely understood and therefore implementable on a machine. For example, regarding the question in the caption of Figure 8, the answer would be that graph ”b” (the graph with vertex ”b”) does in fact contain more knowledge than graph ”a” because graph ”b” contains one more dumbbell than the other: 3 instead of 2. To understand why this claim makes sense in a practical case, reconsider Figure 9, which would represent ”understanding” of the symbol π. Someone who understood that π could also mean ”plaintiff” and not just the well-known mathematical constant would be considered to have better ”understanding” of the symbol π–what it can mean and what it can do–and this better understanding would be solely due to the presence of that one extra dumbbell. If all four components of aggelia are represented simultaneously as a vector [D I K W], then graph ”a” contains [3 2 0 0] graphical units, and graph ”b” contains [4 3 1 0] graphical units–one more IGU than graph ”a”.

A very striking observation is that this example likely generalizes to an extreme degree, to the extent that it may even explain the key to commonsense reasoning. This seems possible because the concept of ”understanding” concept x seems to mean only that all immediately relevant attributes of x are ”felt” or considered simultaneously as x is considered, and since attributes would be represented graphically as the distal vertices of dumbbells with central vertex x, then if vertex x were active, as in a firing neuron, all the simultaneous associations exactly one arc distantwould also be activated (Figure 12). Since neurons have an extreme number of connections, about 104, and since the number of associations necessary to truly understand concept x would also belarge, this suggests that a graphical star (e.g., [10]) whose head and tails represented concepts could well implement a neural network capable of true understanding. Since it is clear the brain has an attention focusing mechanism, called an ”attentional spotlight” [20], the activated central vertex x of a star would implement this focusing mechanism in a simple way. Similarly, ”attentional shift” or ”train of thought” would be implemented simply by activation shifting from the vertex x to an adjacent vertex. In other words, understanding happens automatically in a graph with a single node that serves as the attentional spotlight, and the more adjacent nodes the better it understands. Thinking would then likely equate to some type of controlled, goal-directed activation of such concept nodes. These ideas about the meaning of understanding date back to at least year 2000 [21].

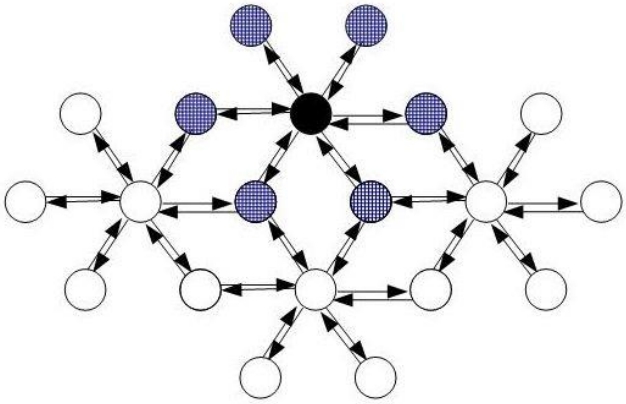


Figure 12. Conjecture: The central node of the active star ”understands” that node’s semantic concept via simultaneous consideration of all associations that are exactly one arc distant.

7.5.Commonsense Reasoning

As mentioned above, AGI systems would be more likely to be designed that would naturally support commonsense reasoning. This could be extremely important because commonsense reasoning was believed by John McCarthy to be one of the key problems of AGI [22].

It seems clear that some sort of subconscious, neural statistics gathering happens in the human brain. During the process of experiencing the real world, huge volumes of real-world information pass through the brain such that common spatiotemporal patterns are gradually learned, though only subconsciously. The result is a statistical prediction of what will follow in time when an early part of a given spatiotemporal pattern is presented to the brain. For example, if a typical object being held in the air is released under typical conditions, the stored statistics predict that the object will fall. Whatever representation the brain uses for the geometrical translation operation is evidently applied to the current object, and a fast but crude simulation then automatically ensues that predicts the outcome, visually.

The group construct that allows graphical implementation of wisdom, as shown earlier in Figure 10, would be nearly ideal for collecting such statistics. The group itself could include only vertices (neurons) that experienced a given event (e.g., releasing), so the weight on the outgoing arc couldbe a statistical summary of what percentage of the time the event represented by the tail vertex (e.g., falling) ensued. In effect, the graph ”understands” not only concept x that it is currently perceiving, but also ”understands” what is likely to happen next. Rather than having a programmer estimate the weight on that arc, the system could estimate that weight based on its own experience (i.e., by counting its own nodes inside that group and calculating the ratio of have-fallen versus have-not-fallen nodes), and better yet might well be able to recall every single event that contributed to that weight, something that a programmer would not likely have the time or ability to encode. In this way wisdom is closely associated with commonsense reasoning, which in turn is closely associated with understanding, and since wisdom would now be understood and therefore implementable, so would understanding and commonsense reasoning, which could lead to a milestone advance in AGI.

7.6.Beyond Wisdom

A natural question to ask when studying the DIKW pyramid is, ”Could there exist a level of higher abstraction than wisdom?” Using the graphical methods of this article, the answer is no, because there does not exist any obvious extension that cause wisdom to undergo qualitative transcendence, no matter how components are connected or grouped.

However, certain real-world applications suggest such a higher level might be useful in practice. This situation would arise naturally, for example, if a country’s economic flow were modeled with a geographical representation such that instead of points of origin there were regions of origin on a map of that country, which would force the arcs to be 3-D and to change in cross-ectional density to represent the weight at each point in continuous 2-D space. Such 3-D arrows may or may not terminate in a single point representing a concept. However, if such arrows both started and ended in 2-D space, that would create a 4-D mapping, which would require yet another dimension of representation. Whether such an application would justify a new name for a fifth aggelia level is debatable, but since the cerebellum typically stores such seemingly continuous mappings of vectors (e.g., [23]) during creation of its own brand of fine-tuned motor-coordination ”knowledge,” eventually we would want intelligent machines to be able to do the same thing.

8. CONCLUSIONS

The metrics approach to producing qualitative transcendence in graphical representations of the different levels of aggelia is not consistent, but the dimensions approach of Figure 11 is completely consistent, therefore the increasing dimensions approach’s implied definitions of knowledge and wisdom are the definitions proposed here. Those definitions are not only based on solid mathematics in the form of graph theory and dimensions, but are psychologically justified since there exists a qualitative difference between aggelia levels with those definitions.

9. FUTURE WORK

The given example of how to measure aggelia in a 1-table database could likely be extended easily to an n-table database. A complication that might be interesting to pursue for future research is mappings of vectors to vectors, which would create a fifth level of aggelia, a level that would require 4-D representations.

REFERENCES

[1] H. L. Dreyfus, What Computers Can't Do, Revised Edition: The Limits of Artificial Intelligence, New York, N.Y.: Harper & Row, 1979.

[2] J. Hawkins, On Intelligence, New York: Times Books, 2004.

[3] E. Bolisani and C. Bratianu, “The elusive definition of knowledge,” in Emergent knowledge strategies: Strategic thinking in knowledge management, New York City, New York, Springer International Publishing, 2018, pp. 1-22.

[4] D. Andriessen, “On the metaphorical nature of intellectual capital: A textual analysis,” Journal of Intellectual Capital, vol. 7, no. 1, pp. 93-110, 2006.

[5] T. Davenport and L. Prusak, Working Knowledge, Boston: Harvard Business School Press, 2000.

[6] G. Polya, How To Solve It: A New Aspect of Mathematical Method, Second Edition, Garden City, New York: Doubleday & Company, 1957.

[7] M. Nagao, Knowledge and Inference, San Diego, CA: Academic Press, 1990.

[8] R. J. Brachman and H. J. Levesque, Knowledge Representation and Reasoning, San Francisco, CA: Morgan Kaufmann Publishers, 2004.

[9] H. J. Levesque and G. Lakemeyer, The Logic of Knowledge Bases, Cambridge, Massachusetts: The MIT Press, 2000.

[10] R. Merris, Graph Theory, New York, NY: John Wiley & Sons, 2001.

[11] J. Gross and J. Yellen, Graph Theory and Its Applications, Boca Raton, Florida: CRC Press, 1999.

[12] G. F. Luger and W. A. Stubblefield, Artificial Intelligence and the Design of Expert Systems, Redwood City, California: The Benjamin/Cummings Publishing Company, 1989.

[13] J.-C. Fournier, Graph Theory and Applications, Hoboken, NJ: John Wiley & Sons, 2009.

[14] G. Lesniak and L. Chartrand, Graphs & Digraphs, Fourth Edition, Boca Raton, Florida: Chapman & Hall/CRC, 2005.

[15] M. Argyle, The Psychology of Happiness, 2nd Edition, New York, NY: Taylor & Francis, 2001.

[16] C. Berge, Graphs and Hypergraphs, New York, NY: American Elsevier Publishing Company, 1973.

[17] D. I. Moldovan, Parallel Processing: From Applications to Systems, San Mateo, California: Morgan Kaufmann Publishers, 1993.

[18] D. F. Stanat and D. F. McAllister, Discrete Mathematics in Computer Science, Englewood Cliffs, N.J.: Prentice-Hall, 1977.

[19] S. Theodoridis and K. Koutroumbas, Pattern Recognition, Third Edition, San Diego, CA: Academic Press, 2006.

[20] L. C. Robertson, Space, Objects, Minds, and Brains, New York, NY: Psychology Press, 2004.

[21] M. Atkins, S-96: A Semantic Net Implemented With Synchronized Neurons for Binding and Inferencing. Phd Diss., Melbourne, Florida: Florida Tech, 2000

.

[22] E. Davis, Representations of Commonsense Knowledge, San Mateo, California: Morgan Kaufmann Publishers, 1990.

[23] P. S. Churchland, Neurophilosophy: Toward a Unified Science of the MindBrain, Cambridge, Massachusetts: MIT Press, 1989

FROM QUALITY ASSURANCE TO QUALITY ENGINEERING FOR DIGITAL TRANSFORMATION

Kiran Kumaar CNK

Capgemini India Private Limited, Inside Divyasree Techno Park, Kundalahalli, Brookefield, Bengaluru, Karnataka 560037, India

ABSTRACT

Defects are one of the seven prominent wastes in lean process that arises out of the failure of a product or functionality from meeting customer expectations. These defects, in turn, can cause rework and redeployment of that product or functionality again, which costs valuable time, effort, and money. As per the survey, most of the clients invest much time, energy, and money in fixing production defects.

This paper provides information about ways to move into quality engineering from quality assurance mode for digital transformation by diagnostic, Predictive & Prescriptive approaches, it also outlines the overall increase in quality observations, given QA shift left and continuous delivery through Agile with the integration of analytics and toolbox.

KEYWORDS

Diagnostic, Predictive & Prescriptive approaches, continuous delivery through Agile

1. INTRODUCTION

In this contemporary world, the quality of the product determines the furthering and sustaining of any software business. Currently, quality assurance (QA) plays a continuous and consistent role in improving the QA process in most of the software businesses to ensuring that quality of the product, i.e., is enhanced by reducing and eliminating defects as clients invest much time, effort and money in fixing defects. This process of quality assurance is achieved by applicationlevel testing, test automation, inward-focused and descriptive mode.

On the other hand, for the actual digital transformation, quality of the product must be concatenated and engineered rather than just only being assured. It is chieved by migrating from quality assurance to quality engineering. The framework of quality engineering comprises of customer-focused, diagnostic, predictive, prescriptive, Shift left & continuous delivery through agile.

Natarajan Meghanathan et al. (Eds): SIPP, BIGML, DaKM, SOEN, AISC - 2020
pp. 75-82, 2020. © CS & IT-CSCP 2020 DOI: 10.5121/csit.2020.100206

The main objective of implementation of quality engineering are as follows,

a. QA role in agility and DevOps

b. Integration of Analytics

c. Test driven development in agile mode

d. Structured Testing process & Adaptivity

2. QA ROLE IN 'AGILITY' AND 'DEVOPS'

Agile refers to an incremental and iterative approach that focuses on collaboration, continuous customer feedback, and small, rapid releases. DevOps is a concept of executing end to end engineering processes that focuses on constant testing and delivery, which brings in both development and operation team together.

In this environment of agility in combination with DevOps, the tester performs quality engineering and assurance in all stages of the Agile development phase as opposed to participating only at the end of the cycle. During Requirement Analysis & Grooming phase (Stage1), QA can synergize with Business Analysts to picture the client's viewpoint, to ensure quality at the requirement level, also to ensure that acceptance criteria have covered all validations points and to eradicate defect ambiguity profoundly. In the Planning & Estimation phase (stage 2), QA plays a vital role through effective communication & brainstorming that helps in arriving at accurate estimations and in the effective discharge of SCRUM events, including DEMO.

In the next Functional Analysis & Design phase (Stage 3), QA plays a pivotal role to prevent defects upfront, Which could be thoroughly established by keen story/ requirement analysis, understanding junit coverages from Dev team, Designing high-level test scenarios and providing walkthrough of the same to all stakeholders like BA, Dev, Architect, Product owner, etc., on a scheduled 'Test scenario walkthrough meet' and finally coming out with a master test plan document i.e., inclusion of unit, SIT and Regression test scenarios/cases based on the discussion which binds insights and inputs that brewed out in the 'Test scenario walkthrough meet'. This way, we ensure that all stakeholders are on the same page, removing discrepancies in approaches and eliminating requirement ambiguousness. Thus, filling up the gap in understanding in advance and effectively proceeding with a concrete plan.

In the implementation phase (stage 4), QA can effectively perform technical normalization by performing API / web service testing or smoke testing in Dev Env, to further minimize the proximity of defects. Finally, in the quality assurance phase (Stage 5), QA can engineer and assure the quality by incorporating different layers of testing like Build Verification Testing (BVT, without which QA do not proceed with SIT), full-fledged SIT, Regression (mostly automated) and smart testing like crowd testing. Refer to figure 1 for an overview of testing types and corresponding responsibilities. [1] [2]

Activities	Testing Types	DEV/QA
Unit testing	Unit testing is done by the developer in Local environment	DEV/QA
Smoke Testing	Smoke testing of developed system, by the developer, on the environment provided for testing post deployment	DEV/QA
Build Verification testing	To validate whether the basic functionalities are working as expected (if not, to reject the build and will not proceed to SIT)	QA
SIT	To carry out the complete Functional testing	QA
Regression	To validate that the existing functionalities are not impacted by the addition of new functionalities and defect fixes	QA/Dev

Figure 1. Testing types and Responsibilities

3. INTEGRATION OF ANALYTICS

In this context, analytics enables testers to evaluate the performance of the test outcome. It can be tracked with the help of various potential metrics and parameters involved in the process of a test engineering exercise. In other words, the health quotient of the software product that is being tested is well being measured. The final status of the test results provides us with a perfect picture of the state of functionalities of the tested software. Refer to figure 2 to have an overview of prominent test analytics [3]

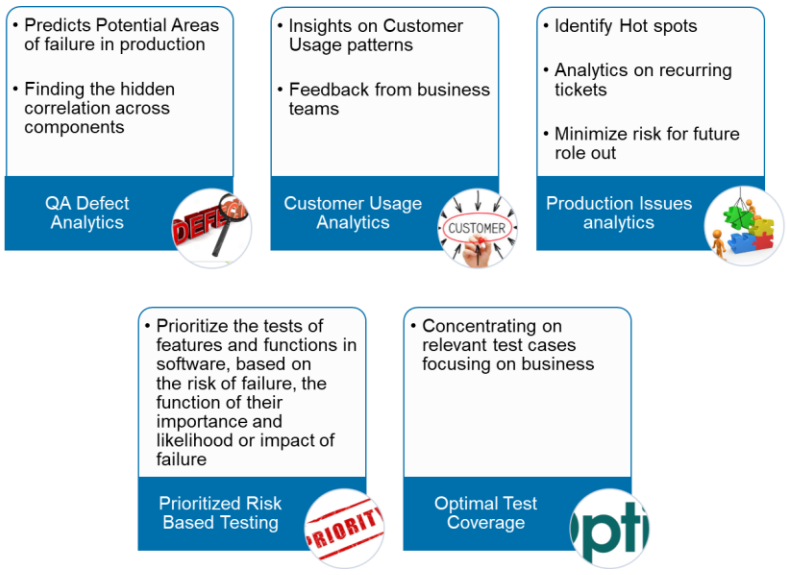


Figure 2. Prominent test analytics

4. TEST DRIVEN DEVELOPMENT IN AGILE MODE

Test-driven development (TDD) is one of the common practices of Agile core development. It is one of the engineering techniques for developing the software in collaboration where the code of programming design and its corresponding testing are executed in series of micro-iterations. It is an evolutionary approach that combines test first – Development next approach. The very purpose of this TDD is to focus on customer specification and not an end-phase validation primarily. Refactoring is also associated with this process, and it plays an important role in restructuring the code piece. Further, the tests should succeed to reduce complexity and enhance its understandability, maintainability, and clarity. Refer to figure 3 for an overview of the test-first development process. There are two levels of TDD. They are as follows, [4]

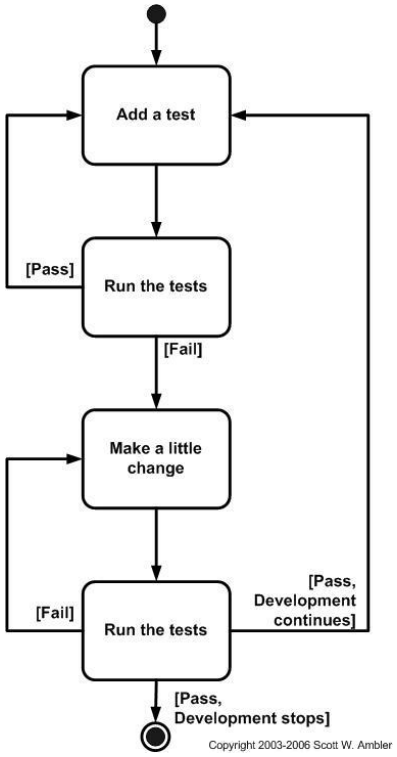


Figure 3. Test-first development process

4.1. Acceptance TDD (ATDD)

With ATDD you write a single acceptance test. This test fulfils the requirement of the specification or satisfies the behavior of the system. After that, write just enough production/functionality code to fulfil that acceptance test. The acceptance test focuses on the overall behavior of the system. ATDD also was known as Behavioural Driven Development (BDD).

4.2. Developer TDD

With Developer TDD you write a single developer test i.e., unit test and then just enough production code to fulfill that test. The unit test focuses on every small functionality of the system. Developer TDD is simply called as TDD.

The main goal of ATDD and TDD is to specify detailed, executable requirements for your solution on a just in time (JIT) basis. JIT means taking only those requirements into consideration that are needed in the system. So, increase efficiency and providing accurate coverage. Refer to figure 4 to know how acceptance TDD and development TDD work hand in hand.[4]

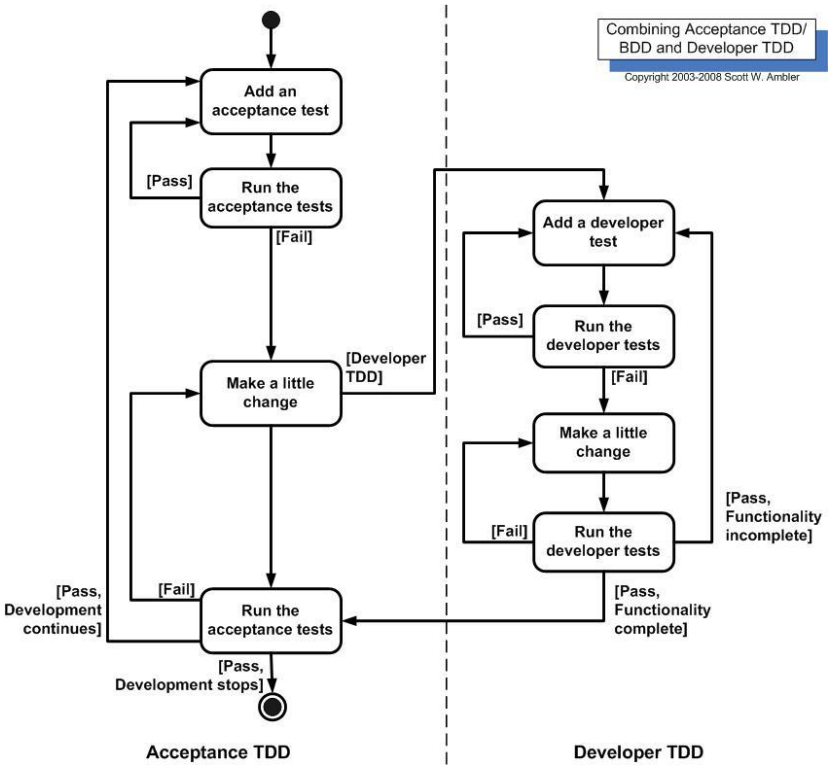


Figure 4. How acceptance TDD and developer TDD work together

5. STRUCTURED TESTING PROCESS & ADAPTIVITY

The structured testing process is one in which we would come up with a master test plan that comprises of test scenarios and test cases of unit testing, SIT and UAT managed together. The primary objective of designing a master testing plan document is to enable inhibition-free coordination and communication between different testing levels. By managing all testing activities on the same platform, we would be able to have a check on a few major aspects. First, it provides transparency between different phases of testing as we get to know what exactly covered as part of each testing phase. This way, we could carry out precise validation rather than exhaustive testing with redundancies. Second, we could eliminate any miss in scenario coverage

with a preconceived assumption. Finally, it further strengthens the core principle of agile. i.e., collaboration. Figure 5 illustrates the structured testing process.

Adaptivity is more of a combating process where testers from all phases pool together and learn from experience and response to changes instantly. It also ensures the QA team of different phases to stay on the same page with a clear understanding of in-scopes and out of scopes. This would eventually enable QA to provide strategic inputs on business functionalities on further development. Figure 6 illustrates the flow of adaptivity.

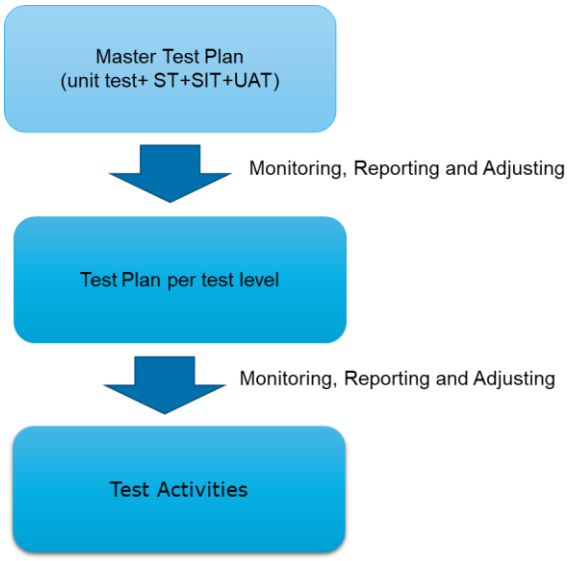


Figure 5. Structured testing Process

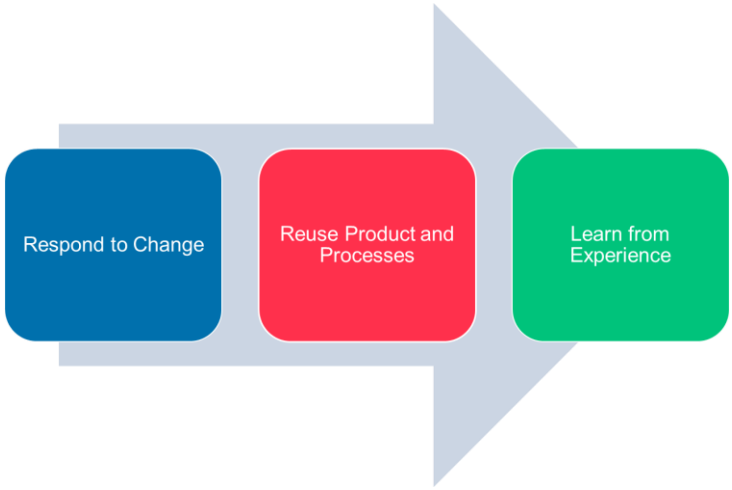
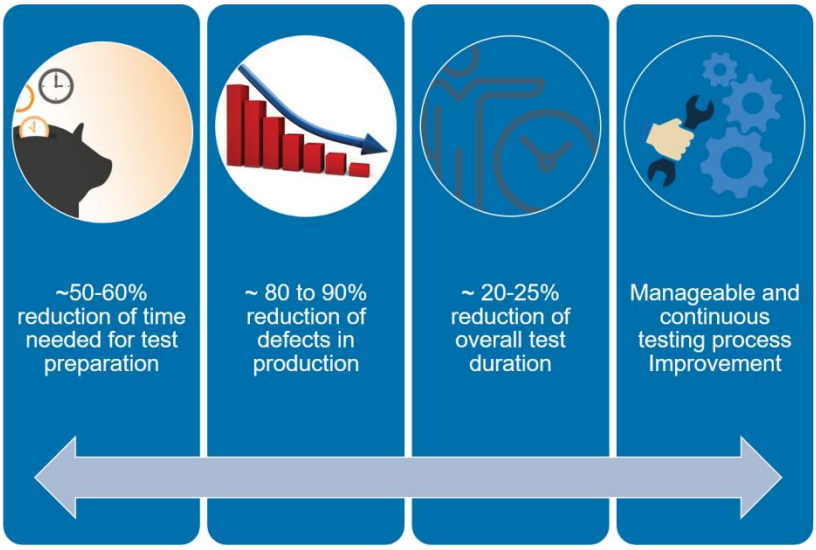


Figure 6 illustrates the flow of adaptivity

6. RESULTS FROM THE PILOT PHASE

Having run a pilot phase by implementing this approach across two digitally transformed projects of team capacity 17 and 9 each in the automotive domain, we have observed the below illustrated results. As it goes by, there is almost 50 – 60 % of the reduction of overall time spent in the test preparation phase that includes data settings, test scenario design, test case design, review and review comment incorporation. Next, to be keenly observed, there is almost an 80 to 90% reduction in the defect leakage in production environment due to systematic working in a preventive defect mode rather than defect deduction mode. Also, almost 20 – 25 % of the time-saving in the overall test duration. Finally, this approach persists in providing us with a channel to manage and improve the testing process periodically and dynamically.



7. CONCLUSIONS

Hereby I conclude that by employing the above suggested quality engineering framework, we can dramatically increase the quality of the deliverables. This model also strongly advocates for the saying '(Defect) Prevention is always better than cure.' This, on the other hand, helps in continuous testing process improvement and in testing advancements like automation, Robotic Automation process, etc. Finally, the tester gets transformed as 'Quality Engineer and assurer' by acquiring complete knowledge from functional, business, and technical spheres.

8. ACKNOWLEDGEMENTS

I would like to thank all my peers, managers, and mentors for their support, directly and indirectly, helping me to in making this article

9. REFERENCES

[1]Tridibesh Satpathy, (2017) A Comprehensive Guide to Deliver Projects using Scrum -Third Edition, SCRUMstudy™, a brand of VMEdu, Inc.

[2]Saliya Sajith Samarawickrama, (2018) "Continuous scrum: a framework to enhance scrum with Devops, 2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer)IEEE

[3]Harsh Vardhan (2019), "Applying Data Analytics to Test Automation", Stickyminds, TechWell Corp.

[4]Max Guernsey Iii, (2013) Test-Driven Database Development: Unlocking Agility-1st Edition, Net Objectives Lean-Agile Series, Addison-Wesley Professional

AUTHORS

Kiran Kumaar CNK works as a Senior Agile Quality Engineer with Capgemini India Pvt Limited. Currently, he is working on an onsite assignment at Capgemini, Netherlands. He holds B. Tech & MBA (Executive program in project management). He possesses over 10 years of experience in IT industry, predominantly in Insurance, automotive and public sectors. He is a certified Scrum Master and certified in ISTQB (advanced level) as well. Unequivocally, he possesses good presentation and communication skills that allowed him, from offshore, efficiently collaborate with different onshore clients and stakeholders across countries like US, Germany, Poland, Morocco, France, UK & Netherlands.

DESIGN OF SOFTWARE TRUSTED TOOL BASED ON SEMANTIC ANALYSIS

Guofengli

Faculty of Information Technology, Beijing University of Technology, Beijing, China

ABSTRACT

At present, the research on software trustworthiness mainly focuses on two parts: behavioral trustworthiness and trusted computing. The research status of trusted computing is in the stage of active immune of trusted 3.0. Behavioral trustworthiness mainly focuses on the detection and monitoring of software behavior trajectory. Abnormal behaviors are found through scene and hierarchical monitoring program call sequence, Restrict sensitive and dangerous software behavior.

At present, the research of behavior trust mainly uses XML language to configure behavior statement, which constrains sensitive and dangerous software behaviors. These researches are mainly applied to software trust testing methods. The research of XML behavior statement file mainly uses the method of obtaining sensitive behavior set and defining behavior path to manually configure. It mainly focuses on the formulation of behavior statements and the generation of behavior statement test cases. There are few researches on behavior semantics trustworthiness. Behavior statements are all based on behavior set configuration XML format declaration files. There are complicated and time-consuming problems in manual configuration, including incomplete behavior sets. This paper uses the trusted tool of semantic analysis technology to solve the problem of behavior set integrity And can generate credible statement file efficiently

The main idea of this paper is to use semantic analysis technology to model requirements, including dynamic semantic analysis and static semantic analysis. This paper uses UML model to automatically generate XML language code, behavioral semantic analysis and modeling, and formal modeling of non functional requirements, so as to ensure the credibility of the developed software trusted tools and the automatically generated XML files. It is mainly based on the formal construction of non functional requirements Model research, semantic analysis of the state diagram and function layer in the research process, generation of XML language trusted behavior declaration file by activity diagram established by model driven method, and finally generation of functional semantic set and functional semantic tree set by semantic analysis to ensure the integrity of the software. Behavior set generates behavior declaration file in XML format by the design of trusted tools Trusted computing is used to verify the credibility of trusted tools.

KEYWORDS

behavior declaration, behavior semantic analysis, trusted tool design, functional semantic set.

Natarajan Meghanathan et al. (Eds): SIPP, BIGML, DaKM, SOEN, AISC - 2020
pp. 83-90, 2020. © CS & IT-CSCP 2020 DOI: 10.5121/csit.2020.100207

1. INTRODUCTION

Behavior declaration refers to the collection of application software descriptions for its sensitive behaviors. In this set, sensitive behaviors include behaviors that may infringe the user's wn rights, behaviors that may affect the normal operation of the application software, and behaviors that may cause unexpected configuration changes of the software and hardware environment.

The generation of trusted behavior statement is implemented in the implementation phase of software. Through the definition of trusted behavior declaration completed in the software design phase, XML In the software requirement analysis stage, the software credible requirement is acquired; in the software design stage, the definition and design of the trusted behavior statement is carried out; in the software implementation stage, the preparation of the trusted behavior statement is carried out; in the software testing stage, the behavior statement is carried out Verification and improvement.

The research of software behavior in semantic analysis is at the forefront stage. Qu Yanwen put forward the concept of software behavior semantics. From the perspective of behavior semantics, the software function is complete. The main body traverses the whole behavior tree to ensure the integrity of function connection and prevent the lack of software function. Yang Xiaohui and others used API call analysis and instruction execution analysis to conduct behavior semantic analysis, and established a corresponding relationship between network data and malware behavior. It can be seen that simply analyzing the program or behavior sequence, ignoring the internal relations, will cause the lack of behavior expression information, can not meet the development of behavior analysis, modern modeling towards the direction of functional semantic analysis. Fu Jianming successfully parsed the system object from the system call parameters, giving the meaning of state semantics. Because semantics itself has its inherent logic, and is closer to the actual operation of users, and can find more hidden application layer attacks, so semantic analysis has gradually become the focus of current software behavior and network behavior researchers.

The research content of this paper uses the method of semantic analysis of software behavior to constrain the functional requirements. The UML use case diagram, activity diagram and state diagram generated by the demand modeling are used to produce the functional semantic tree, and then the functional semantic set is generated. The behavioral statement file is generated by the trusted operation behavior set, so as to avoid the lack or attack of the developed software function and ensure the reliability of the software.

2. SCHEME DESIGN:

Design ideas of trusted tools

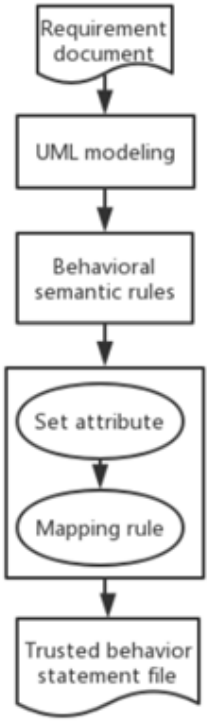


Figure 2-1Schematic diagram of trusted tool development process

2.1. Trusted Requirement Stage

According to the traditional requirement analysis process. Acquisition of non functional requirements. It is also necessary to obtain credible requirements for credibility.

2.2. Design of Behavior Statement Document

Statement of credible behavior

Statement file of trusted behavior: 1. Describe expected behavior of application software; 2. Implementation protocol of trusted software; 3. Evidence of trusted verification; 4. Auxiliary other documents

2.3. Template Design of Trusted Behavior Statement

This paper uses XML based format as the description and expression of trusted behavior statement, and its structure is consistent with the structure of trusted behavior statement file described above. The general definition of behavior statement is as follows:

< behavior declarationlist> / / behavior list
< behavior name one > / / represents a behavior
< behavior ID >< behavior ID > / / behavior number
< behavior item one > * * * < / behavior item one > / / behavior sub item 1
< behavior item two > * * * < / behavior item two > / / behavior sub item 2
... .
< credit level > dangerous < / credit level > / / credit level
</Behavior Name One>
< Behavior Name Two >
...
Other behaviors
< /Behavior Declaration List >

In the above example, < behavior declaration list > represents a list of all behaviors contained in the behavior declaration file. It contains multiple behavior items < behavior name XXX >, and multiple behavior sub items < behavior item XXX > constitute the specific operation and flow of the behavior item. Each behavior item must have corresponding trust level (such as danger, suspicion, trust, etc.), and the < security level > is used in the trusted behavior statement.

As an important parameter of trusted statement, the trustworthiness level clearly defines the trustworthiness of this behavior item, which provides an extremely important basis for trusted design, implementation and testing.

2.4. Credibility Detection of Behavioral Semantic Rules

If each function has credibility, the validity is tested. First, record the function trace, and then check whether it deviates from the function tree. If the function trace is < connect, verify, disconnect >, which is a path in the semantic tree, then the behavior has validity; if the function trace is < connect, communicate, disconnect, bypass the verification state, then there is an exception. The function level detection is based on the state level detection for software function semantics. Each function behavior should conform to the definition of function semantics set, and the function transformation should not exceed the function semantics tree. In this paper, combining the characteristics of software behavior, the function semantics rules are formulated to detect software behavior.

3. DESIGN AND IMPLEMENTATION OF TRUSTED TOOLS

Through the description of the first two chapters, the UML activity diagram is generated from the trusted requirement acquisition, and then the semantic analysis traverses the activity diagram to obtain the functional semantic set and the functional semantic tree. The software function defects are determined by rules, the software function semantic set and function semantic tree are divided, the system call sequence is used to ensure the ergodic integrity of the software call, and finally the software function integrity is guaranteed, and the software function credibility and complete semantic set data are used to develop the credible software behavior statement file.
Through a GUI tool generation, UML activity diagram behavior sets and attributes generate XML language form files,

Trusted tool design

GUI design mainly uses Tkinter module of Python third-party library to design gui. According to the template rules, the UML activity diagram is mapped to behavior set and attribute, and the XML file of behavior statement is generated

trusted statement tool

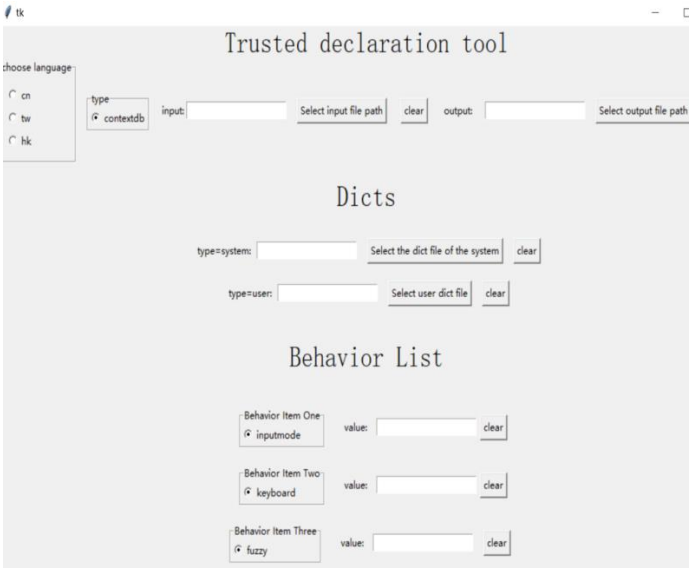


Figure 3-1schematic diagram of trusted declaration tool

4. GENERATE XML TRUSTED BEHAVIOR DECLARATION FILE

```
<?xml version="1.0" encoding="utf-8" ?>
<!-- TODO: xml schema definition -->
  <login_List>
   <Behavior Name_One>
    <Behavior Id>10001</Behavior_Id>
    <Behavior Item_One>Operation</Behavior_Item_One>
       <Behavior Item_Two>login</Behavior_Item_Two>
       <Behavior Item_Three>username</Behavior_Item_Three>
       <Behavior Item_Four>password</Behavior_Item_Four>
     <Credibility Level>Dangerous</ Credibility_Level >
</Behavior_Name_One>
<_Behavior_Name_Two_>
<Behavior Id>10002</Behavior_Id>
    <Behavior Item_One>Operation</Behavior_Item_One>
       <Behavior Item_Two>logout</Behavior_Item_Two>
       <Behavior Item_Three>username</Behavior_Item_Three>
       <Behavior Item_Four>password</Behavior_Item_Four>
     <Credibility Level>credit</ Credibility_Level >
</Behavior_Name_Two>
</login_List >
```

Figure 3-2 Result Display of XML Behavior Declaration File Generation

5. EXPERIMENT SIMULATION AND RESULT ANALYSIS

4-1 table of experimental environment

development tool	IntelliJIDEA	development language	Java
operating system	CentOS 6.5	Tomcat	8.0
JDK	1.8	Mysql	5.7

Experimental case data: two kinds of data of file function (normal and illegal)

The definition of trusted level shows that it is divided into trusted danger

Experiment on software applications with and without trusted behavior statements
Through the design of functional behavior, normal login, registration, exit, upload, download and other behavior sets

Untrustworthy behavior: illegal login, illegal registration, illegal exit, upload dangerous files, illegal download behavior

Credibility calculation and analysis

Tool credibility

In order to verify the credibility of the tool, the concept of credibility is proposed. If a is a credible behavior, then t (a, b) is the credibility of B relative to a. In fact, it is to compare the indicator data of two behaviors a and B with the sample data defined in the behavior declaration file. According to the credibility of a single test data point to the sample data set, we turn the problem into the average of the sum of the credibility of each data point in B to the data set a, which is expressed as:

To achieve the effect of trusted behavior screening, through the trusted / untrusted operation of the application, the results are as follows:

$$T(\mathrm{A},\mathrm{B}) = \sum_{n=1}^{N} \frac{t(\mathrm{A},\mathrm{B}_n)}{n}$$

Calculate and analyze the reliability of the experiment

Table 5-1 configuration behavior statement trusted experiment group

Action	Test1	Test2	Test3	Test4	Test5
Sign in	0.98	0.96	0.97	0.92	0.96
register	0.94	0.98	0.94	0.89	0.99
Sign out	0.98	0.91	0.89	0.98	0.97
upload	0.93	0.92	0.90	0.93	0.93
download	0.97	0.99	0.91	0.92	0.95

Table 5-2unconfigured behavior statement untrusted experimental group

Action	Test1	Test2	Test3	Test4	Test5
Sign in	0.18	0.16	0.17	0.12	0.06
register	0.04	0.08	0.14	0.09	0.29
Sign out	0.28	0.21	0.09	0.08	0.07
upload	0.13	0.02	0.10	0.03	0.03
download	0.07	0.19	0.11	0.02	0.05

Analysis of experimental results

Through trusted computing, it can be seen clearly that the behavior statement file can detect illegal software behavior. When the file system is operated normally, the credibility is about 0.9 The software can identify dangerous / suspicious operations, and the reliability is less than 0.2. The application can deny access.

6. CONCLUSION

In this paper, based on the early stage of software requirement analysis, the developed software functions are divided into use case diagrams and activity diagrams, and the activity diagrams obtained from the modeling are extracted. Through the formal modeling of software functional requirements, the UML activity diagrams extract the software functional lines as a set, conduct behavioral semantic Analysis on the functional layer, form the functional semantic tree, and generate the functional semantics Set, reduce functional error logic, conduct behavior trust detection constraints on functional semantic set, generate XML behavior statement through trusted statement generation tool according to the pre design behavior statement template, and use trusted computing method to verify the functional data set, which meets the expected results.

In this experiment, we use the experimental data from functional semantic set to divide the attribute and behavior level to generate set template data. The behavior statements generated by trusted tools are verified to be credible by trusted computing. Simulation experimental data and experimental simulation results show that the software configures trusted behavior statements by using untrusted function behaviors and trusted function behaviors as test cases, which can effectively deny access to untrusted behaviors. The behavior statements generated by trusted computing methods meet the explicit indicators and behavioral reliability standards.

REFERENCES

[1] Tian L Q, Lin C, Ni Y. Evaluation of User behavior trust in cloud computing[C]. Computer Application and system Modeling. 2010(7):567-572.

[2] Song H, Kim B W, Mukherjee B. Multi-thread polling: A dynamic bandwidth distribution scheme in long-reach PON[J]. Selected Areas in Communications, IEEE Journal on, 2009, 27(2): 134-142.

[3] Wei H, Chen X Y, Wang C. User Behavior analysis based on network data stream scenario. IEEE 14th International Conference on Communication Conference. 2012:1017-1021.

[4] Gan T, Lin F H, Chen C J. User Behavior Analysis in Website Identification Registration[C]. China Communications.2013(3):76-81

[5] Su D, Li J, Wang ZY. A method of dynamic trusted researching of software behavior and its trusted elements.Network Security Technology & Application, 2013, (4):14–17.

[6] TIAN J, HAN J. Trustiness Evaluation Model Based on Software Behavior[J]. Energy Procedia, 2011, 13：7991-8002.

[9] Clercq R D, Keulenaer R D, Coppens B, et al. SOFIA: Software and control flow integrity architecture[C]. Design, Automation& Test in Europe Conference & Exhibition. 2016:1172-1177.

[10] Gao D, Reiter M K, Song D. Behavioral distance for intrusion detection[C]//Recent Advances in Intrusion Detection. Springer Berlin Heidelberg, 2006: 63-81.

[11] Lin C, Tian L Q, Wang Y Z. Trusted network user behavior in the credible research [J].Research and development of the computer.2008,45(2):2033-2043.

[12] Paul C.Jorgensen. Software Testing [M]. CRC Press.2002.6.

[13] TIAN J, WANG Y. Dynamic credibility detection model base on scene mining for software behavior[J]. Energy Procedia, 2011, 13：577-584.

BUILDING A BI-OBJECTIVE QUADRATIC PROGRAMMING MODEL FOR THE SUPPORT VECTOR MACHINE

Mohammed Zakaria Moustafa[1] , Mohammed Rizk Mohammed[2] , Hatem Awed Khater[3] , Hager Ali Yahia[4]

[1]Department of Electrical Engineering (Power and Machines Section) Alexandria University, Alexandria, Egypt
[2]Department of Communication and Electronics Engineering, Alexandria University, Alexandria, Egypt
[3]Department of Computer science, HORAS University, Damietta, Egypt
[4]Department of Communication and Electronics Engineering, Alexandria University, Alexandria, Egypt

ABSTRACT

A support vector machine (SVM) learns the decision surface from two different classes of the input points, in many applications there are misclassifications in some of the input points. In this paper a biobjective quadratic programming model is utilized and different feature quality measures are optimized simultaneously using the weighting method for solving our bi-objective quadratic programming problem. An important contribution will be added for the proposed bi-objective quadratic programming model by getting different efficient support vectors due to changing the weighting values. The experimental results, give evidence of the effectiveness of the weighting parameters on reducing the misclassification between two classes of the input points.

KEYWORDS

Support vector machine (SVMs), Classification, Multi-objective problems, weighting method, Quadratic programming.

1. INTRODUCTION

Support Vector Machines (SVMs) are classification techniques developed by Vapnik at the end of ’60s [1]. The theory of support vector machines (SVMs) is a new classification technique and has drawn much attention on this topic in recent years [6]. So, the SVMs have been deeply improved to be applied in many different applications.

In many applications, SVM has been shown to provide higher performance than traditional learning machines [6]. SVMs are known as maximum margin classifiers, since they find the optimal hyperplane between two classes as shown in Fig.1, defined by a number of support vectors [4].

Natarajan Meghanathan et al. (Eds): SIPP, BIGML, DaKM, SOEN, AISC - 2020
pp. 91-98, 2020. © CS & IT-CSCP 2020 DOI: 10.5121/csit.2020.100208

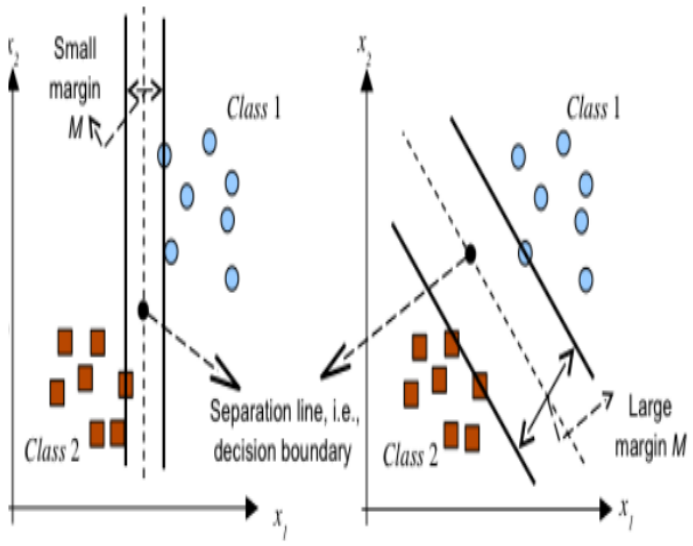


Figure 1. maximization of the margin between two classes

The well-known generalization feature of the technique is mainly due to the introduction of a penalty factor, named C that allows to prevent the effects of outliers by permitting a certain amount of misclassification errors.

In this paper, the idea is to apply the multi-objective programming technique for developing the set of all efficient solutions for the classification problem with minimum errors. The weighting method is used to solve the proposed multi-objective programming model. The remainder of this paper is organized as follows. Section 2 describes a brief review for the SVM. Section 3 describes the proposed multiobjective model for the Support Vector Machine. NEXT, section 4 presents three numerical examples. Section 5 provides our general conclusions.

2. SUPPORT VECTOR MACHINES

SVM is an efficient classifier to classify two different sets of observations into their relevant class as shown in figure 2 where there are more than straight line separates between the two sets. SVM mechanism is based upon finding the best hyperplane that separates the data of two different classes of the category. The best hyperplane is the one that maximizes the margin, i.e., the distance from the nearest training points [2]. Support vector machine has been utilized in many applications such as biometrics, chemoinformatics, and agriculture. SVM has penalty parameters, and kernel parameters that have a great influence on the performance of SVM [3]. We review the basis of the theory of SVM in classification problems [7].

Let a set S of labeled training points

$$S=(y_1, x_1),(y_2, x_2),\ldots(y_l, x_l) \quad (1)$$

Where$x_i \in \mathcal{R}^N$ belongs to either of two classes and is given a label$y_i = \{-1,1\}$ for $i = 1, \ldots, l$.

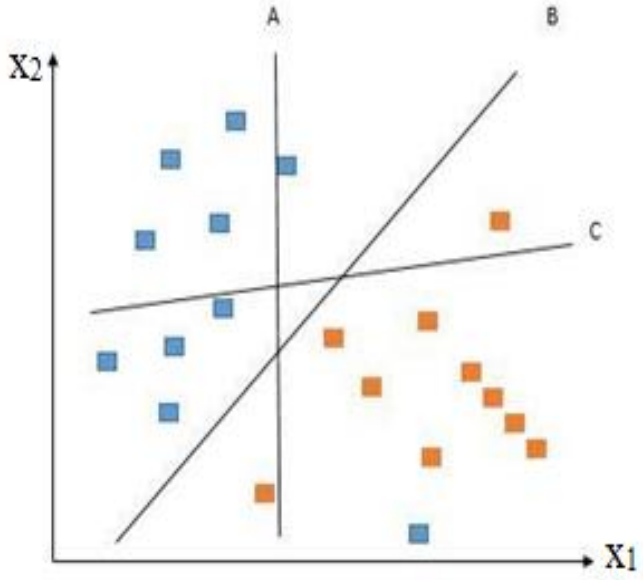


Figure 2. Data classification using support vector machine

In some cases, to get the suitable hyperplane in an input space, mapping the input space into a higher dimension feature space and searching the optimal hyperplane in this feature space.
Let z = $\varphi(x)$ denotes the corresponding feature space vector with mapping φ from $\mathcal{R}^N$ to a feature space $\mathbb{z}$. We wish to find the hyperplane

$$w.z + b = 0 \tag{2}$$

defined by the pair (w,b) according to the function

$$f(x_i) = sign(w.z_i + b) = \begin{cases} 1, & if\ \ y_i = 1 \\ -1, & if\ \ y_i = -1 \end{cases} \tag{3}$$

where $w \in \mathbb{z}$ and $b \in \mathcal{R}$. For more precisely, the equation will be

$$\begin{cases} (w.z_i + b) \geq 1, & if\ \ y_i = 1 \\ (w.z_i + b) \leq -1, & if\ \ y_i = -1, \end{cases} i = 1, \dots, l \tag{4}$$

For the linearly separable set S, a unique optimal hyperplane is determined for which the margin between the projections of the training points of two different classes is maximized.
For the data that are not linearly separable as shown in Fig. 3, the previous analysis can be generalized by introducing some non-negative variables $\xi_i \geq 0$ then,

$$y_i(w.z_i + b) \geq 1 - \xi_i, \qquad i = 1, \dots, l. \tag{5}$$

The term $\sum_{i=1}^{l} \xi_i$can be thought of as some measure of the amount of misclassifications.The optimal hyperplane problem is then regarded as the solution to the problem

$$minimize \frac{1}{2} w.w + C \sum_{i=1}^{l} \xi_i$$

$$subject\ \ to\ \ y_i(w.z_i + b) \geq 1 - \xi_i\ \ , i = 1, \dots, l \tag{6}$$

$$\xi_i \geq 0, \quad i = 1, \ldots, l$$

Where C is a constant. The parameter C can be regarded as a regularization parameter [5]. SVM algorithms use a set of mathematical functions that are defined as the kernel.

The function of kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions (i.e. linear, nonlinear, polynomial, radial basis function (RBF) and sigmoid).

Basically, the training part consists in finding the best separating plane (with maximal margin) based on specific vector called support vector. If the decision is not feasible in the initial description space, the space dimension is increased using the kernel functions and a hyperplane that will be the decision separator is determined.

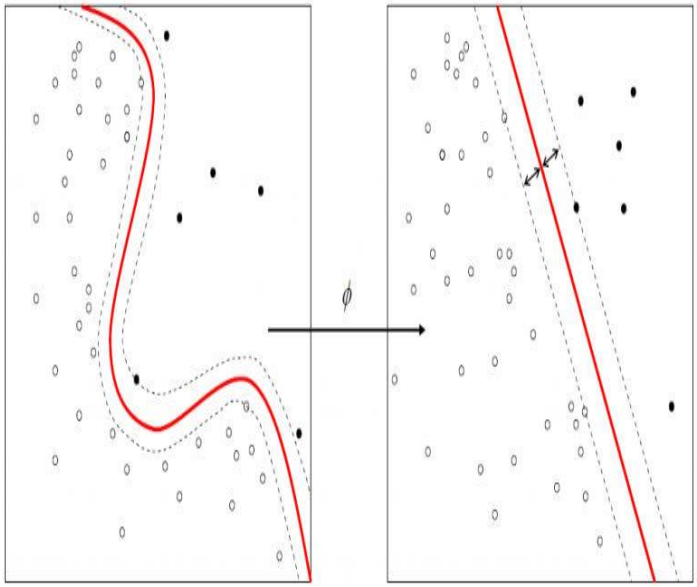


Figure 3. linearly separable and nonlinearly separable

3. FORMULATION OF THE BI-OBJECTIVE QUADRATIC PROGRAMMING MODEL OF SVM

In this section, a detailed description about the idea and formulation of the bi-objective programming model for the SVM are introduced. SVM is a powerful tool for solving classification problems, but due to the nonlinearity separable in some of the input data, there is an error in measuring the amount of misclassification. This leads us to add another objective function for the previous model in section 2 to be in the following form

$$\text{Min} \parallel w \parallel^2,$$

$$\text{Min} \sum_{i=1}^{l} \xi_i$$

$$\text{Subject to} \qquad (7)$$

$$y_i(w.x_i + b) \geq 1 + \xi_i \quad , i = 1,2, \ldots, l$$

$$\xi_i \geq 0 \qquad , i = 1,2, \ldots, l$$

This problem is a bi-objective quadratic programming problem. The first objective is to maximize the gap between the two hyperplanes which used to classify the input points. The second objective is to minimize the errors in measuring the amount of misclassification in case of nonlinearity separable input points. The problem 7 can be solved by the weighting method to get the set of all efficient solutions for the classification problem. The right choice of weightage for each of these objectives is critical to the quality of the classifier learned, especially in case of the class imbalanced data sets. Therefore, costly parameter tuning has to be undertaken to find a set of suitable relative weights [10].

3.1 The Weighting Method

In this method each objective $f_i(X), i = 1,2,\dots,k,$ is multiplied by a scalar weigh $w_i \geq 0\ and \sum_{i=1}^{k} w_i = 1$. Then, the k weighted objectives are summed to form a weighted-sums objective function [8].

$$Assume\ \ W \text{ as } \begin{Bmatrix} w \in R^k : w_i \geq 0, \\ i = 1,2,\dots,k \\ and \sum_{i=1}^{k} w_i = 1 \end{Bmatrix} \tag{8}$$

be the set of nonnegative weights. Then the weighting problem is defined as:

$$P(W): Min \sum_{i=1}^{k} w_i f_i$$

$$\text{Subject to } M = \begin{Bmatrix} X \in R^n : g_r(X) \leq 0, \\ r = 1,2,\dots,m \end{Bmatrix}. \tag{9}$$

Then, in this paper the weighting method takes the form

$$\text{Inf z} = w_1 \parallel w \parallel^2 + w_2 \sum_{i=1}^{l} \xi_i$$

Subject to

$$y_i(w.x_i + b) \geq 1 + \xi_i \quad , i = 1,2,\dots,l$$

$$\xi_i \geq 0 \quad , i = 1,2,\dots,l \tag{10}$$

$$w_1 > 0, w_2 \geq 0$$

$$w_1 + w_2 = 1$$

Here we use "Inf "instead of "Min" since the set of constraints is unbounded, where $w_1 \neq 0$.Also, we avoid the redundant solutions by adding the constraint $w_1 + w_2 = 1$.

4. NUMERICAL EXAMPLES

Using python programming language, the previous problem is solved and the effect of different values of the weighting parameters is described. The data set that is used in these examples consist of 51 points and each point has two features, table 1 shows part of this data.

Table 1. Description of part of datasets used in our study.

X1	X2	Y
1.9643	4.5957	1
2.2753	3.8589	1
2.9781	4.5651	1
2.932	3.5519	1
3.5772	2.856	1
0.9044	3.0198	0
0.76615	2.5899	0
0.086405	4.1045	0

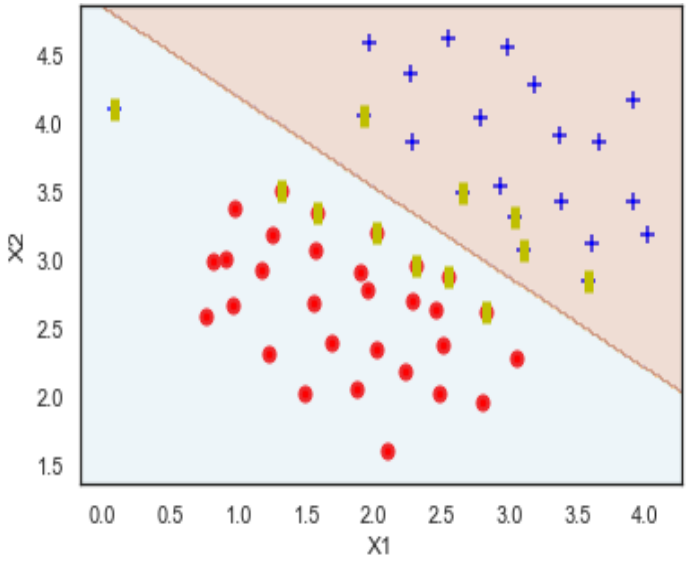


Figure 4. $w_2 = \frac{1}{2}, w_1 = \frac{1}{2}$, number of support vectors = 12

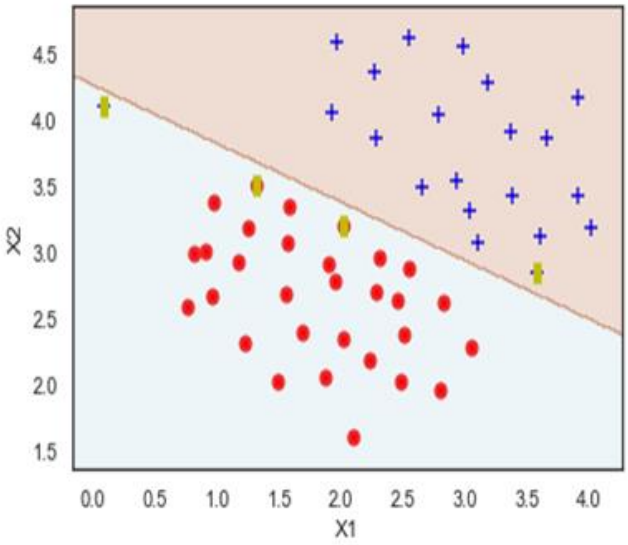


Figure 5. $w_2 = \frac{20}{21}, w_1 = \frac{1}{21}$, number of support vectors = 4

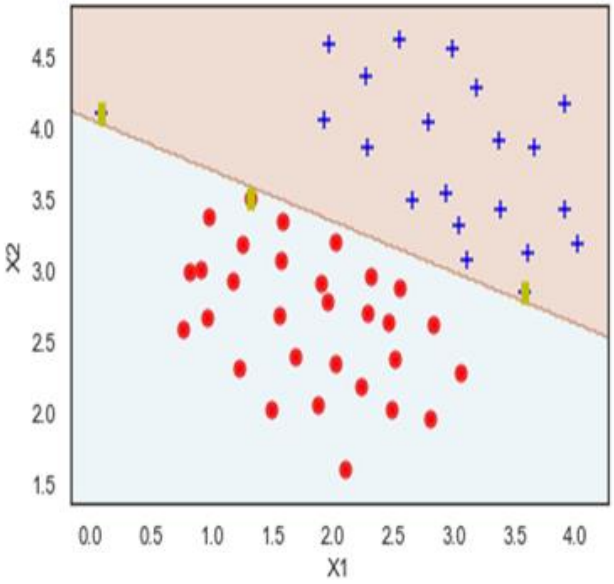


Figure 6. $w_2 = \frac{99}{100}, w_1 = \frac{1}{100}$, number of support vectors=3

So, the previous results, by using different values of weighting parameters, show how these parameters effect on the performance of SVM. For the first values of w_1& w_2, there is one point of the blue set can't be classified to its set, the second values make this point closes to its set and the third values of w_1&w_2, this point can be joined to its set. So, when the weighting parameter w_2 is increased, the misclassification and the number of support vectors will be reduced as shown in Fig. 5 and Fig. 6. There are good reasons to prefer SVM^s with few support vectors (SV^s). In the hard-margin case, the number of SV^s is an upper bound on the expected number of errors made by the leave-one-out procedure [9].

So, we can control the performance of SVM according to the requirements by adjusting the values of the weighting parameters.

5. CONCLUSIONS

This paper introduced the multi-objective programming technique for developing the set of all efficient solutions for the classification problem with minimum errors and how to solve the proposed multi-objective programming model by using the weighting method. The experimental evaluation was carried out using 51 datasets, each one has two features. The experimental results show the effect of the weighting parameters on the misclassification between two sets.

The future work can include a construction of a utility function to select the best compromised hyperplane from the generated set of the efficient solutions and building a fuzzy bi-objective quadratic programming model for the support vector machine.

REFERENCES

[1] Cortes, Corinna; Vapnik, Vladimir N (1995) "Support vector networks" (PDF). Machine learning. 20 (3):273297. CiteSeerX 10.1.1.15.9362. DOI:10.1007/BF00994018.

[2] Asmaa Hamad1,3(B), Essam H. Houssein1,3, Aboul Ella Hassanien2,3, and Aly A. Fahmy2 :Hybrid Grasshopper Optimization Algorithm and Support Vector Machines for Automatic Seizure Detection in EEG Signals. Faculty of Computers and Information, Minia University, Minya, Egypt. January 2018. DOI: 10.1007/978-3-319-74690-6_9.

[3] Alaa Tharwat1;_, Thomas Gabel1, Aboul Ella Hassanien2;_ Parameter Optimization of Support Vector Machine using Dragon_y Algorithm. Faculty of Computer Science and Engineering, Frankfurt University of Applied Sciences, Frankfurt am Main, Germany ,Faculty of Computers and Information, Cairo University, Egypt. January 2018 DOI: 10.1007/978-3-319-64861-3_29.

[4] Gray, D., Bowes, D., Davey, N., Sun, Y., Christianson, B.: Using the Support Vector Machine as a Classification Method for Software Defect Prediction with Static Code Metrics. In: Palmer Brown, D., Draganova, C., Pimenidis, E., Mouratidis, H. (eds.) EANN 2009. Communications in Computer and Information Science, vol. 43, pp. 223–234. Springer,Heidelberg (2009).

[5] Chun-Fu Lin and Sheng-De Wang: Fuzzy Support Vector Machines. Article in IEEE Transaction on neural networks March 2002. DOI:10.1109/72.991432.

[6] C. Burges, ”A tutorial on support vector machines for pattern recognition,” Data Mining and Knowledge Discovery, vol.2, no.2,1998.

[7] C. Cortes and V. N. Vapnik, “Support vector networks,” Machine Learning, vol.20,pp.273-297,1995.

[8] Chankong V. and Haimes Y.Y., Multi-objective Decision-Making: Theory and Methodology (North Holland Series in System Science and Engineering, 1983).

[9] Yaochu Jin (Ed.), Multi-objective Machine Learning Studies in Computational Intelligence, Vol. 16, pp. 199-220, Springer-Verlag, 2006.

[10] Shounak Datta and Swagatam Das, Multiobjective Support Vector Machines: Handling Class Imbalance With Pareto Optimality, IEEE Transactions on Neural Networks and Learning Systems, 2019.DOI:10.1109/TNNLS.2018.2869298.

A BRIEF SURVEY OF DATA PRICING FOR MACHINE LEARNING

Zuoqi Tang[1], Zheqi Lv[2], Chao Wu[3,4 *]

[1]Department of Computer Science and Technology, Zhejiang University, China
[2]Department of Marine Informatics, Zhejiang University, China
[3]Department of Public Affairs, Zhejiang University, China
[4]Center of Social Welfare and Governance, Zhejiang University, China

ABSTRACT

Big data and machine learning are poised to revolutionize the field of artificial intelligence and represent a step towards building an intelligent society. Big data is considered to be the key to unlocking the next great waves of growth in productivity, the value of data is realized through machine learning.

In this survey, we begin with an introduction to the general field of data pricing and distributed machine learning then progress to the main streams of data pricing and mechanism design methods. Our survey will cover several current areas of research within the field of data pricing, including the incentive mechanism design for federated learning, reinforcement learning, auction, crowdsourcing, and blockchain, especially, focus on reward function for machine learning and payment scheme. In parallel, we highlight the pricing scheme in data transactions, focusing on data evaluation via distributed machine learning. To conclude, we discuss some research challenges and future directions of data pricing for machine learning.

KEYWORDS

Data pricing, Big data, Machine learning, Data transaction

1. INTRODUCTION

Nowadays, we live in the era of mobile Internet, big data and artificial intelligence, these technologies are changing our lives.

In the era of the mobile Internet, especially the coming 5G and the Internet of Things, which is more open and more connected. In this scenario, the data which distributed the different edging nodes is exploding exponentially, these data belong to the different organizations and individuals. Data collection and data value become more important for economic and social activities.

According to the report of the Forbes, the worldwide Big Data market revenues for software and services are projected to increase from $42B (Billion) in 2018 to $103B in 2027, attaining a Compound Annual Growth Rate (CAGR) of 10.48%. As a part of this forecast, Wikibon estimates the Worldwide Big Data market is growing at an 11.4% CAGR between 2017 and 2027, growing from $35B to $103B, as illustrated in figure 1. At the same time, artificial intelligence (AI) is transforming economies and societies, changing the way we communicated and work. As the machine learning models are growing larger and more complex, it requires a variety of data, which is heterogeneous. How to effectively process data while protecting privacy is a huge challenge in the field of machine learning for academic and industrial circles.

Natarajan Meghanathan et al. (Eds): SIPP, BIGML, DaKM, SOEN, AISC - 2020
pp. 99-110, 2020. © CS & IT-CSCP 2020 DOI: 10.5121/csit.2020.100209

So, in the age of data-driven technologies, data pricing or incentive is an important topic between data consumers and data owners for machine learning. One of the central question in machine learning deals with pricing or reward the behavior of a data provider. In addition to the perspective of the AI, data pricing would be key to support the setup as well as to sustain a good ecosystem for data markets.

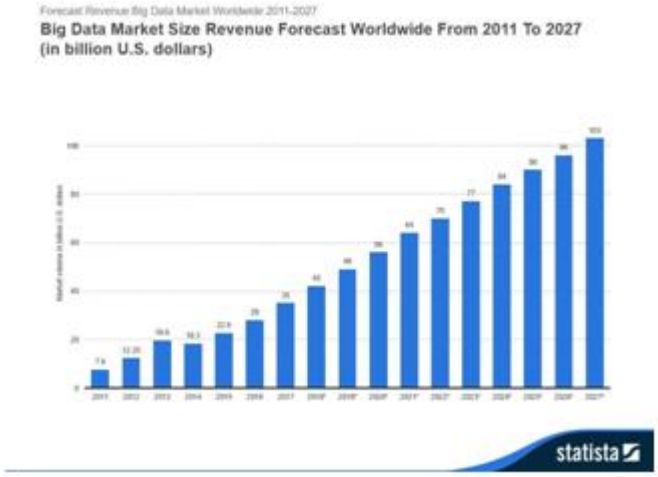


Figure 1. Revenue forecast (Source: Wikibon and reported by Statista)

The remainder of this survey is organized as follows: In section 2, we briefly introduce the basic concept related to the data pricing for machine learning and the research progress of data pricing. In section 3, we review the mechanism design in artificial intelligence. In section 4, we review the pricing mechanism in the data transaction. In section 5, we discuss the challenges and future directions. Finally, we conclude the paper in section 6.

2. DATA PRICING

In this section, we review some research progress of data pricing.

2.1. Data Market

The amount of collected data in our world has been exploding due to a number of new applications, especially mobile applications and the Internet of Thing-based smart systems. With the exponential growth of data, how to efficiently utilities the data becomes a critical issue. This calls for the development of a big data platform that enables efficient data utilization between data providers and data consumers (Machine learning) [15]. In the data market, three agents are involved, namely, the seller who provides the datasets, the buyer who is interested in buying ML models instance, and the market who interacts between the seller and buyer. In data markets, how to verify whether the service provider has truthfully collected and processed data becomes a pressing problem for the data consumer. Niu et al. [20] proposed the first secure mechanism TPDM (integrating Truthfulness and Privacy preservation in Data Markets) for personal data markets, achieving both truthfulness and privacy preservation.

The Internet of Things (IoT) has emerged as a new paradigm for the future Internet. In IoT, devices are connected to the Internet and thus are a huge data source for numerous applications. In [22], Niyato et al. focused on addressing data management in IoT through using a smart data pricing approach and proposed a new pricing scheme for IoT service providers to determine the sensing data buying price and IoT service subscription fee offered to sensor owners and service user.

In [21], the authors introduced a big data market model, which is composed of a data source, a service provider, and consumer, and proposed utility functions of data when the data is used in big data analytics. Based on the data utility functions, the authors developed an optimal pricing scheme that allows the service provider to determine the amount of data to be acquired to provide services to users. Additionally, the authors showed the Stackelberg game can be model the strategy of the data source to achieve the maximum profit.

2.2. Categories of the Data Pricing Models

There are two pricing models according to the different principles. The traditional models are economic-based pricing, which establishes the price strategies based on classic economic theory. The second model is game theory-based pricing, which is the dynamic price in competitive scenarios [17].

The game theory finds nowadays a broad range of applications in engineering and machine learning. The Game theory being a mathematical tool to analyze strategic interactions between rational decision-makers, in this survey, we review the usage of Game Theory in different machine learning settings involving usage of large amounts of data[23]. The goal is to provide an overview of the use of game theory in different applications that rely extensively on big data.
In the traditional data markets, Muschalle et al. [18] listed the six main categories of pricing models.

- **Free** data can be obtained from public authorities.
- **Usage-Based prices** correspond to the human rationality that every single unit of a commodity raises the total amount of money to pay for.
- **Package pricing** refers to a pricing model that offers a customer a certain amount of data for a fixed fee.
- **Flat fee tariff** is one of the simplest pricing models with minimal transaction costs. It is based on time as the only parameter.
- **Two-Part Tariff** is a combination of package pricing and flat pricing strategies.
- **Freemium** is another approach of pricing data and algorithms on marketplaces. The idea is to let users join and use basic services for free and charge them for premium services that provide additional value to customers.

On the other hand, Raskar et al. [24] proposed a thorough data pricing strategy that needs to adhere to the following guiding principles.

- **Liquidity**: models freshness of data in terms of value vs diminished/increase value over time.
- **Traceability**: can be only ’sold’ once, or sold non-exclusively.
- **Consent**: maintains the privacy of the owner, tracks consent over time, and reduces.
- **Neutrality:** accessible to all buyers to prevent unfair trading practices.
- **Resource:** allows for calling back, providers right to be forgotten, allows for some course correction, broadly remains self-sustaining.

2.3. Data Pricing for Private Data

Personal data has value to both its owner and to institutions. Li et al. [13] introduced a framework for selling private data. Buyers can purchase any liner query, with any amount of perturbation, and need to pay accordingly. Data owners, in turn, are compensated according to the privacy loss they incur for each query. Moreover, with the help of comparative analysis of existing data pricing models and strategies, Shen et al. [29] proposed a pricing model for Big Data based on tuple granularity, which includes information entropy, weight value, data reference index, cost, and credit rating.

3. MECHANISM DESIGN IN AI

Mechanism design is a field in economics and game theory that takes an engineering approach to design economic mechanisms or incentives, toward desired objectives, in strategic settings, where players act rationally. Because it starts at the end of the game, then goes backward, it is also called reverse game theory. Mechanism design studies solution concepts for a class of private-information games.

In this section, we review state-of-the-art research works on the application of the Mechanism design.

3.1. Incentive Mechanism in Federated Learning

Today's artificial intelligence still faces two major challenges. One is that, in most industries, data exists in the form of isolated islands. The other is how to price data value in the different organizations. Most existing data pricing methods are very ineffective and unavailable for training Machine learning models when we cannot directly access the training dataset. The Federated Learning (FL) [11] first proposed by Google in 2016, which is a possible solution to these challenges. Next, we will make a brief introduction to distributed machine learning and federated learning.

Distributed machine learning commonly refers to multi-node machine learning algorithms and architecture, which are designed to improve performance, increase accuracy, and adapt to bigger train dataset or models. The architecture of the distributed machine learning is shown in Fig 2.

- Data/Model aggregation
- Single machine/model optimization
- Submodel/Local Data
- Data/Model partition

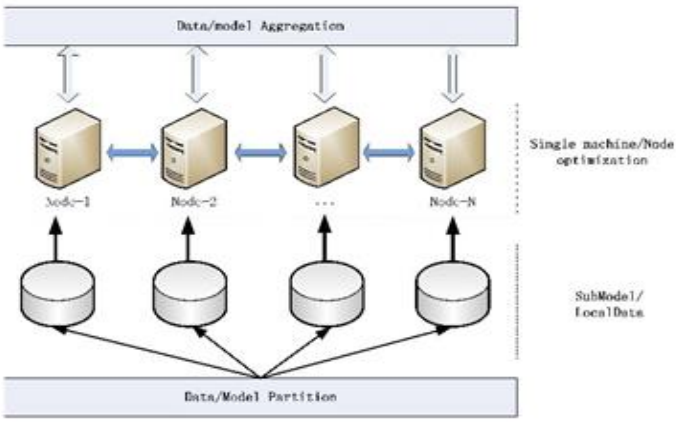


Figure 2. Distributed machine learning architecture

Federated Learning is a distributed machine learning approach that enables model training on a large corpus of decentralized data [2]. Federated Machine learning (FML) [34] creates an ecosystem for multiple parties to collaborate on the building models while protecting data privacy for the participants. Instead of transferring data directly into a centralized data warehouse for building machine learning models, Federated Learning allows each party to own the data in its place and still enables all parties to build a machine learning model together. This method stands in contrast to the traditional centralized machine learning approach where all data samples are uploaded to one data center.

To fully commercialize federated learning among different organizations, a fair platform, and incentive mechanism needs to be developed. After the model simulations built, the performance of the model will be manifested in the actual application. This performance can be recorded in a permanent data recording mechanism. Organizations that provide more data will be better off, and the model's effectiveness depends on the data provider's contribution to the system. The effectiveness of these models is distributed to parties based on the federated mechanism and continue to motivate more organizations to join the data federation [37].

Currently, federated learning is an emerging machine learning technique, most of the existing studies mainly focus on optimizing federated learning algorithms to improve model training performance, data privacy, and security. However, incentive mechanisms to motivate mobile edging smart devices to join model training have been largely overlooked. Edging devices suffer from considerable overhead in terms of computation and communication during the federated mode learning process. Without a well-designed incentive, self- interested mobile devices will be unwilling to join federated learning tasks. In [10], Kang et al. adopted the contract theory to design an effective incentive mechanism for simulating mobile devices with high-quality data participate in federated learning.

3.2. Incentive Mechanism in Reinforcement Learning

The design of reward functions in reinforcement learning (RL) is a human skill that comes with experience. Unfortunately, there is not any methodology in the literature that could guide a human to design the reward function. In [4], Clayton et al. used systematic instructional Design, an approach in human education, to engineer a machine education methodology to design reward functions for reinforcement learning. The methodology can guide the design of hierarchy learning incrementally through a multi-part reward function. The hierarchy acts as a decision fusion function that combines the individual behaviors and skill learn by each instruction to create a smart shepherd to control the swarm.

3.3. Mechanism Design in Auction

Auctions are protocols to allocate goods to buyers who have preferences over them and collect payments in return. Economists have invested significant effort in designing auction rules that result in allocations of the goods that are desirable for the group as a whole. Mechanism design is a field in economics that deals with setting incentives and interaction rules among self- interested agent samples to achieve desired objects for the group as a whole. In [33], Teerapittayanon et al. proposed a deep learning-based approach to automatically design auctions in a wide variety of domains, shifting the design work from human to machine. On the other hand, considering the traditional markets, it is extremely difficult for e-commerce companies to adjust prices when they receive more information from consumers. Shen et al. [28] proposed a reinforcement mechanism design [32] to tackle the dynamic pricing problem in sponsored Search auctions.

3.4. Mechanism Design in Crowdsourcing

An interesting recent scenario of inaccurate supervision occurs with crowdsourcing, a popular paradigm to outsource work to the individual. For machine learning, crowdsourcing is commonly used as a cost-saving way to collect labels for training data. The ability to quickly collect large-scale and high labeled datasets is crucial for Machine Learning. In online crowdsourcing, designing optimal pricing policies and determining the right monetary incentives is central to maximizing the requester's utility and workers' profit. To address these questions, regret minimization mechanisms are presented, which combine procurement auctions and multi-armed

bandits [30]. Considering existing mechanisms are often developed as a one- shot static solution, assuming a certain level of knowledge about worker models. In [6], a reinforcement incentive learning (RIL) method was proposed, to uncover how workers respond to different payments. RIL dynamically determines the payment without accessing any ground truth labels, and RIL can incentive rational workers to provide high-quality labeled. Designing an effective Crowdsourcing protocol is important. In [26], a 'double or nothing' incentive- compatible mechanism is proposed to ensure workers behave honesty based on their self- confidence; this protocol is provable to avoid spammers from the crowd, under the assumption that every worker wants to maximize their expected payment. In [40, 39, 41, 42, 44, 43], Zheng et al, the authors leverage the tools of game theory and mechanism design to analyze the interaction of rational and selfish mobile users, then design efficient incentive mechanisms for four classical and representative applications in mobile Internet: dynamic spectrum redistribution, mobile crowdsensing, data marketplace, and cloud bandwidth management, to stimulate selfish mobile users to cooperate, achieving a win-win situation.

With the rapid growth of smart IoT devices, a mobile crowd-sensing is becoming an important paradigm to acquire information from the physical environment. However, it is challenging to estimate the data quality without the availability of ground truth data. Liu et al. [16] proposed a context-aware data quality estimation in an online manner. Zheng et al. [43] presented the first architecture of the mobile crowd-sensed data market, and conduct an in-depth study of the design problem of online data pricing. A novel online query-based crowd-sensed data pricing mechanism was proposed to determine the trading price of crowd-sensed data.

To improve the efficiency and utility of mobile Crowdsourcing system, Wang et al. [35] proposed an incentive mechanism that selects the worker candidates statically, and then dynamically selects winners after bidding. The proposed incentive mechanism includes two algorithms which are an improved two-stage auction algorithm and a truthful online reputation updating algorithm.

3.5. Reward Mechanism in Blockchain

In traditional blockchain [19], proof-of-work (PoW) incentivizes people to participate in the consensus protocol for a reward. Teerapittayanon et al. [33] introduced DaiMoN, a decentralized artificial intelligence model Network, which incentivizes peer collaboration in improving the accuracy of machine learning models. It is an autonomous Network where peers may submit models with improved accuracy and other peers may verify the accuracy improvement. DaiMoN rewards these contributing peers with cryptographic tokens by using a novel learnable Distance Embedding for Labels (DFL) function.

As the core issue of blockchain, mining requires solving a proof-of-work puzzle, which is resource expensive to implement in mobile devices due to the high computing power needed. Thus, the development of blockchain in the mobile application is restricted. To support offloading from mobile blockchain mining, an optimal pricing-based edge computing resource management approach was proposed. In [36], a two-stage Stackelberg game was adopted to jointly maximize the profit of Edge computing service Provider (ESP) and the individual utilities of different miners.

4. PRICING MECHANISM IN DATA TRANSACTIONS

This section reviews some research progress of data pricing, focusing on two types of data pricing: data-based pricing and model-based pricing.

4.1. Data-based Pricing

Most online markets are characterized by competitive settings and limited demand information. Due to the complexity of such markets, efficient pricing strategies are hard to derive. Schlosser et al. [25] analyzed stochastic dynamic pricing models in competitive markets with multiple offer dimensions, such as price, quality, and rating. Yu et al. [38] presented a bi-level mathematical programming model for the data-pricing problem that considers both data quality and data versioning strategies and a genetic algorithm was used to solve the model.

4.2. Machine-learning-based Pricing

Currently, there are two main solutions to train the machine learning model in the literature. The first is the traditional machine learning modeling method, which collects local data and uploads to the data center and then trains the machine learning model, namely centralized machine learning. The second is the distributed machine learning method, as known as decentralized machine learning. The data is at the edge node. The submodel is trained at the edge nodes and then the submodel is assembled in the central server. Moreover, collecting data through the data market and crowdsourcing is an effective method for centralized machine learning, however, decentralized machine learning is realized through blockchain or federated learning. The methods of machine learning modeling are shown in figure 3. Regardless of the way, how to do data incentives or pricing is the core and key issue for solving machine learning.

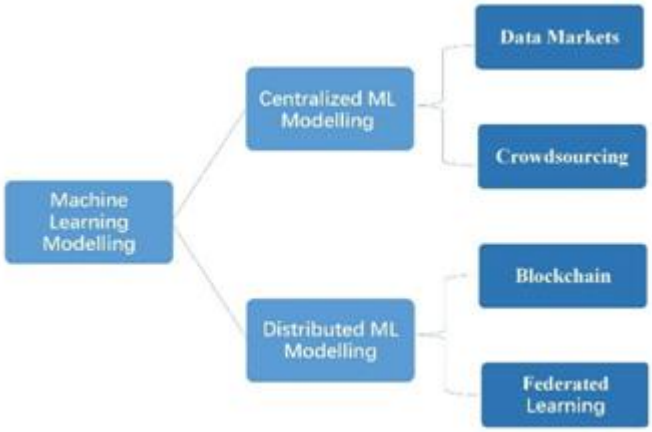


Figure 3. Methods of machine learning modeling

Data analytics using machine learning (ML) has become ubiquitous in science and engineering domain. While a lot of work focuses on reducing the training cost and storage cost of ML models, little work studies how to reduce the cost of data acquisition. Especially in the age of the Internet, data is increasingly concentrated in large firms; in the era of the Internet of things, tremendous sensor data distributed in the edging smart devices. For startups, small organizations, and institutes it is increasingly difficult to compete as the lack of availability of data can stymie any and all efforts to build a better machine-learning algorithm. Algorithm capability indeed increases with the availability and quality of the data. One way to tackle this is the marketplace approach. By creating conditions such that data, the raw material for Artificial Intelligence, can be bought and sold with security, privacy, and consent safeguard. Koutris et al. [12] proposed a framework for pricing data on the Internet that, given the price of a few views, allows the prices of any query to be derived automatically, and proposed a generalized chain queries and obtain the price by computing the time complexity of the query. Chen et al. [3] proposed a model-based pricing (MBP) framework, which instead of pricing the data, directly prices model instance. The price should depend on the accuracy of the model purchased, and not the underlying datasets.

The Shapley value [27], which is originated from coalitional game theory with proven theoretical properties, provides an effective approach to distribute contribution among features in a fair way by assigning to each feature a number which denotes its influence.

In data Markets, a fundamental challenge is how to quantify the value of data in algorithm predictions and decisions. Ghorbani et al. [5] developed a principle framework to address data valuation in the context of Supervised machine learning. Given a learning algorithm trained on n data points to produce a predictor, data Shapley was proposed to quantify the value of each training datum to the predictor performance. Although, cooperative game theory suggests as a unique way to distribute payment to data contributors such that some important theoretical properties are satisfied-data Shapley value uniquely satisfies several natural properties of equitable data valuation. Unfortunately, computing Shapley Value exactly is prohibitively expensive, therefore authors developed a sampling-based approximation algorithm-Truncated Monte Carlo Shapley. Besides, "How much is my data worth?" is an increasingly common question posed by organizations and individuals. To solve this question of fairly distributing profits among multiple data contributors, Jia et al. [9] studied the problem of data valuation by utilizing the Shapley Value, a popular notion of value that originated in cooperative game theory. In [8], jia et al. categorized firstly a data valuation problem according to whether data contributors are valued in tandem with a data analyst; where each data contributor provides a single data instance or multiple ones; whether the underlying ML model is a weighted KNN or unweighted; and whether the model solves a regression or a classification task. Then introduced two game-theoretical models for distributing the gains from an ML model and would like to understand how the shares of the analyst and the data contributors differ in the two models.

5. CHALLENGES AND FUTURE DIRECTIONS

This section discusses some research challenges and future directions of data pricing, focusing on Machine-learning-based pricing.

5.1. Challenges

Currently, the biggest challenge in the data trading market is the lack of a unified pricing benchmark, especially for machine learning, so all kinds of data trading mechanisms are not available in practice. To solve the biggest problem, the pricing benchmark of the dataset is essential, such as MNIST, CIFAR-10, CIFAR-100, and ImageNet et al.

Another challenge is how to design an efficient and appropriate incentive mechanism for decentralized machine learning. Thanks to the incentive mechanism on distributed machine learning are still in its early stages and despite the apparent opportunities it offers both federated learning and other machine learning, there exist several critical challenges in data pricing. First of all, data pricing methods are hard to evaluate empirically because it is difficult to distinguish the error of the model from the error of the data method explaining the machine learning model. Besides, for this reason, the state-of-the-art data pricing methods are often qualitative, based on the contributions of the produced machine learning model. To develop better quantitative data pricing methods for the evaluation of data value, we will need to define the goal that an ideal data pricing method should achieve, in terms of different methods that might be suitable for different machine learning tasks.

5.2. Futures Research Directions

Currently, due to slightly different data pricing formulations, lack of definition of the utility function for the variety of existing Machine learning models and no common benchmark,

comprehensive data pricing is not available. Besides, the data pricing issues of the key task on distributed machine learning over time have not been completely addressed in the literature. However, in the machine learning scenario, using game theory and mechanism design tools will be an effective way to solve the data pricing problem. So, the research about data pricing for machine learning is still one of the research focuses on the field of artificial intelligence and big data. Moreover, understanding why complex Machine Learning Model makes a precise prediction can be as crucial as the big data value in many applications.

A research solution of data pricing based on federated learning as illustrated in figure 4.

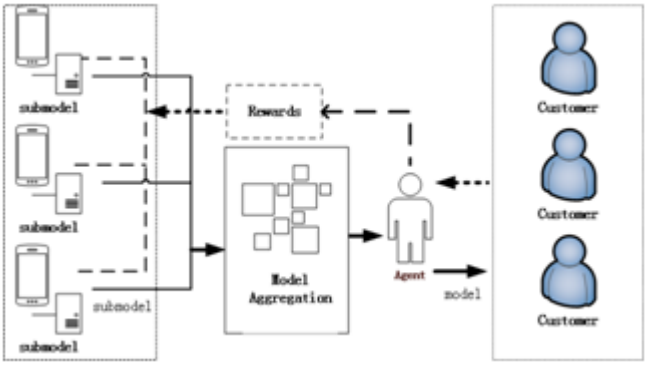


Figure 4. Overview of the data pricing based FL

For centralized machine learning settings, gradient-based attribution methods [1] can be used to explain deep neural networks (DNN) and to evaluate data value, including global attribution methods and local attribution methods. A model aggregating with attention mechanism [7] maybe be used to evaluate the client data value by computing the contribution of distributed client models to the global model during server aggregation. Considering heterogeneous federated learning, use transfer learning and knowledge distillation [14] to develop a universal data pricing framework for distributed machine learning, which enables each participant can gain rewards through owning their private data. Exploring the transferability between heterogeneous [31] datasets sheds light on their intrinsic data value, and consequently enables knowledge transfer from different datasets to the machine learning model to valuation the value of training datasets the latter.

The reliable and efficient operation of distributed machine learning relies on the cooperation of edging nodes, which are the organizations or individuals with different optimization goals. There exist conflicts between individual goals and the overall system objectives. The rational and selfish behaviors of edging nodes would lead the system to an anarchic state and degrade system performance. Therefore, we need to design efficient mechanisms to incentives the cooperation among edging nodes guaranteeing that distributed machine learning runs in an efficient and ordered way. However, designing an appropriate incentive mechanism for machine learning is a complex and challenging task. Intuitively, data is essential in many AI application scenario, and this data may be available with strategic players who need not reveal it truthful to the (AI) system designer unless there are offered proper incentives.

6. CONCLUSIONS

In this survey, we present an overview of the progress of data pricing in data transactions. In particular, we introduce the research progress of the game theory and mechanism design in distributed machine learning in detail, analyze the challenges faced, and look forward to future research directions.

Federated learning, as a type of distributed machine learning, has promising future development, but the challenges it faces are also huge, and these challenges are exactly the direction of a large number of researchers in federated learning.

REFERENCES

[1] Marco Ancona, Enea Ceolini, Cengiz Oztireli, & Markus Gross, (2018) Towards a better understanding of gradient-based attribution methods for deep neural networks. In 6th International Conference on Learning Representations (ICLR 2018).

[2] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H Brendan McMahan, et al., (2019) Towards federated learning at scale: System design. arXiv preprint arXiv:1902.01046.

[3] Lingjiao Chen, Paraschos Koutris, & Arun Kumar, (2019) Towards model-based pricing for machine learning in a data marketplace. In Proceedings of the 2019 International Conference on Management of Data, pages 1535–1552. ACM.

[4] Nicholas R Clayton & Hussein Abbass, (2019) Machine teaching in hierarchical genetic reinforcement learning: Curriculum design of reward functions for swarm shepherding. arXiv preprint arXiv:1901.00949.

[5] Amirata Ghorbani & James Zou, (2019) Data shapley: Equitable valuation of data for machine learning. In International Conference on Machine Learning, pages 2242–2251.

[6] Zehong Hu, Yitao Liang, Jie Zhang, Zhao Li, & Yang Liu, (2018) Inference aided reinforcement learning for incentive mechanism design in crowdsourcing. In Advances in Neural Information Processing Systems, pages 5507–5517.

[7] Shaoxiong Ji, Shirui Pan, Guodong Long, Xue Li, Jing Jiang, & Zi Huang, (2019) Learning private neural language modeling with attentive aggregation. In International Joint Conference on Neural Networks (IJCNN).

[8] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gurel, Bo Li, Ce Zhang, Dawn Song, & Costas Spanos, (2019) Towards efficient data valuation based on the shapley value. arXiv preprint arXiv:1902.10275.

[9] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas Spanos, & Dawn Song, (2019) Efficient task-specific data valuation for nearest neighbor algorithms. Proceedings of the VLDB Endowment, 12(11):1610–1623.

[10] Jiawen Kang, Zehui Xiong, Dusit Niyato, Han Yu, Ying-Chang Liang, & Dong In Kim, (2019) Incentive design for efficient federated learning in mobile networks: A contract theory approach. arXiv preprint arXiv:1905.07479.

[11] Jakub Konecˇny`, H Brendan McMahan, Felix X Yu, Peter Richta´rik, Ananda Theertha Suresh, & Dave Bacon, (2016) Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492.

[12] Paraschos Koutris, Prasang Upadhyaya, Magdalena Balazinska, Bill Howe, & Dan Suciu, (2015) Query-based data pricing. Journal of the ACM (JACM), 62(5):43.

[13] Chao Li, Daniel Yang Li, Gerome Miklau, & Dan Suciu, (2014) A theory of pricing private data. ACM Transactions on Database Systems (TODS), 39(4):34.

[14] Daliang Li & Junpu Wang, (2019) FedMD: Heterogenous federated learning via model distillation. arXiv preprint arXiv:1910.03581.

[15] Fan Liang, Wei Yu, Dou An, Qingyu Yang, Xinwen Fu, & Wei Zhao, (2018) A survey on big data market: Pricing, trading and protection. IEEE Access, 6:15132–15154.

[16] Shengzhong Liu, Zhenzhe Zheng, Fan Wu, Shaojie Tang, & Guihai Chen, (2017) Context-aware data quality estimation in mobile crowdsensing. In IEEE INFOCOM 2017-IEEE Conference on Computer Communications, pages 1–9. IEEE.

[17] Nguyen Cong Luong, Dinh Thai Hoang, Ping Wang, Dusit Niyato, Dong In Kim, & Zhu Han, (2016) Data collection and wireless communication in internet of things (IioTt) using economic analysis and pricing models: A survey. IEEE Communications Surveys & Tutorials, 18(4):2546–2590.

[18] Alexander Muschalle, Florian Stahl, Alexander Lo¨ser, & Gottfried Vossen, (2012) Pricing approaches for data markets. In international workshop on business intelligence for the real-time enterprise, pages 129–144. Springer.

[19] Satoshi Nakamoto et al., (2008) Bitcoin: A peer-to-peer electronic cash system.

[20] Chaoyue Niu, Zhenzhe Zheng, Fan Wu, Xiaofeng Gao, & Guihai Chen, (2017) Trading data in good faith: Integrating truthfulness and privacy preservation in data markets. In 2017 IEEE 33rd International Conference on Data Engineering (ICDE), pages 223–226. IEEE.

[21] Dusit Niyato, Mohammad Abu Alsheikh, Ping Wang, Dong In Kim, & Zhu Han, (2016) Market model and optimal pricing scheme of big data and internet of things (IioTt). In 2016 IEEE International Conference on Communications (ICC), pages 1–6. IEEE.

[22] Dusit Niyato, Dinh Thai Hoang, Nguyen Cong Luong, Ping Wang, Dong In Kim, & Zhu Han, (2016) Smart data pricing models for the internet of things: a bundling strategy approach. IEEE Network, 30(2):18–25.

[23] Praveen Paruchuri & Sujit Gujar, (2018) Fusion of game theory and big data for ai AI applications. In International Conference on Big Data Analytics, pages 55–69. Springer.

[24] Ramesh Raskar, Praneeth Vepakomma, Tristan Swedish, & Aalekh Sharan, (2019) Data markets to support AIai for all: Pricing, valuation and governance. arXiv preprint arXiv:1905.06462.

[25] Rainer Schlosser & Martin Boissier, (2018) Dynamic pricing under competition on online marketplaces: A data-driven approach. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 705–714. ACM.

[26] Nihar Bhadresh Shah & Dengyong Zhou, (2015) Double or nothing: Multiplicative incentive mechanisms for crowdsourcing. In Advances in neural information processing systems, pages 1–9.

[27] Lloyd S Shapley, (1953) A value for n-person games. Contributions to the Theory of Games, 2(28):307–317.

[28] Weiran Shen, Binghui Peng, Hanpeng Liu, Michael Zhang, Ruohan Qian, Yan Hong, Zhi Guo, Zongyao Ding, Pengjun Lu, & Pingzhong Tang, (2017). Reinforcement mechanism design, with applications to dynamic pricing in sponsored search auctions. arXiv preprint arXiv:1711.10279.

[29] Yuncheng Shen, Bing Guo, Yan Shen, Xuliang Duan, Xiangqian Dong, & Hong Zhang, (2016) A pricing model for big personal data. Tsinghua Science and Technology, 21(5):482–490.

[30] Adish Singla & Andreas Krause, (2013) Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. InProceedings of the 22nd international conference on World Wide Web, pages 1167–1178. ACM.

[31] Jie Song, Yixin Chen, Xinchao Wang, Chengchao Shen, & Mingli Song, (2019) Deep model

transferability from attribution maps. In Advances in Neural Information Processing Systems, pages 6179–6189.

[32] Pingzhong Tang, (2017) Reinforcement mechanism design. In IJCAI, vol-ume 17, pages 26–30.

[33] Surat Teerapittayanon & HT Kung, (2019) DaiMoN: A decentralized artificial intelligence model network. arXiv preprint arXiv:1907.08377.

[34] Guan Wang, Charlie Xiaoqian Dang, & Ziye Zhou, (2019) Measure contribution of participants in federated learning. arXiv preprint arX- iv:1909.08525.

[35] Yingjie Wang, Zhipeng Cai, Guisheng Yin, Yang Gao, Xiangrong Tong, & Guanying Wu, (2016) An incentive mechanism with privacy protection in mobile crowdsourcing systems. Computer Networks, 102:157–171.

[36] Zehui Xiong, Shaohan Feng, Dusit Niyato, Ping Wang, & Zhu Han, (2018) Optimal pricing-based edge computing resource management in mobile blockchain. In 2018 IEEE International Conference on Communications (ICC), pages 1–6. IEEE.

[37] Qiang Yang, Yang Liu, Tianjian Chen, & Yongxin Tong, (2019) Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2):12.

[38] Haifei Yu & Mengxiao Zhang, (2017) Data pricing strategy based on data quality. Computers & Industrial Engineering, 112:1–10.

[39] Zhenzhe Zheng, Yang Gui, Fan Wu, & Guihai Chen, (2014) Star: strategy-proof double auctions for multi-cloud, multi-tenant bandwidth reservation. IEEE Transactions on Computers, 64(7):2071–2083.

[40] Zhenzhe Zheng, Fan Wu, & Guihai Chen, (2014) A strategy-proof combinatorial heterogeneous channel auction framework in noncooperative wireless networks. IEEE Transactions on Mobile Computing, 14(6): 1123–1137.

[41] Zhenzhe Zheng, Fan Wu, Shaojie Tang, & Guihai Chen, (2015) Aegis: an unknown combinatorial auction mechanism framework for heterogeneous spectrum redistribution in noncooperative wireless networks. IEEE/ACM Transactions on Networking, 24(3):1919–1932.

[42] Zhenzhe Zheng, Fan Wu, Xiaofeng Gao, Hongzi Zhu, Shaojie Tang, & Guihai Chen, (2016) A budget feasible incentive mechanism for weighted coverage maximization in mobile crowdsensing. IEEE Transactions on Mobile Computing, 16(9):2392–2407.

[43] Zhenzhe Zheng, Yanqing Peng, Fan Wu, Shaojie Tang, & Guihai Chen, (2017) An online pricing mechanism for mobile crowdsensing data markets. In Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing, page 26. ACM.

[44] Zhenzhe Zheng, Yanqing Peng, Fan Wu, Shaojie Tang, & Guihai Chen, (2017) Trading data in the crowd: Profit-driven data acquisition for mobile crowdsensing. IEEE Journal on Selected Areas in Communications, 35(2):486–501.

RESEARCH ON FARMLAND PEST IMAGE RECOGNITION BASED ON TARGET DETECTION ALGORITHM

Shi Wenxiu and Li Nianqiang

School of Information Science and Engineering,
University of Jinan, Jinan, China

ABSTRACT

In order to achieve the automatic identification of farmland pests and improve recognition accuracy, this paper proposes a method of farmland pest identification based on target detection algorithm .First of all, a labeled farm pest database is established; then uses Faster R-CNN algorithm, the model uses the improved Inception network for testing; finally, the proposed target detection model is trained and tested on the farm pest database, with the average precision up to 90.54%.

KEYWORDS

Object detection algorithm, Faster R-CNN, Inception network

1. INTRODUCTION

The harm of farmland pests is one of the main factors affecting agricultural production, and its economic impact has spread all over the world. In this case alone, the annual agricultural economic losses in Europe reach 28.2%, North America reaches 31.2%, and the economic losses in Asia and Africa are as high as 50%[1]. For a long time, we have mostly adopted manual identification and counting methods for Integrated Pest Management (IPM)[2], which is closely related to the comprehensive quality of professionals. The subjective factors have a great influence on the accuracy and timeliness of pest control, which is not conducive to the effective work of farmland pest control. Therefore, it is necessary to study a fast and low-cost automatic detection method of farmland pests and diseases.

Li Wenbin [3] used a single pixel background segmentation method based on RGB color to segment rice pest images, and used SVM support vector machine to identify and classify. Compared with traditional recognition methods, the recognition rate is better, but the number of pest image samples is small, and the identification type is single. Pan Chunhua [4] and others used a grid search algorithm to improve the search efficiency of the image target pest area, and used SVM to train multiple classifiers to identify four major vegetable pests, with an average recognition rate of 93%.Yang Guoguo [5] used Otsu algorithm to find threshold adaptively to complete image segmentation. The SVM classifier was used to classify and identify the Chinese rice locust, with an accuracy rate of 88.3%.Yang Wenhan [6] used Canny operator and Otsu threshold segmentation method to segment 15 cotton pests, and used binary tree classification and support vector machine for classification and recognition, and the recognition effect was good. The actual environment of real-life farmland pest classification is very complicated, and there are many types of pests. In order to achieve more effective and wider application of farmland pest

Natarajan Meghanathan et al. (Eds): SIPP, BIGML, DaKM, SOEN, AISC - 2020
pp. 111-117, 2020. © CS & IT-CSCP 2020 DOI: 10.5121/csit.2020.100210

detection technology, this paper combines deep learning and field pest detection, and proposes a field pest identification method based on target detection algorithm, which greatly improves the accuracy of field pest detection.

2. MODEL BUILDING

2.1. Farmland Pest Database

A good sample set is the basis of image recognition research [7]. Because no public data for farmland pest detection is currently available, this article has collected 2,472 farmland pest image samples through the Internet for the goals and tasks of pest recognition. In order to prevent overfitting due to insufficient data during training, this paper expands the training samples. The main methods of image data expansion include: resize, scale, and noise noise, rotate, flip, zoom, zoom, shear, shift, contrast, random channel shift, Principal Component Analysis (PCA), etc. Finally, a data set of farmland pests including 10 categories was compiled, as shown in Table 1, with a total of 12,474 images.

2.2. Object Detection Model Design

2.2.1. Target Detection Algorithm Faster R-CNN

In order to realize the end-to-end operation of the entire network, Faster R-CNN unifies the region suggestion algorithm on the convolutional neural network [8]. This method does not need to manually select the suggestion region and can fully utilize the features extracted by the neural network.

Regional suggestion network (RPN) [9] is a set that takes an image of any size as input and outputs a rectangular target recommendation box, as shown in Figure 1. The regional recommendation network is connected to the last layer of the feature convolution layer. A small 3×3 network sliding window is used on the feature map of this layer. Only one sliding traversal can extract candidate windows for the entire image, reducing the network calculation burden. The point in the center of the sliding window is called the anchor, and the position of each anchor point can be mapped to the original image, corresponding to the target suggested area on the original image. In order to make the recommendation window meet the target needs without size, the network adopts a multi-scale method, so each sliding window has three scale ratios, and the aspect ratio is 1: 1, 1: 2, 2: 1, and 9 types of suggestion windows are generated with three scales, and there are K suggestion windows on the right side of the figure.

As shown on the left side of Figure 1, each sliding window generates a recommendation window, and the recommendation window is mapped to a low-dimensional vector, which is output to two fully connected layers at the same level, bounding box regression layer (reg) and bounding box classification layer (els). The bounding box regression layer contains the position information of each window. A single suggestion window has four coordinate values to determine the accuracy of the suggestion window generated by RPN. The purpose of the bounding box classification layer is to output the score of the target category of the recommendation window. Each recommendation window has two outputs, corresponding to the probability that the target recommendation window is the foreground / background. For the bounding box generated by the region suggestion network, the part outside the target suggestion region will be discarded, and the remaining regions will be assigned multiple binary labels (target or background). If the highest overlap ratio of the prediction area overlapping the ground truth box (Intersection-over-Union, IoU) is greater than the defined threshold, a positive label is assigned to it. If the IOU ratio of the prediction area is lower than the defined threshold, a negative label is assigned to it. The definition of IoU is as follows:

$$IoU = \frac{area(B_{in\sec t} \cap B_{group})}{area(B_{in\sec t} \cup B_{group})} \quad (1)$$

$area(B_{in\sec t} \cap B_{group})$ represents the overlapping area of the target recommendation area and the ground truth area, $area(B_{in\sec t} \cup B_{group})$ represents the union of the target recommendation area and the ground truth area.

RPN and Fast R-CNN use the same loss function. The calculation formula for this multi-tasking loss function is as follows:

$$L(\{p_i\},\{p_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (2)$$

N is the total amount of anchors, i is the index of anchors in a mini-batch, and p_i is the predicted probability of the i-th anchor. If anchor is positive, the ground truth label p_i* is 1, and if anchor is negative, p_i* is 0.

t_i represents the four parameterized vectors of the coordinates of the predicted rectangular frame, and t_i* is the coordinate vector of the ground truth corresponding to the positive anchor. Classification loss (L_{cls}) is the log loss of two categories (foreground and background).

$$L_{cls}(p_i, p_i^*) = -\log[p_i^* p_i + (1 - p_i^*)(1 - p_i)] \quad (3)$$

Return loss is L_{reg} :

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*) \quad (4)$$

2.2.2. Inception Network

This article considers the use of Googlenet Inception [10] structure to build an Inception network. A simple Inception network contains 22 layers of deep network. Because of the problem of blocked information flow, the optimization ability of the deep model is greatly reduced. To solve this problem, Googlenet [11] added two additional Softmax layers to calculate the new loss value, and then recalculated the network gradient based on the new loss value. Ballester [12] proposed the use of Shortcut Connection to reduce the effect of gradient disappearance. This structure can also reduce the Degradation phenomenon. However, some farmland pests have smaller targets. Using the simple concept structure can not solve the problem of pest detection and recognition in specific situations. Therefore, this paper uses an improved Inception network [13] for farmland pest detection,the improved neural network is shown in Figure 2.

As can be seen from Figure 2, the improved Inception [13] network uses the Shortcut Connection, and a deconvolution layer is connected behind the second Inception structure, and the feature map is doubled to the original one. At the same time, after extracting the feature map of the seventh perception structure, connecting a full connection layer will reduces the dimension of the feature map to the feature vector of 1024 dimension, Then the two feature vectors are stitched into a 2048-dimensional feature vector, and then the 2048-dimensional feature vector is reduced to a 1024-dimensional feature output as a feature output of the picture. And the Inception network can directly transfer the gradient from the deep layer of the network back to the shallow layer. At the same time, the Shortcut Connection can extract the shallow feature map. The scale of the shallow

feature map is 1/8 of the original image. After the deconvolution layer, the feature map is increased to 1/4 of the original image. This kind of network combining multiple layers of features and different scales has better ability to detect and recognize targets, and can get better results in farmland pest recognition.

2.3. Experimental Results and Analysis

2.3.1. Experimental Results

When using the farm pest database detection model to train, in order to ensure that all samples can be used for training and testing, this paper uses K-fold cross-validation [14], where K is selected 10 and 9 subsets are selected for training Data, 1 subset as test data. The results of this experiment were evaluated using mean Average Precision (mAP). The formula is as follows:

$$mAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q} \times 100\% \tag{5}$$

Q is the total number of pest categories, and AP(q) is the average accuracy rate of the detection results of category.

The average accuracy of the experimental results and the final average of the average accuracy are shown in Figure 3. From Figure 3, it can be seen that the average accuracy of detection of most types of farm pests is high, and the average average accuracy can be seen from the broken lines in the figure mAP reached 90.54%.

2.3.2. Experimental Comparison

In this paper, two detection methods are selected for comparison. The first is a method based on SVM [15] for agricultural pest detection proposed by Mundada et al. This paper does not preprocess the test data for the convenience of training data and the uniformity of comparison experiments. The second is a pest detection model proposed by Liu et al[16]. Which is modified on the native AlexNet network structure. In order to ensure the uniformity of training data, this paper replaces the fully connected layer in AlexNet with the Global Average Pooling (GAP) [17] can ensure that the input image data does not need to be of a fixed size. The same farmland pest data set was used to test with the method proposed in this paper and SVM and AlexNet models, and the comparison results are listed in Table 2. It can be found that the classification accuracy of the monitoring model used in this paper is improved by about 17% compared with AlexNet, and the overall detection classification accuracy has been greatly improved.

2.4. Figures and Tables

Table 1. Heading and text fonts.

Category	**Quantity**	**Proportion**
cutworms	1213	9.72%
aphidoidea	1220	9.78%
armyw	1202	9.64%
chafer	1205	9.66%
locust	1225	9.82%
plantho	1297	10.40%

psilogramma menephron	1315	10.54%
ostrinia furnacalis	1207	9.68%
clanis bilineata	1285	10.30%
helicoverpa armigera	1305	10.46%

Table 2. Comparison of test results

Method	**Mean precision /%**
SVM	52.89
AlexNet	73.57
Method of this article	90.54

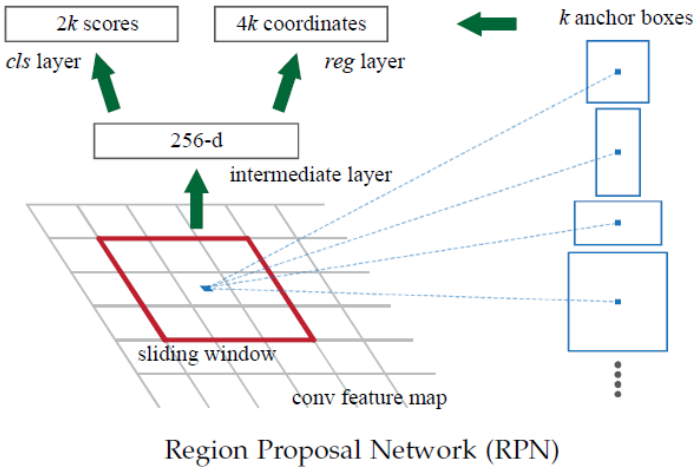


Figure 1. Faster R-CNN framework

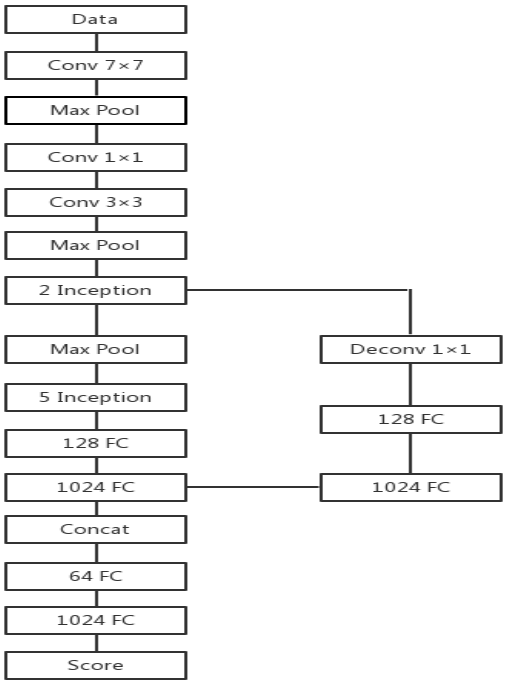


Figure 2. Inception network structure diagram

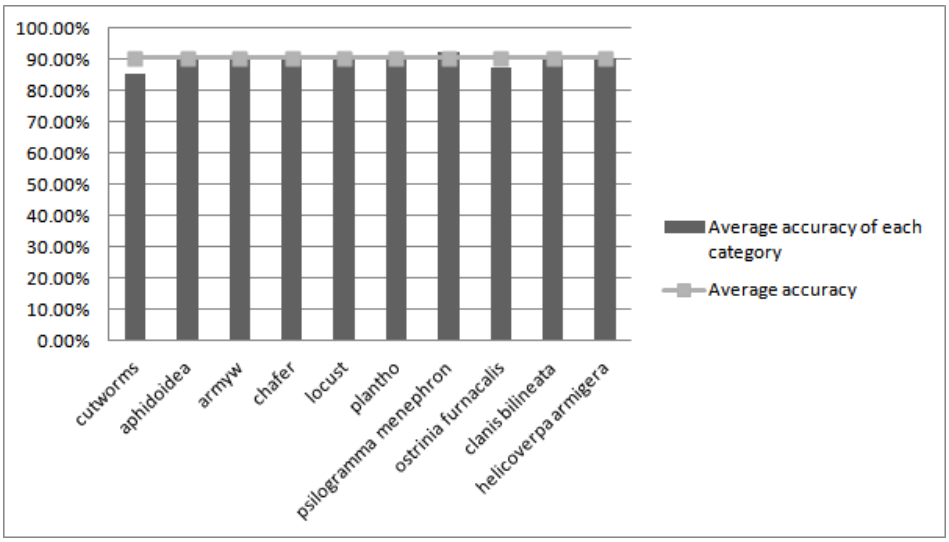


Figure 3. Farm pest detection results

3. CONCLUSIONS

Based on the idea of target detection algorithm in deep learning, this paper proposes a farmland pest detection model based on the target detection algorithm. This model combines a faster R-CNN algorithm and uses an improved Inception network, and test this model with a established farmland pest database. The detection accuracy rate of 90.54% was obtained. The experimental results show that the detection model proposed here can well perform the detection of farmland pests, and the detection results are accurate and the detection speed is fast. However, the method of this paper still has some shortcomings. The improved Inception network also needs to design a lot of hyperparameters. In the experiments, it is impossible to avoid the complicated hyperparameter tuning process, which brings the risk of overfitting to the recognition model.

ACKNOWLEDGEMENTS

The successful completion of this article is inseparable from the meticulous guidance and supervision of my tutor, as well as the help and encouragement of the lab brothers and sisters. Thank everyone.

REFERENCES

[1] Li Y , Xia C , Lee J . Detection of small-sized insect pest in greenhouses based on multifractal analysis[J]. Optik - International Journal for Light and Electron Optics, 2015, 126(19):2138-2143.

[2] Parsa S , Morse S , Bonifacio A , et al. Obstacles to integrated pest management adoption in developing countries[J]. Proceedings of the National Academy of Sciences, 2014, 111(10):3889-3894.

[3] Li Wenbin. Research on rice pest image recognition technology based on SVM [D]. Hangzhou University of Electronic Science and technology, 2015.

[4] Pan Chunhua, Xiao Deqin, Lin Tanyu, et al. Classification and identification of major vegetable pests in South China based on SVM and regional growth algorithm [J]. Journal of agricultural engineering, 2018 (8): 192-199

[5] Yang Guoguo. Identification and detection of early locust pupae of Chinese rice locust based on machine vision [D]. Zhejiang University, 2017.

[6] Yang Wenhan. Research on cotton pest identification system based on digital image processing [D]. Sichuan Agricultural University, 2015.

[7] Qin Fang. Insect image recognition based on deep learning [D]. Southwest Jiaotong University,2018:31-34

[8] Krizhevsky A , Sutskever I , Hinton G . ImageNet Classification with Deep Convolutional Neural Networks[C]// NIPS. Curran Associates Inc. 2012:25(2)

[9] Xia Denan. Research on agricultural insect image recognition based on deep learning [D]. Anhui University, 2019:26-30.

[10] Szegedy C , Liu W , Jia Y , et al. Going Deeper with Convolutions[J]. 2014.

[11] 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 Computer Society, 2016.

[12] Ballester P L , Araujo R M . On the performance of GoogLeNet and AlexNet applied to sketches[C]// AAAI. AAAI Press, 2016.

[13] Shen Yufeng.Research on Stored Grain Pest Detection Algorithm Based on Deep Learning[D].2018:36-45.

[14] Hu Juxin,Zhang Gongjie.Selective ensemble classification algorithm based on K-fold cross validation[J].Science and Technology Bulletin,2013(12):123-125.

[15] Rupesh G. Mundada Rupesh G. Mundada.Detection and Classification of Pests in Greenhouse Using Image Processing[J].IOSR Journal of Electronics and Communication Engineering,2013,5（6）：57-63.

[16] Yang G , Bao Y , Liu Z . Localization and recognition of pests in tea plantation based on image saliency analysis and convolutional neural network[J]. Transactions of the Chinese Society of Agricultural Engineering, 2017, 33(6):156-162.

[17] He Kaiming, Gkioxari Georgia, Dollar Piotr,et. Mask R-CNN[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence:1-1.

AUTHOR

Shi Wenxiu, 1995.03.01, School of Information Science and Engineering, University of Jinan, Postgraduate, Mainly engaged in image processing research.

AUTHOR INDEX

Alexandria Dominique Farias	*39*
Chao Wu	*99*
Faisal Saeed	*21*
Gongling Sun	*39*
Guofengli	*83*
Hager Ali Yahia	*91*
Hatem Awed Khater	*91*
Hentabli Hamza	*21*
Hitoshi Kiya	*01*
Kiran Kumaar CNK	*75*
Li Nianqiang	*111*
Maged Nasser	*21*
Mark Atkins	*59*
Mohammed Rizk Mohammed	*91*
Mohammed Zakaria Moustafa	*91*
Naomie Salim	*21*
Shi Wenxiu	*111*
Sven Lončarić	*13*
Takayuki Nakachi	*01*
Vedran Stipetić	*13*
Zheqi Lv	*99*
Zuoqi Tang	*99*