Princess Sumaya University for Technology
جامعة الأميرة سميّة للتكنولوجيا

صندوق دعم البحث العلمي
Scientific Research Support Fund

Jordanian Journal of Computers and Information Technology

Decemper 2021 VOLUME 07 NUMBER 04
ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

JJCIT

www.jjcit.org jjcit@psut.edu.jo

An International Peer-Reviewed Scientific Journal
Financed by the Scientific Research Support Fund

Jordanian Journal of Computers and Information Technology (JJCIT)

The Jordanian Journal of Computers and Information Technology (JJCIT) is an international journal that publishes original, high-quality and cutting edge research papers on all aspects and technologies in ICT fields.

JJCIT is hosted and published by Princess Sumaya University for Technology (PSUT) and supported by the Scientific Research Support Fund in Jordan. Researchers have the right to read, print, distribute, search, download, copy or link to the full text of articles. JJCIT permits reproduction as long as the source is acknowledged.

Aims and Scope

The JJCIT aims to publish the most current developments in the form of original articles as well as review articles in all areas of Telecommunications, Computer Engineering and Information Technology and make them available to researchers worldwide. The JJCIT focuses on topics including, but not limited to: Computer Engineering & Communication Networks, Computer Science & Information Systems and Information Technology and Applications.

Indexing

JJCIT is indexed in:

ELSEVIER Scopus

DOAJ DIRECTORY OF OPEN ACCESS JOURNALS

crossref Text and Data Mining

OCLC WorldCat

BASE

ACADEMIA

Google scholar

JJCIT Address

Website: www.jjcit.org
Email: jjcit@psut.edu.jo
Address: Princess Sumaya University for Technology, Khalil Saket Street, Al-Jubaiha
B.O. Box: 1438 Amman 11941 Jordan
Telephone: +962-6-5359949
Fax: +962-6-7295534

Editorial Board

International Advisory Board

"Opinions or views expressed in papers published in this journal are those of the author(s) and do not necessarily reflect those of the Editorial Board, the host university or the policy of the Scientific Research Support Fund".

"ما ورد في هذه المجلة يعبر عن آراء الباحثين ولا يعكس بالضرورة آراء هيئة التحرير أو الجامعة أو سياسة صندوق دعم البحث العلمي".

UNCONSTRAINED EAR RECOGNITION USING TRANSFORMERS

Marwin B. Alejo

(Received: 3-Aug.-2021, Revised: 5-Sep.-2021, Accepted: 12-Sep.-2021)

ABSTRACT

The advantages of the ears as a means of identification over other biometric modalities provided an avenue for researchers to conduct biometric recognition studies on state-of-the-art computing methods. This paper presents a deep learning pipeline for unconstrained ear recognition using a transformer neural network: Vision Transformer (ViT) and Data-efficient image Transformers (DeiTs). The ViT-Ear and DeiT-Ear models of this study achieved a recognition accuracy comparable or more significant than the results of state-of-the-art CNN-based methods and other deep learning algorithms. This study also determined that the performance of Vision Transformer and Data-efficient image Transformer models works better than that of ResNets without using exhaustive data augmentation processes. Moreover, this study observed that the performance of ViT-Ear is nearly like that of other ViT-based biometric studies.

KEYWORDS

Deep learning, Neural networks, Transformers, Vision transformer, Data-efficient image transformers, Ear recognition.

1. INTRODUCTION

Biometric recognition is an information system technology that allows identifying any person by his/her unique personal characteristics. Several studies use the common unique traits of an individual, such as fingerprint [1]–[3], face [4], [5], iris [6]–[8], iris and voice [9], [10], gait [11], [12] and ECG and EEG [13]–[15] for biometric recognition. However, recent studies suggested using ears for biometric recognition due to its advantages over using each of these common biometric traits [16]–[19]. Ear-based biometric recognition is both a science and technology that identify and authenticate individuals by their ear images in a constrained or unconstrained environment [20]. This method gained a momentum of interest in computational method research and application due to many advantages over other forms of biometric recognition. Although ear recognition offers numerous advantages over fingerprint, iris and face, it still faces significant levels of difficulty and challenges in unconstrained environments [21]-[22].

Modern studies utilize image processing algorithms, machine learning techniques or the fusion of both for the computational method of ear-based biometric recognition. One of these papers that utilizes these algorithms is Kavipriya et al. [23]. Similar to the enhanced method of Cheribet and Mazouzi [24], their method uses the canny edge detection algorithm and contour tracking method for ear biometric and personal identification. The paper of Mangayarkarasi et al. [25] proposed the same ear recognition method, but using only the contour method. The study of Jiddah and Yurtkan [26] presented an ear recognition method utilizing the used ear image dataset's fused geometric and texture features. The works of Zarachoff et al. [27] presented the 2D Wavelet-based Multi-Band PCA (2DWMBPCA) method, inspired by PCA (Principal Component Analysis) –a machine learning technique– for an ear-based biometric recognition. The paper of Sajadi et al. [28] utilized the genetic algorithm to extract the local and global features of ear images for ear recognition. While these ear biometric methods achieved exemplary results, most recent studies suggested using deep learning algorithms –a machine learning technique– in developing an ear-based biometric recognition method.

Deep learning algorithms are the most prevalent technology applied in the computational studies of ear recognition methods. Most of these deep learning studies utilize an improved architecture to learn from a single image [29]. The paper of Khaldi et al. [30] proposed the use of deep unsupervised active learning for ear recognition using the AMI (Mathematical Image Analysis), USTB2 (University of

M. Alejo is with Electrical and Electronics Engineering Institute, University of the Philippines – Diliman, Quezon City, Philippines and also is with Computer Engineering Department, National University, Manila, Philippines. Emails: marwin.alejo@eee.upd.edu.ph and mbalejo@national-u.edu.ph, ORCID:0000-0002-0926-1261.

Science and Technology Beijing), AWE (Annotated Web Ears) datasets and GAN (Generative Adversarial Network) for image coloring. Their method achieved recognition rates of 100.00%, 98.33% and 51.25% on the used datasets. The works of Lei et al. [31] used the SSD_MobileNet_v1 model on USTB datasets and achieved a recognition accuracy of 99%. Ying et al. [32] designed a DCNN (Deep Convolutional Neural Network) architecture called ear-recognition-Net for the ear recognition task. Their approach achieved a recognition rate of 95% to 98%. Chowdhury et al. [33] proposed using a handcrafted neural network algorithm for robust ear recognition and achieved a recognition accuracy of 98.2%. The study of Alshazly et al. [34] proposed using pre-trained AlexNet, VGGNet, Inception, ResNet and ResNeXt models for unconstrained ear recognition EarVN1.0 dataset by transfer learning and fine-tuning. Their method determined that ResNeXt is the best model for the task with a recognition accuracy of 95.85%. On a similar note, the papers of Alejo and Hate [35] and Almisreb et al. [36] utilized transfer learning for unconstrained ear recognition tasks on pre-trained deep convolutional neural network models. The works of Alejo and Hate achieved the recognition accuracy of 97.3%, 93.3%, 96.7%, 94.7%, 100.00%, 96.7%, 87.3%, 86.7% and 81.3% on AlexNet, GoogLeNet, Inception-v3, Inception-ResNet, ResNet-18, ResNet-50, SqueezeNet, ShuffleNet and MobileNet, while 100% was obtained on AlexNet in the works of Almisreb et al., one of the newest developed algorithms of deep learning. Although unrelated to ear recognition, the papers of Zhong and Deng [37] and George and Marcel [38] are among the early studies that adapted TNN or Transformer Neural Network as part of the method of their face recognition pipeline. Moreover, no published or presented study proposed the use of Transformer Neural Network for ear recognition; hence, an open opportunity.

Inspired by the facts and studies above, this paper aimed to investigate the effectiveness of the Transformer Neural Network on unconstrained ear recognition in terms of recognition accuracy performance. Furthermore, this paper (1) provided a deep learning pipeline for unconstrained ear biometric recognition by ViT (Vision Transformer) and DeiT (Data-efficient image Transformer) models and (2) compared the recognition accuracy performance of ViT and DeiT with the recognition accuracy performance of other methods based on deep learning, particularly the CNN.

The organization of the rest of this paper is as follows: Section 2 of this paper discusses Transformers; Section 3 discusses the Transformer-driven deep learning pipeline of this paper; Section 4 compares and discusses the results of this paper with the results of other relevant studies and Section 5 discusses the conclusion of this study.

2. Transformers and their Vision-centric Models

Transformer Neural Network (or Transformers) is a novel deep learning technique developed by Vaswani et al. [39] with a self-attention mechanism as its core. It is a simple and scalable solution that exceeds the state-of-the-art results of the architectures based on RNN (Recurrent Neural Network) and CNN (Convolutional Neural Network) on NLP tasks (Natural Language Processing). The ongoing effort of several studies extended Transformers onto Computer Vision tasks [40], allowing the introduction of deep learning models, like DETR (Detection Transformer) [41] and Deformable DETR [42] for object detection, Axial-DeepLab [43] and Cross-Model Self-Attention [44] for image segmentation, Image Transformer [45], Image GPT [46], Transformer-induced Biases [47], TransGAN [48] and SceneFormer [49] for image generation and CLIP (Contrastive Language-Image Pre-training) [50], ViT (Vision Transformer) [51] and DeiT (Data-efficient image Transformers) [52] for object recognition. Furthermore, this paper focuses only on implementing Transformers through Vision Transformer and Data-efficient image Transformer models due to constraints with the used computational resources. The following subsections briefly discuss these two models of object recognition.

2.1 Vision Transformer (ViT)

The absence of an end-to-end object recognition architecture through Transformers provided an avenue for the development of Vision Transformer or ViT. Vision Transformer (ViT) is a brainchild algorithm of Dosovitskiy et al. [51] that employs a modified transformer network architecture to operate on images instead of text directly. Vision Transformer aims to provide an object recognition architecture without relying on CNN. Moreover, while Vision Transformer consumes extensive computational resources during implementation over CNN [53], the works of Naseer et al. [54] emphasize that (1) ViT demonstrates strong robustness over occlusions, spatial patch-level permutations, adversarial perturbations and common natural signal corruptions for object recognition, (2) ViT performance for shape recognition is comparable to that of humans and (3) ViT exceptionally generalizes pre-trained ImageNet models or transfer learning for new domains of object recognition.

The Vision Transformer of this paper divides the input image into grid square patches flattened into a single vector by joining all the channels of pixels in a patch and linearly injecting each by the desired dimension. Moreover, this model employs a learnable position embedding into each patch and allows the Transformer network to learn the image's positional patch. Figure 1 shows the architecture of the Vision Transformer for unconstrained ear recognition (Vit-Ear) as adapted from the original paper [51]. Like the concept of transfer learning in CNN, the ViT architecture of this study replaced and fine-tuned the final layer to satisfy the recognition requirements accordingly.

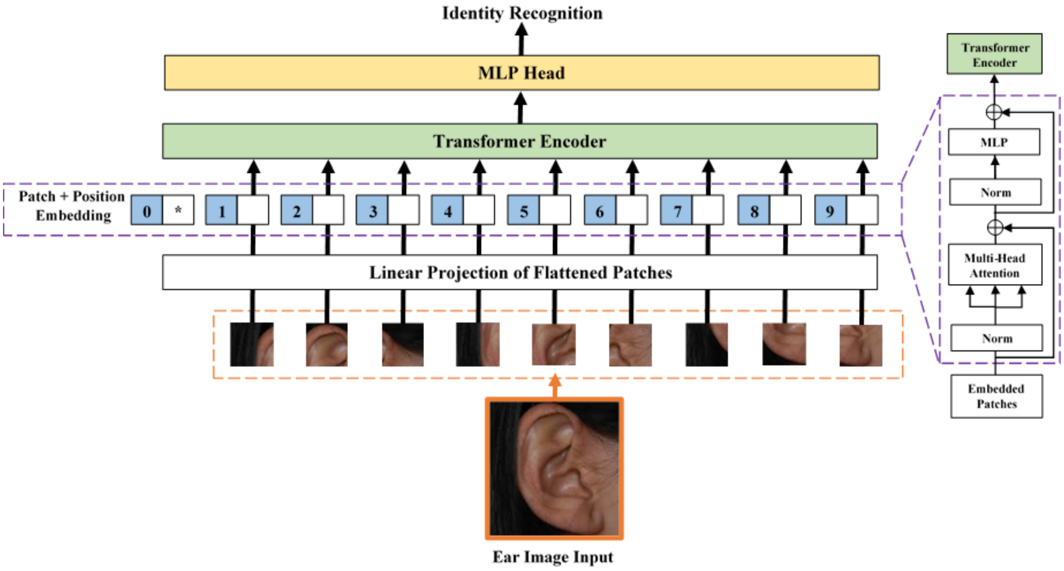


Figure 1. Vision transformer architecture of this study as adapted from the original paper [51].

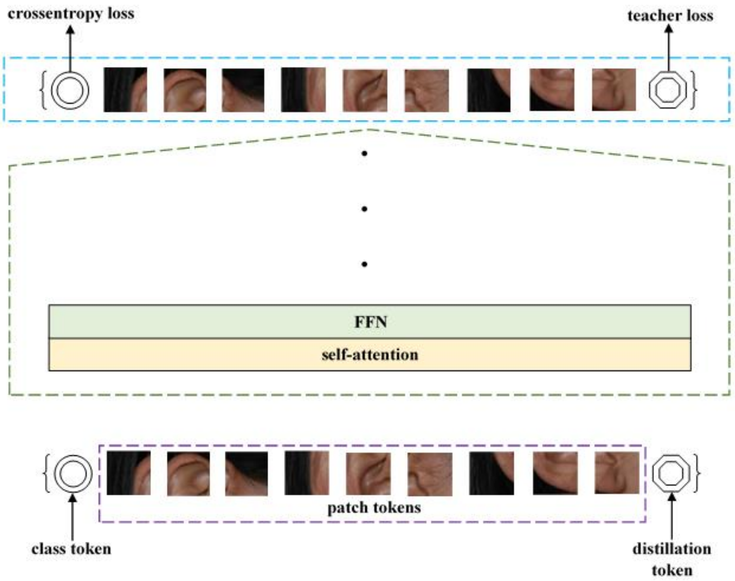


Figure 2. Data-efficient image transformer architecture of this study as adapted from the original paper [52].

2.2 Data-efficient Image Transformer (DeiT)

Data-efficient image Transformers (DeiTs) is another Transformer-based object recognition algorithm developed by Touvron et al. [52] with ViT in its core. DeiT aimed to overcome the excessive usage of computational resources while exceeding the performance accuracy of CNN-based methods for object recognition. DeiT can maintain a ~60% accuracy on object recognition while zero accuracies were obtained for CNN on ImageNet task [54]. This is due to DeiT's use of a teacher-student strategy on its Transformer neural network to train directly on the used datasets. This teacher-student strategy of DeiT relies on distillation tokens to ensure that the student (model) learns through attention from the teacher. Figure 2 shows the DeiT architecture of this study (DeiT-Ear) as adapted from the original paper [52].

3. METHODOLOGY

The methodology of this study consisted of four subsequent phases: (1) Dataset and Input Data, (2) Data Preprocessing, (3) Training and Modeling and (4) Classification. Figure 3 visually shows these phases.

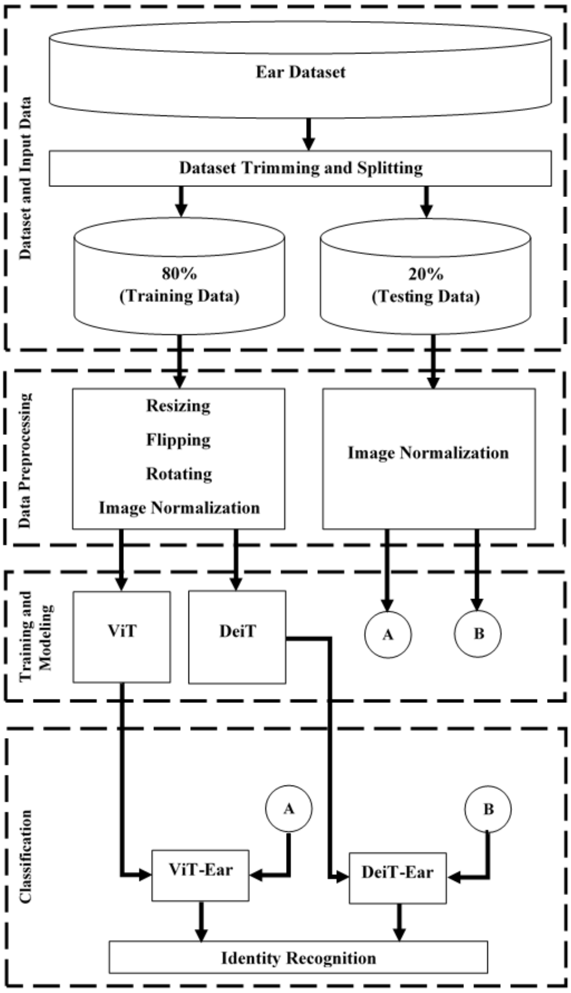


Figure 3. Deep learning pipeline of this study.

3.1 Dataset and Input Data

This study uses two different ear databases: (1) EarnVN1.0 dataset [55] and (2) UERC (Unconstrained Ear Recognition Challenge) dataset [56]. The EarVN1.0 dataset is the largest ear images dataset mainly collected for the task of ear recognition. It consisted of unprocessed ear images in the wild (unconstrained) of 164 individuals, with each having ~180 images for a totality of 28,412 ear images. The UERC dataset consisted of 3,300 ear images of 330 distinct identities. Due to the computational resources' constraint, this study considered only the first 20 classes of the EarVN1.0 dataset for a total of ~4000 ear images and the first ten classes of the UERC dataset for a total of 100 ear images. Furthermore, this study partitioned the trimmed dataset by 80% training and 20% testing dataset. The output of this phase is a set of training and testing datasets of the two databases. Figures 4 and 5 show samples of ear images of the used EarVN1.0 and UERC datasets.

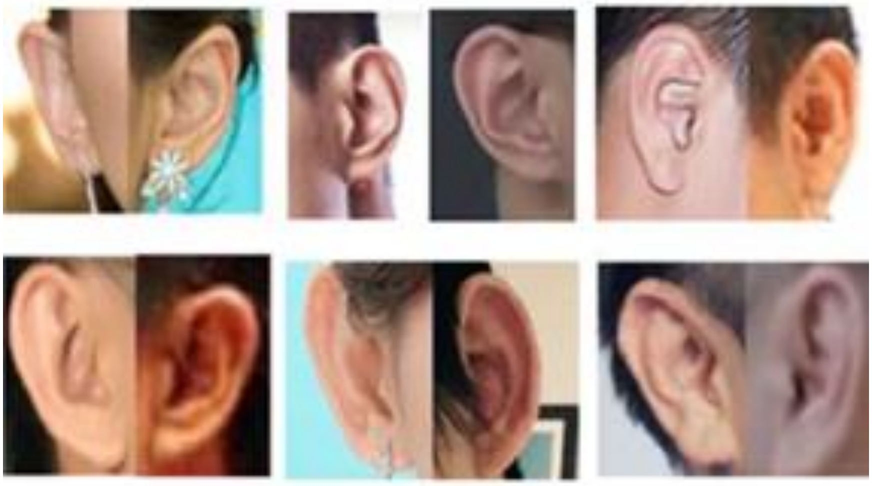


Figure 4. Sample ear images of EarVN1.0 dataset.

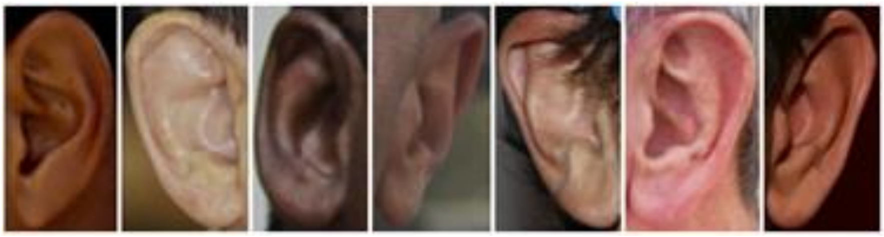
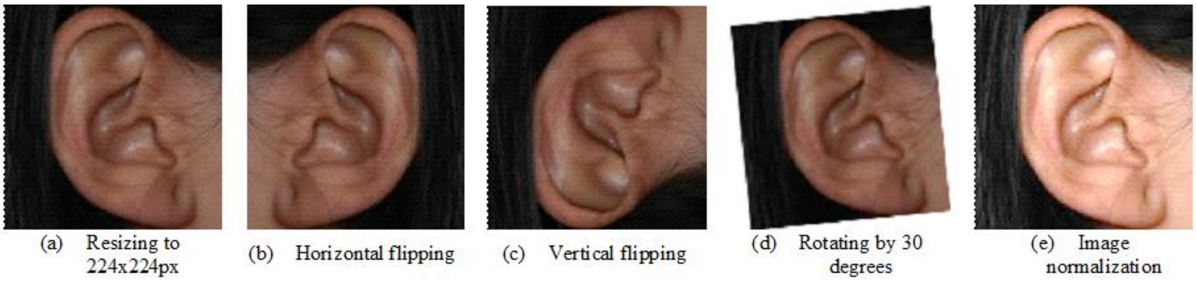


Figure 5. Sample ear images of UERC dataset.

3.2 Data Pre-processing

This study pre-processed the partitioned training dataset by (a) resizing each ear image into 224 square pixels, (b) horizontal and (c) vertical flipping, (d) rotating by 30 degrees and (e) normalization using the standard ImageNet normalization values. Moreover, this study pre-processed the testing/validation dataset by resizing 224 square pixels and normalizing each resized ear image using the standard ImageNet normalization values. The output of this phase is a set of pre-processed training and testing/validation ear images. Figure 6 shows the sample output of these processes.



(a) Resizing to 224x224px
(b) Horizontal flipping
(c) Vertical flipping
(d) Rotating by 30 degrees
(e) Image normalization

Figure 6. Sample output of each data processing process.

3.3 Training and Modeling – ViT and DeiT Implementation

Following both the description of ViT and DeiT in their respective papers [51], [52] and the context of transfer learning [50], [57], [58] due to the statistics of the used datasets, this study implemented these architectures with their pre-trained ImageNet-21k model and fine-tuned each of the architecture's final layers to only recognize 20 people from the used EarVN1.0 dataset and ten people from the used UERC dataset. This study implemented ViT using PyTorch-XLA on Google Colab TPU with eight as batch size, a learning rate of 0.00002, a gamma rate of 0.7 and 20 epochs. On the contrary, this study implemented the DeiT using PyTorch on Google Colab GPU with 32 as batch size, a learning rate of 0.001 and epoch values of 20, 30, 40 and 50. The used optimizer of this study in both ViT and DeiT is Adam. Table 1 summarizes the used training and modeling configuration of this study in implementing ViT and DeiT.

Table 1. Training and modeling configuration of this study.

Configuration	ViT	DeiT
Batch size	8	32
Learning rate	0.00002	0.001
Epoch	20	20, 30, 40, 50
Optimizer	Adam	Adam
Gamma	0.7	Not applicable

3.4 Classification

This phase utilizes the pre-processed testing datasets of this study. This phase aimed to determine the recognition accuracy of the trained unconstrained ear recognition models of this study on ViT and DeiT. The output of this phase is a comparative analysis of the recognition performance of ViT and DeiT in the context of unconstrained ear recognition over the recognition accuracy of other existing deep learning methods, like CNN. Since there are no published studies that proposed the use of Transformers for ear recognition, this paper considered comparing the results of this study with those of the existing CNN-based unconstrained ear recognition studies and considered transfer learning as the common ground.

4. Results and Discussion

4.1 ViT and DeiT on EarVN1.0

This paper's trained unconstrained ear recognition model on Vision Transformer (ViT) achieved a training accuracy of 100.00% and a recognition accuracy of 95.31% with a loss of 26.36%. The implementation of this model took 20 minutes and 40 seconds to train the ViT model on the preprocessed EarVN1.0 training dataset. This study also observed that overfitting occurs when training the ViT beyond 20 epochs. Overfitting is a phenomenon when the observed recognition accuracy is higher than the observed training accuracy [59].

On the contrary, the implementation of DeiT on the preprocessed EarVN1.0 dataset took ~15 minutes. The trained DeiT model of this study achieved a training accuracy of 100.00% in all the specified epoch configurations and a recognition accuracy of 88.33% with a loss of 1.4% on 20 epochs, 93.33% recognition accuracy with 1.02% loss on 30 epochs, 96.11% recognition accuracy with 0.88% loss on 40 epochs and 96.11% recognition accuracy with 0.69% loss on 50 epochs. This paper observed that recognition accuracy remains at 96.11% when training the DeiT model beyond 50 epochs.

4.2 ViT and DeiT on UERC

Given that the used UERC dataset of this study consisted of only ten subjects with each having ten images, the ViT model of this paper on the UERC dataset achieved a training accuracy of 100.00% and a recognition accuracy of 96.48% with a loss of 20.08%. The implementation of this model took ~16 minutes to train on the preprocessed UERC training dataset. Like the observations on the implementation of ViT on the EarVN1.0 dataset, overfitting occurs in this ViT implementation when training beyond 20 epochs.

The implementation of DeiT on the preprocessed UERC dataset took ~10 minutes. It achieved a training accuracy of 100% in all the epoch configurations and a recognition accuracy of 94.45% with a loss of 0.98% on 20 epochs, 97.81% recognition accuracy with a loss of 0.93% on 30 epochs and 100.00% recognition accuracy on 40 to 50 epochs with a loss of 0.43% to 0.51%. Overfitting also occurred in this implementation when training on epochs beyond 50.

4.3 Comparative Results

These recognition results of ViT and DeiT are closely comparable to the results of the relevant studies on state-of-the-art CNN-based methods through transfer learning. The results of this paper achieved a comparable result to the works of Lei et al. [31], Alshazly et al. [34], Alejo and Hate [35] and Almisreb et al. [36]. The performance of the trained DeiT model of this paper on EarVN1.0 on 20 epochs is similar to the recognition accuracy performance of the SqueezeNet and ShuffleNet models of Alejo and Hate, while the trained ViT and DeiT models on EarVN1.0 on 30 to 50 epochs and

UERC on 20 epochs are comparable to the performance of inception-based, ResNet-50 and MobileNet models of Alejo and Hate and the ResNext model of Alshazly et al. The trained DeiT models on UERC with 30 to 50 epochs achieved a comparable result over the SSD_MobileNet model of Lei et al. and the AlexNet and ResNet-18 models of Alejo and Hate and Almisreb et al. Furthermore, Alejo and Hate took 2.5 hours to develop their ResNet models for the unconstrained ear recognition task considering extensive data augmentations to achieve 100% recognition accuracy, while the transformer network of this paper took ~10 minutes to achieve the same recognition accuracy without relying heavily on data augmentation. Hence, this paper also proved the claim of Chen et al. [60] that Vision Transformers can achieve similar or more excellent results than ResNets even without extensive data augmentation. Table 2 shows the comparative results of the method of this study with those of the other related studies.

Table 2. Comparative results of this study over the results of methods of other CNN-based studies.

Study	**Dataset**	**Common Method**	**Method**	**Results in % (Recognition Accuracy)**
This study	EarVN1.0	Transfer Learning	Vision Transformer	95.31
			Data-efficient image Transformers	88.33 @ 20 epochs
				93.33 @30 epochs
				96.11 @ 40-50 epochs
This study	UERC	Transfer Learning	Vision Transformer	96.48
			Data-efficient image Transformers	94.45 @ 20 epochs
				97.81 @30 epochs
				100.00 @ 40-50 epochs
Lei et al. [27]	USTB2 (University of Science and Technology Beijing)	Transfer Learning	SSD_MobileNet_v1	99.00
Alshazly et al. [30]	EarVN1.0 [49]	Transfer Learning	ResNeXt	95.85
Alejo and Hate [31]	Handcrafted (own)	Transfer Learning	AlexNet	97.30
			GoogLeNet	93.30
			Inception-v3	96.70
			Inception-ResNet	94.70
			ResNet-18	100.00
			ResNet-50	96.70
			SqueezeNet	87.30
			ShuffleNet	86.70
			MobileNet	91.30
Almisreb et al. [32]	Handcrafted (own)	Transfer Learning	AlexNet	100.00

5. CONCLUSION

This paper investigated the use of Transformers in unconstrained ear recognition, particularly the Vision Transformer (ViT) and Data-efficient image Transformers (DeiT). This paper also provided a deep learning pipeline that employed these models. Like the concept of transfer learning on pre-trained state-of-the-art CNN architectures, this study replaced the final layer of ViT and DeiT to enable the Transformer network to learn the features from the extracted training ear images of the EarVN1.0 and UERC datasets. The ViT-Ear or Vision Transformer on the unconstrained ear recognition model of this study achieved a recognition accuracy of 95.31% on EarVN1.0 dataset and 96.48% on UERC dataset. The DeiT-Ear or Data-efficient image Transformers on this paper's unconstrained ear recognition model achieved a recognition accuracy of 88.33%, 93.33% and 96.11% on EarVN1.0 dataset and 94.45%, 97.81% and 100.00% on UERC dataset.

This paper determined that Transformers through ViT and DeiT achieved comparable or excellent results compared to state-of-the-art CNN-based methods for unconstrained ear recognition. Both the

ViT and DeiT achieved a similar recognition score of Inception-v3 and ResNet-50, but with faster modeling time, thus proving that Transformer networks work similarly or better than ResNets regardless of the particularity of the computer vision task. Additionally, this paper observed that the performance of ViT-Ear is like the recognition rate of a recently published face recognition method based on ViT, hence inferring that ViT might achieve an approximate 95% in biometric recognition studies regardless of the used modalities.

Future studies suggest exploring and investigating the performance of DeepVit (Deep Vision Transformer), CaiT (Class-Attention in Image Transformers), T2TViT (Tokens-to-Tokens Vision Transformer), CrossViT (Cross-Attention Multi-Scale Vision Transformer), PiT (Pooling-based Vision Transformer), LeViT (Vision Transformer in ConvNet's Clothing for Faster Inference) and CvT (Convolutions to Vision Transformers) on ear recognition and other biometric recognition modalities. Since this paper is an initial study of the ear recognition on Transformers and considering the limitations of conducting the experiments of this study, the proponent also suggested providing a comparative performance of these Transformer models over the results of this study.

ACKNOWLEDGMENTS

The author would like to acknowledge Professor Rowel Atienza of EEE Institute of the University of the Philippines – Diliman for his guidance and knowledge contribution towards the development of this study. Also, the author would like to acknowledge Dr. Rhandley Cajote for allowing the author to partake in the DSP Laboratory of the same institute. Moreover, this research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors and did not have conflict of interests from anybody.

REFERENCES

[1] S. Shi, J. Cui, X. L. Zhang, Y. Liu, J. L. Gao and Y. J. Wang, "Fingerprint Recognition Strategies Based on a Fuzzy Commitment for Cloud-Assisted IoT: A Minutiae-based Sector Coding Approach," IEEE Access, vol. 7, pp. 44803–44812, DOI: 10.1109/ACCESS.2019.2906265, 2019.

[2] I. Elzein and M. Kurdi, "Analysis of Embedded Fingerprint Biometric Recognition System Algorithm," Proc. of the 12th IEEE International Symposium on Advanced Topics in Electrical Engineering (ATEE 2021), DOI: 10.1109/ATEE52255.2021.9425124, Bucharest, Romania, Mar. 2021.

[3] M. H. Hersyah, D. Yolanda and H. Sitohang, "Multiple Laboratory Authentication System Design Using Fingerprints Sensor and Keypad Based on Microcontroller," Proc. of the IEEE International Conference on Information Technology Systems and Innovation (ICITSI 2020), pp. 14–19, DOI: 10.1109/ICITSI50517.2020.9264969, Bandung, Indonesia, Oct. 2020.

[4] M. Sahu and R. Dash, "Study on Face Recognition Techniques," Proc. of the 2020 IEEE Int. Conf. on Communication and Signal Processing (ICCSP 2020), pp. 613–616, Chennai, India, Jul. 2020.

[5] A. A. Sukmandhani and I. Sutedja, "Face Recognition Method for Online Exams," Proc. of the IEEE International Conference on Information Management and Technology (ICIMTech 2019), pp. 175–179, DOI: 10.1109/ICIMTECH.2019.8843831, Jakarta/Bali, Indonesia, Aug. 2019.

[6] C. S. Hsiao, C. P. Fan and Y. T. Hwang, "Design and Analysis of Deep-learning Based Iris Recognition Technologies by Combination of U-Net and EfficientNet," Proc. of the 9th IEEE Int. Conf. on Information and Education Technology (ICIET 2021), pp. 433–437, Okayama, Japan, Mar. 2021.

[7] H. D. Rafik and M. Boubaker, "A Multi Biometric System Based on the Right Iris and the Left Iris Using the Combination of Convolutional Neural Networks," Proc. of the 4th IEEE Int. Conf. on Intelligent Computing in Data Sciences (ICDS 2020), DOI: 10.1109/ICDS50568.2020.9268737, Fez, Morocco, Oct. 2020.

[8] S. D. Shirke and C. Rajabhushnam, "Biometric Personal Iris Recognition from an Image at Long Distance," Proceedings of the International Conference on Trends in Electronics and Informatics (ICOEI 2019), vol. 2019-April, pp. 560–565, DOI: 10.1109/ICOEI.2019.8862640, Apr. 2019.

[9] R. Giorgi, N. Bettin, S. Ermini, F. Montefoschi and A. Rizzo, "An Iris+Voice Recognition System for a Smart Doorbell," Proc. of the 8th IEEE Mediterranean Conference on Embedded Computing (MECO 2019), DOI: 10.1109/MECO.2019.8760187, Budva, Montenegro, Jun. 2019.

[10] O. Tymchenko, B. Havrysh, O. O. Tymchenko, O. Khamula, B. Kovalskyi and K. Havrysh, "Person

Voice Recognition Methods," Proc. of the IEEE 3rd Int. Conf. on Data Stream Mining and Processing (DSMP 2020), pp. 287–290, Aug. 2020.

[11] E. M. Owaidah, K. S. Aloufi and J. H. Alkhatib, "Gait Recognition for Saudi Costume Using Kinect Skeletal Tracking," Proc. of the 2nd Int. Conf. on Computer Applications and Information Security (ICCAIS 2019), DOI: 10.1109/CAIS.2019.8769552, Riyadh, Saudi Arabia, May 2019.

[12] H. M. L. Aung and C. Pluempitiwiriyawej, "Gait Biometric-based Human Recognition System Using Deep Convolutional Neural Network in Surveillance System," Peoc. Of IEEE Asia Conference on Computers and Communications (ACCC 2020), pp. 47–51, DOI: 10.1109/ACCC51160.2020.9347899, Singapore, Sep. 2020.

[13] R. Srivastva, A. Singh and Y. N. Singh, "PlexNet: A Fast and Robust ECG Biometric System for Human Recognition," Information Sciences, vol. 558, pp. 208–228, DOI: 10.1016/J.INS.2021.01.001, May 2021.

[14] M. Wang, K. Kasmarik, A. Bezerianos, K. C. Tan and H. Abbass, "On the Channel Density of EEG Signals for Reliable Biometric Recognition," Pattern Recognition Letters, vol. 147, pp. 134–141, DOI: 10.1016/J.PATREC.2021.04.003, Jul. 2021.

[15] W. Cui, Z. Wang and Y. Li, "ECG-based Biometric Recognition under Exercise and Rest Situations," Biomedical Engineering Advances, p. 100008, DOI: 10.1016/J.BEA.2021.100008, Jul. 2021.

[16] Z. Wang, J. Yang and Y. Zhu, "Review of Ear Biometrics," Archives of Computational Methods in Engineering, vol. 28, no. 1, pp. 149–180, DOI: 10.1007/S11831-019-09376-2, Nov. 2019.

[17] A. Abaza, A. Ross, C. Hebert et al., "A Survey on Ear Biometrics," ACM Computing Surveys (CSUR), vol. 45, no. 2, pp. 1-35, DOI: 10.1145/2431211.2431221, Mar. 2013.

[18] Ž. Emeršič, V. Štruc and P. Peer, "Ear Recognition: More than a Survey," Neurocomputing, vol. 255, pp. 26–39, DOI: 10.1016/J.NEUCOM.2016.08.139, Sep. 2017.

[19] L. P. Etter, E. J. Ragan, R. Campion, D. Martinez and C. J. Gill, "Ear Biometrics for Patient Identification in Global Health: A Field Study to Test the Effectiveness of an Image Stabilization Device in Improving Identification Accuracy," BMC Medical Informatics and Decision Making, vol. 19, no. 1, pp. 1–9, DOI: 10.1186/S12911-019-0833-9, Jun. 2019.

[20] B. Bhanu, "Ear Shape for Biometric Identification," Encyclopedia of Cryptography and Security, pp. 372–378, DOI: 10.1007/978-1-4419-5906-5_738, 2011.

[21] A. Kamboj, R. Rani and A. Nigam, "A Comprehensive Survey and Deep Learning-based Approach for Human Recognition Using Ear Biometric," The Visual Computer, vol. 2021, pp. 1–34, DOI: 10.1007/S00371-021-02119-0, 2021.

[22] S. Ntshangase and D. Mathekga, "Ear Recognition for Young Children," Proc. of the IEEE International Multidisciplinary Information Technology and Engineering Conference (IMITEC 2019), DOI: 10.1109/IMITEC45504.2019.9015852, Vanderbijlpark, South Africa, Nov. 2019.

[23] P. Kavipriya, M. R. Ebenezar Jebarani, T. Vino and G. Jegan, "Ear Biometric for Personal Identification Using Canny Edge Detection Algorithm and Contour Tracking Method," Materials Today: Proceedings, DOI: 10.1016/J.MATPR.2021.03.351, Apr. 2021.

[24] M. Cheribet and S. Mazouzi, "A New Adapted Canny Filter for Edge Detection in Range Images," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 7, no. 3, pp. 278-291, DOI: 10.5455/JJCIT.71-1620428305, Sep. 2021.

[25] N. Mangayarkarasi, G. Raghuraman and A. Nasreen, "Contour Detection Based Ear Recognition for Biometric Applications," Procedia Computer Science, vol. 165, pp. 751–758, DOI: 10.1016/J.PROCS.2020.01.016, Jan. 2019.

[26] S. M. Jiddah and K. Yurtkan, "Fusion of Geometric and Texture Features for Ear Recognition," Proc. of the 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT 2018), DOI: 10.1109/ISMSIT.2018.8567044, Ankara, Turkey, Dec. 2018.

[27] M. Zarachoff, A. Sheikh-Akbari and D. Monekosso, "Single Image Ear Recognition Using Wavelet-based Multi-band PCA," Proc. of the 27th IEEE European Signal Processing Conference (EUSIPCO 2019), vol. 2019-September, DOI: 10.23919/EUSIPCO.2019.8903090, A Coruna, Spain, Sep. 2019.

[28] S. Sajadi and A. Fathi, "Genetic Algorithm Based Local and Global Spectral Features Extraction for Ear Recognition," Expert Systems with Applications, vol. 159, p. 113639, DOI: 10.1016/J.ESWA.2020.113639, Nov. 2020.

[29] S. F. A. Abuowaida and H. Y. Chan, "Improved Deep Learning Architecture for Depth Estimation from Single Image," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 4, pp. 434–445, DOI: 10.5455/JJCIT.71-1593368945, Dec. 2020.

[30] Y. Khaldi, A. Benzaoui, A. Ouahabi, S. Jacques and A. Taleb-Ahmed, "Ear Recognition Based on Deep Unsupervised Active Learning," IEEE Sensors Journal, Early Access, vol. 2021, DOI: 10.1109/JSEN.2021.3100151, 2021.

[31] Y. Lei, B. Du, J. Qian and Z. Feng, "Research on Ear Recognition Based on SSD-MobileNet-v1 Network," Proceedings of the Chinese Automation Congress, (CAC 2020), pp. 4371–4376, DOI: 10.1109/CAC51589.2020.9326541, Nov. 2020.

[32] T. Ying, W. Shining and L. Wanxiang, "Human Ear Recognition Based on Deep Convolutional Neural Network," Proc. of the 30th Chinese Control and Decision Conference (CCDC 2018), pp. 1830–1835, DOI: 10.1109/CCDC.2018.8407424, Jul. 2018.

[33] M. Chowdhury, R. Islam and J. Gao, "Robust Ear Biometric Recognition Using Neural Network," Proc. of the 12th IEEE Conference on Industrial Electronics and Applications (ICIEA 2017), vol. 2018-February, pp. 1855–1859, DOI: 10.1109/ICIEA.2017.8283140, Feb. 2018.

[34] H. Alshazly, C. Linse, E. Barth and T. Martinetz, "Deep Convolutional Neural Networks for Unconstrained Ear Recognition," IEEE Access, vol. 8, pp. 170295–170310, DOI: 10.1109/ACCESS.2020.3024116, 2020.

[35] M. Alejo and C. P. G. Hate, "Unconstrained Ear Recognition through Domain Adaptive Deep Learning Models of Convolutional Neural Network," International Journal of Recent Technology and Engineering, vol. 8, no. 2, DOI: 10.35940/ijrte.B2865.078219, 2019.

[36] A. A. Almisreb, N. Jamil and N. M. Din, "Utilizing AlexNet Deep Transfer Learning for Ear Recognition," Proc. of the 4th International Conference on Information Retrieval and Knowledge Management: Diving into Data Sciences (CAMP 2018), pp. 8–12, DOI: 10.1109/INFRKM.2018.8464769, 2018.

[37] Y. Zhong and W. Deng, "Face Transformer for Recognition," arXiv, arXiv:2103.14803v2, [Online], Available: https://arxiv.org/abs/2103.14803v2, Mar. 2021.

[38] A. George and S. Marcel, "On the Effectiveness of Vision Transformers for Zero-shot Face Anti-Spoofing," Proc. of IEEE International Joint Conference on Biometrics (IJCB), pp. 1–8, DOI: 10.1109/IJCB52358.2021.9484333, Shenzhen, China, 2021.

[39] A. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems, vol. 2017-December, pp. 5999–6009, [Online], Available: https://arxiv.org/abs/1706.03762v5, 2017.

[40] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan and M. Shah, "Transformers in Vision: A Survey," arXiv, [Online], Available: http://arxiv.org/abs/2101.01169, 2021.

[41] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov and S. Zagoruyko, "End-to-End Object Detection with Transformers," Proc. of the European Conference on Computer Vision, Part of the Lecture Notes in Computer Science Book Series, vol. 12346, pp. 213–229, [Online], Available: https://arxiv.org/abs/2005.12872v3, 2021.

[42] X. Zhu, W. Su, L. Lu, B. Li, X. Wang and J. Dai, "Deformable DETR: Deformable Transformers for End-to-End Object Detection," arXiv, [Online]. Available: https://arxiv.org/abs/2010.04159v4, Oct. 2020, Accessed: Aug. 02, 2021.

[43] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille and L.-C. Chen, "Axial-DeepLab: Stand-alone Axial-attention for Panoptic Segmentation," Proc. of the European Conference on Computer Vision, Part of the Lecture Notes in Computer Science Book Series, vol. 12349, pp. 108–126, [Online], Available: https://arxiv.org/abs/2003.07853v2, 2021.

[44] L. Ye, M. Rochan, Z. Liu and Y. Wang, "Cross-modal Self-attention Network for Referring Image Segmentation," Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR), vol. 2019, pp. 10494–10503, DOI**:** 10.1109/CVPR.2019.01075, CA, USA, 2019.

[45] N. Parmar et al., "Image Transformer," Proc. of the 35th Int. Conf. on Machine Learning (ICML 2018), vol. 9, pp. 6453–6462, [Online], Available: https://arxiv.org/abs/1802.05751v3, Feb. 2018.

[46] M. Chen et al., "Generative Pretraining from Pixels," Proc. of the 37th International Conference on Machine Learning, pp. 1691–1703. [Online]. Available: http://proceedings.mlr.press/v119/chen20s. html, Nov. 2020.

[47] P. Esser, R. Rombach and B. Ommer, "Taming Transformers for High-resolution Image Synthesis," arXiv, [Online], Available: https://arxiv.org/abs/2012.09841v3, Dec. 2020.

[48] Y. Jiang, S. Chang and Z. Wang, "TransGAN: Two Pure Transformers Can Make One Strong GAN and That Can Scale Up," arXiv, [Online], Available: http://arxiv.org/abs/2102.07074, Feb. 2021.

[49] X. Wang, C. Yeshwanth and M. Nießner, "SceneFormer: Indoor Scene Generation with Transformers," arXiv, [Online], Available: https://arxiv.org/abs/2012.09793, Dec. 2020.

[50] A. Radford et al., "Learning Transferable Visual Models from Natural Language Supervision," arXiv, [Online], Available: http://arxiv.org/abs/2103.00020, Feb. 2021.

[51] A. Dosovitskiy et al., "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale," arXiv, [Online], Available: http://arxiv.org/abs/2010.11929, Oct. 2020.

[52] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles and H. Jégou, "Training Data-efficient Image Transformers & Amp; Distillation through Attention," arXiv, [Online], Available: https://arxiv.org/abs/2012.12877, Dec. 2020.

[53] A. Bakhtiarnia, Q. Zhang and A. Iosifidis, "Single-layer Vision Transformers for More Accurate Early Exits with Less Overhead," arXiv, [Online], Available: https://arxiv.org/abs/2105.09121v1, May 2021.

[54] M. Naseer, K. Ranasinghe, S. Khan, M. Hayat, F. S. Khan and M.-H. Yang, "Intriguing Properties of Vision Transformers," arXiv, [Online], Available: https://arxiv.org/abs/2105.10497v2, May 2021.

[55] V. T. Hoang, "EarVN1.0: A New Large-scale Ear Images Dataset in the Wild," Data in Brief, vol. 27, p. 104630, DOI: 10.1016/J.DIB.2019.104630, Dec. 2019.

[56] Ž. Emeršič et al., "The Unconstrained Ear Recognition Challenge 2019 - ArXiv Version with Appendix," arXiv, [Online], Available: https://arxiv.org/abs/1903.04143v3, Mar. 2019.

[57] K. Weiss, T. M. Khoshgoftaar and D. D. Wang, "A Survey of Transfer Learning, " Journal of Big Data, vol. 3, no. 1, pp. 1–40, DOI: 10.1186/s40537-016-0043-6., Dec. 2016.

[58] B. Zoph, V. Vasudevan, J. Shlens and Q. V. Le, "Learning Transferable Architectures for Scalable Image Recognition, "Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 8697–8710, DOI: 10.1109/CVPR.2018.00907, 2018.

[59] X. Ying, "An Overview of Overfitting and Its Solutions," Journal of Physics: Conference Series, vol. 1168, no. 2, DOI: 10.1088/1742-6596/1168/2/022022, Mar. 2019.

[60] X. Chen, C.-J. Hsieh and B. Gong, "When Vision Transformers Outperform ResNets without Pretraining or Strong Data Augmentations," arXiv, arXiv: 2106.01548, [Online], Available: https://arxiv.org/abs/2106.01548, Jun. 2021.

ملخص البحث:

لقد وفّرت أفضلية الأُذُن في تمييز الأشخاص مقارنةً بغيرها من وسائل التّمييز مجالاً خصْباً للباحثين لإجراء الدّراسات على طُرُق الحوسبة. تقدّم هذه الورقة طريقةً قائمةً على التّعلُّم العميق للتّمييز غير المقيَّد للأشخاص عن طريق الأُذُن باستخدام شبكة المحوّلات العصبيّة: محوّل الصُّور (ViT) ومحوّلات الصُّور فعّالة البيانات (DeiTs).

النّماذج المستخدمة في هذه الدّراسة لتمييز الأشخاص عن طريق صُوَر الأُذُن حقّقت دقّة تمييزٍ مماثلةً أو أفضل من الطّرق المستخدَمة في هذا المجال والتي تستند على الشّبكات العصبيّة الالتفافيّة (CNNs) وغيرها من خوارزميات التّعلُّم العميق. كذلك بينت النّماذج المستخدمة في هذه الدّراسة أنها تعمل بصورة أفضل من حيث الأداء مقارنة بشبكات (ResNets) دون استخدام عمليات زيادة استنزافيّة للبيانات. علاوة على ذلك، لاحظت هذه الدّراسة أنّ أداء تقنية (ViT) المستخدمة في هذه الدّراسة كان مُقارباً لأداء مثيلاتها من التّقنيات المستخدمة في الدّراسات البيومتريّة الأخرى.

TWO-WAY METRIC LEARNING WITH MAJORITY AND MINORITY SUBSETS FOR CLASSIFICATION OF LARGE EXTREMELY IMBALANCED FACE DATASET

Ashu Kaushik[1] and Seba Susan[2]

(Received: 16-Jul.-2021, Revised: 12-Sep.-2021, Accepted: 27-Sep.-2021)

ABSTRACT

This paper proposes a new learning methodology involving deep features and two-way metric learning for large, extremely imbalanced face datasets where the number of minority classes and the imbalance ratio are both very high. The problem arises because the faces of some celebrities, being more popular, are readily available in social media and the internet, while the faces of some relatively lesser-known personalities are fewer in number. Resampling being impractical in this scenario, we propose metric learning as the tool for mitigating the class-imbalance problem prior to the classification stage. To reduce the computational overhead associated with metric learning, we separately conduct weakly supervized metric learning with majority and minority class subsets, a process that we call two-way metric learning. Transformation matrices learnt from the majority and minority subsets are used to transform the entire input space twice. The test sample in the transformed space is assigned the class of its nearest neighbor in the training set of the twice-transformed input space. Deep features derived from the state-of-the-art pre-trained deep network VGG-Face form the input space and the aggregate cosine similarity measure is used to find the closest neighbor in the training set of the twice-transformed input space. Experiments on the benchmark LFW face database having 1680 classes of celebrity faces prove that the proposed methodology is more effective than existing methods for the classification of large, extremely imbalanced face datasets. The classification accuracies of the minority classes are especially found to be boosted which is a rare accomplishment among existing methods for imbalanced learning in deep frameworks.

KEYWORDS

Face recognition, Metric learning, VGG-Face, Deep learning, Imbalanced learning, Extremely imbalanced dataset.

1. INTRODUCTION

In this computer and mobile frenzy era, almost everything is getting digitized. The large amount of data that is being generated by the use of such digital devices is creating a havoc and needs to be analyzed, sorted and stored properly and judiciously. Social media platforms, like Facebook, Instagram, Twitter and Zoom, create a large amount of data and logs based upon the usage of the account holder. Face recognition plays a crucial role in detecting and tagging the identity of a person in social media. The users who are very active on social media have a large amount of data associated with them, including images that reveal the identity of the individual. Such users constitute the majority class in the learning framework. On the other hand, users who are less active contribute to lesser data and fewer images that make automated face recognition a difficult task; such users constitute the minority class. Face recognition from such imbalanced datasets, where the difference between the volume of data between the majority and minority classes is very high, is indeed a difficult task [1]-[2]. Learning from imbalanced data is a well-researched problem in data mining [3]-[4] with various solutions proposed ranging from resampling [5] and metric learning [6] to cost-sensitive learning [7]. Selective pruning of majority and minority samples is found helpful, especially when some amount of overlapping is there between the majority and minority classes [8]. Most of the proposed solutions are effective for the binary classification problem, but classification in the multi-class scenario with a highly imbalanced class distribution is still an open research problem [9]. Resampling techniques might work on small toy datasets popular in data mining, but while dealing with a very large dataset comprising of faces derived from social media, consisting of more than a

[1] A. Kaushik is a Postgraduate Student in the Department of Information Technology, Delhi Technological University, Delhi, India. Email: ashukaushik8395@gmail.com

[2] S. Susan is a Professor in the Department of Information Technology, Delhi Technological University, Delhi, India. Email: seba_406@yahoo.in, ORCID: 0000-0002-6709-6591.

thousand classes and with a very high class-imbalance ratio (ratio of majority to minority population), it is not considered a feasible solution [10].

Novel learning methodologies need to be devised for extremely imbalanced large face databases in order to meet the computational overhead and at the same time improve the classification accuracy, especially for the minority classes having inadequate number of samples to learn from. This is the problem tackled in this paper and we choose metric learning [11] as the tool for transforming the entire input space in order to reduce intra-class differences and increase the inter-class differences. This is achieved by identifying two smaller subsets of the large imbalanced face dataset as the majority and minority classes and performing metric learning using these two subsets. Metric learning would be done in a weakly supervized fashion for both majority and minority subsets to learn the distance metric which can be used to transform the entire input space prior to the classification stage. The contributions made by this paper can be summarized as follows:

1. Metric learning with deep features is introduced as a viable tool for large extremely imbalanced facial datasets having more than a thousand minority classes, for which resampling is not feasible.

2. To reduce the computational overhead associated with metric learning, a weakly supervized learning scheme is devised, for which smaller-sized majority and minority class subsets are identified.

3. The entire input space is then transformed twice, once using the transformation matrix learnt from the majority class subset and likewise from the minority class subset.

4. The aggregate cosine similarity measure is eventually used for the classification of the transformed test sample by finding its closest neighbor in the training set of the twice-transformed input space.

5. Experiments on the large, extremely imbalanced LFW face database having 1680 classes, with large disparity in class populations, yield effective classification, especially for the minority classes, a rare accomplishment among existing methods for imbalanced learning in deep frameworks. The methods proposed so far mostly concentrate on the performance of majority classes only and exclude the minority classes in the learning process.

The further sections of the paper are organized as follows. Section 2 describes the motivation for our work and the proposed methodology. Section 3 analyzes the results of the experimentation and Section 4 outlines the conclusions and the future work.

2. Proposed Method

2.1 Motivation and Brief Background

Deep neural networks have been used to classify large image datasets, such as ImageNet, and have achieved excellent results [12]. However, they are computationally costly; it would take a long time to setup and train the network for accurate predictions. Pre-trained deep networks trained on large databases and fine-tuned on smaller datasets have been used successfully for complex computer vision tasks, such as face recognition and age estimation [13]. Deep neural networks have, by themselves, some inherent property of improving the scores of minority classes [14]. Pruning of the insignificant features while passing them into the deep networks is a solution to ease out on the computation part [15]. Resampling strategies prevalent in data mining are infeasible for very large, extremely imbalanced image datasets due to the high computational complexity, as is the case in our current work. We therefore, propose to use metric learning using sparse samples for transforming the input space of the large, extremely imbalanced face dataset. Our work is motivated by prior works [2], [16]-[17], [18] that have applied metric learning to mitigate class imbalance for toy datasets. Application of metric learning for large imbalanced datasets, however, requires a lot of computations and very few works have addressed the problem. In our earlier work, which is a precursor of the current work [2], we identified a majority subset of top-186 classes and learned the distance metric using a few samples of each class of the majority subset. The result was an improvement in the performance of majority classes. However, the improvement in the performance of the minority classes was only marginal. In the current work, emphasis is on improving the performance of minority classes by devising a metric learning scheme that would concentrate on the minority classes as well. Some of the earlier techniques propose oversampling of the minority class for improving the performance [19]-[20]. However, this solution is impractical in our case due to the presence of more than a thousand classes and the large size of the dataset. The process pipeline for our method is described in the following sub-sections.

2.2 Deep Feature Extraction from VGG-Face Deep Pre-trained Network

VGG-Face [22], FaceNet [24], DeepFace [25] and OpenFace [39] are a few state-of-the-art deep pre-trained networks customized for face recognition. The Convolutional Neural Network (CNN) [26] is the core neural network of all these models. Because of the large number of hidden layers in their architecture, they are referred to as deep networks. Parkhi *et al.* introduced the pre-trained network VGG-Face in 2014 [22]. It is based on the VGG-16 [27] architecture, which consists of 16 convolutional layers followed by a series of pooling and activation layers. The pre-trained network originally trained on two million images is used to generate a 2622x1 feature embedding for each image in our dataset, as shown in the VGG-Face model process flow recreated in Figure 1. The 2622-dimensional feature vectors are further used, in our work, for metric learning and subsequent classification by a suitable classifier.

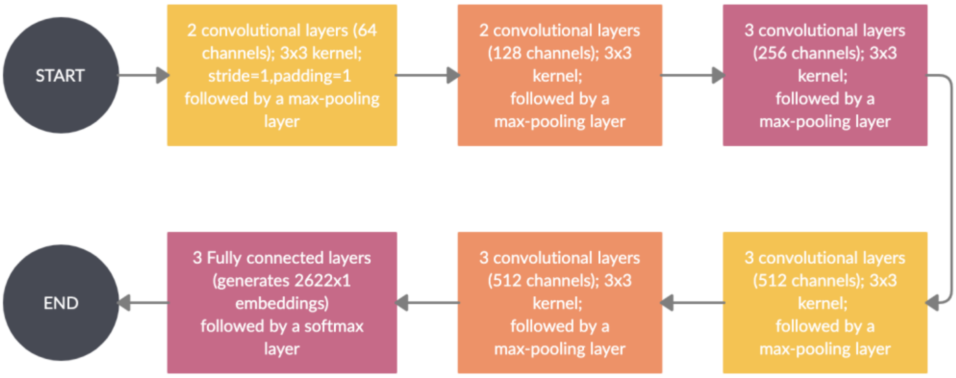

START
2 convolutional layers (64 channels); 3x3 kernel; stride=1,padding=1 followed by a max-pooling layer
2 convolutional layers (128 channels); 3x3 kernel; followed by a max-pooling layer
3 convolutional layers (256 channels); 3x3 kernel; followed by a max-pooling layer
3 convolutional layers (512 channels); 3x3 kernel; followed by a max-pooling layer
3 convolutional layers (512 channels); 3x3 kernel; followed by a max-pooling layer
3 Fully connected layers (generates 2622x1 embeddings) followed by a softmax layer
END

Figure 1. VGG-Face model.

2.3 Metric Learning for Transformation of the Input Space

Various distance metric learning schemes have been proposed, in the past that improved the classification performance of imbalanced datasets. The aim is to transform the input space so as to bring the samples of a class closer and push samples from different classes farther apart. Some of the most prominent distance metric learning algorithms are based on the Mahalanobis distance that is shown in Equation (1) for two feature vectors (**x, y**).

$$\boldsymbol{dm}(\boldsymbol{x},\boldsymbol{y}) = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^{\boldsymbol{T}}\,\boldsymbol{M}(\boldsymbol{x}-\boldsymbol{y})} \tag{1}$$

Here, ***M*** represents the positive semidefinite matrix that is to be estimated. It is similar to the Euclidean distance in a different space or a linear projection of the distance between two points. One of the most popular algorithms using the Mahalanobis distance metric is Large Margin Nearest Neighbor (LMNN). Weinberger *et al.* developed LMNN in 2009 [23] and it has since become one of the most widely used data space modification algorithms. It is a supervized metric learning algorithm that may be used before the classification stage. Optimization problem involved is convex and simple to solve. The cost function shown in Equation (2) is the one that must be minimized.

$$\boldsymbol{costf} = (\delta)\boldsymbol{fu}_{push}(\boldsymbol{L}) + (1-\delta)\boldsymbol{fu}_{pull}(\boldsymbol{L}) \tag{2}$$

The loss function has two terms: one relates to the force that pulls samples from the same class closer together, while the other refers to the force that pushes samples from other classes apart. The cost function in (2) is a weighted sum of push and pull functions. The value of δ lies between 0 and 1. This function's transformation matrix limits the margin between k-similar samples to a minimum and maximizes the margin between samples of different classes. When the number of classes is considerable, direct application of metric learning is not recommended due to the computational complexity involved.

LMNN is based on the principle of bringing the samples belonging to the same class closer and samples belonging to different classes are moved further apart, as illustrated in Figure 2. LMNN thus brings about a global linear transformation of the input space that improves the classification of distance-based classifiers, such as *k*NN. We use the cosine similarity measure in the classification stage that follows the metric learning phase in the process pipeline. The cosine similarity measure between two feature vectors (**x, y**) is shown in Equation (3).

$$\boldsymbol{sim}(\boldsymbol{x},\boldsymbol{y}) = \frac{\sum_i \boldsymbol{x}_i \times \boldsymbol{y}_i}{\sqrt{\sum_i \boldsymbol{x}_i^{\,2}} \times \sqrt{\sum_i \boldsymbol{y}_i^{\,2}}} \tag{3}$$

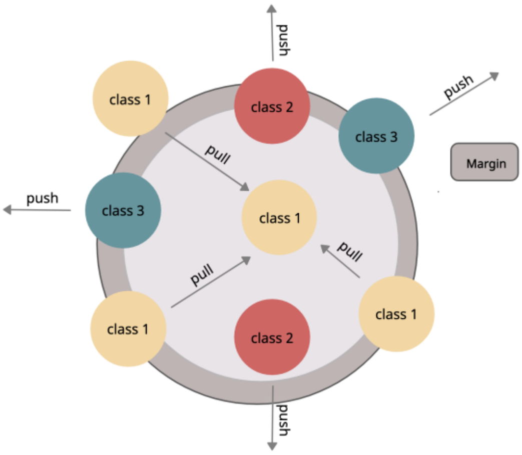


Figure 2. Diagrammatic representation of how LMNN attempts to bring similar & nearest k samples closer and moves dissimilar samples further apart (k=3, as in original paper [23]).

Other examples of metric learning are Neighborhood Component Analysis (NCA) [28], Metric Learning for Kernel Regression (MLKR) [29], Information Theoretic Metric Learning (ITML) [30], Least Squares Metric Learning (LSML) [31] and Sparse Discriminant Metric Learning (SDML) [32]. It was proved in a recent work that LMNN, NCA and MLKR yield the best performance for the *k*NN classification of toy datasets having high imbalance ratio.

2.4 Methodology

The basic outline of the methodology used in our experiments is described next. We segregate the majority and minority classes based upon the number of samples that each class contains. Figure 3 shows the class populations of the Labeled Faces in the Wild (LFW) face dataset [21] used in our experiments, that range from 530 to 2. The graph shows an extremely uneven population distribution.

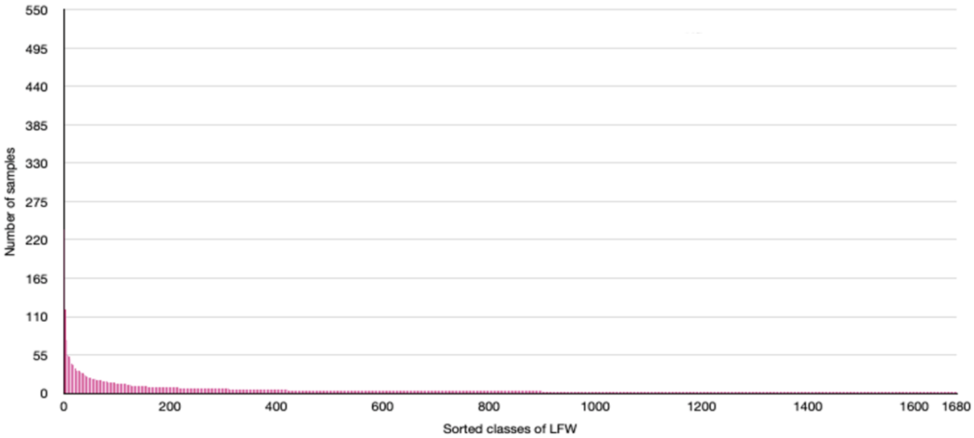
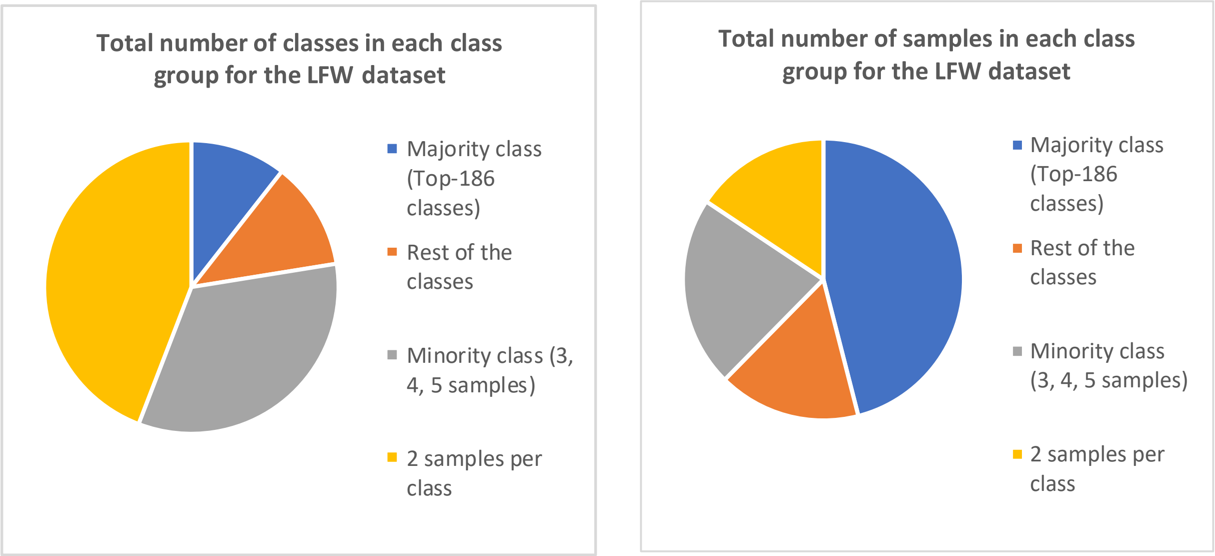


Figure 3. Sorted class populations of the LFW dataset containing 1680 classes of celebrity faces.

As the number of minority classes is very high as compared to the majority classes in the extremely imbalanced LFW dataset, we select a majority class subset and a minority class subset to perform metric learning twice, one from the perspective of the majority class and the other from the perspective of the minority class. This is the primary contribution of our work. We divided the classes based on the class populations, as shown in Figure 4. We define the group of top-186 classes as the majority class and the group of classes with 3, 4 or 5 samples per class as the minority class. To derive the top-186 classes, as per the procedure in our previous work [2], the class populations are sorted in the decreasing order of their populations and a sum-based partitioning of the sorted class populations yields the threshold as 186 as the lower boundary of the majority class in the LFW dataset. The class-wise and sample-wise groupings are shown in the pie charts in Figure 4 (a) and Figure 4 (b), respectively. It is noted from the pie chart in Figure 4 (a) that the number of minority classes having less than or equal to 5 samples is more than 1300 out of the total available 1680 classes.



(a) (b)

Figure 4. Grouping of the 1680 classes of LFW dataset into majority and minority classes (a) class-wise distribution (b) sample-wise distribution.

Due to the computational complexity involved in metric learning, we have excluded the rest of the classes having samples in the range of 6 to 8 and those having 2 samples from the metric learning computations, since the number of such classes is large and inclusion of these two groups would render metric learning computationally infeasible and impractical. The input space transformation after learning the distance metric is, however, applied to the entire training space. Also, only a few samples from each class are taken into consideration while performing metric learning to reduce the computational expense. Our learning framework is thus an instance of weakly supervized learning. The block diagram for our learning model is shown in Figure 5.

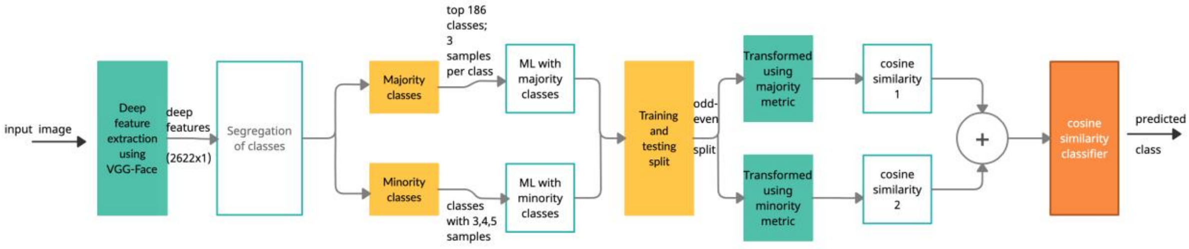

input image
Deep feature extraction using VGG-Face
deep features (2622x1)
Segregation of classes
Majority classes
top 186 classes; 3 samples per class
ML with majority classes
Minority classes
classes with 3,4,5 samples
ML with minority classes
Training and testing split
odd-even split
Transformed using majority metric
Transformed using minority metric
cosine similarity 1
cosine similarity 2
+
cosine similarity classifier
predicted class

Figure 5. Proposed model.

A deep neural network VGG-Face [22] that is pre-trained on two-million facial images is used to generate the feature embeddings for the LFW face dataset. The gray input images of dimension 64x64 were given as input to the VGG-Face model and vector embeddings of dimension 2622x1 were extracted as per the guidelines in the original paper of VGG-Face [22]. We perform metric learning for both majority class and minority class subsets and get two different transformation metrics. We considered only 3 samples per class from both subsets to reduce the computational cost, since a large number of inter- and intra-class distances need to be calculated. The entire input space is divided into two parts; i.e., training and testing based on alternate sampling and after that, the entire training subset is transformed using both the minority class metric and majority class metric. We use Large Margin Nearest Neighbor (LMNN) as the metric learning technique and it is based on the nearest neighbor rule. The number of nearest neighbors is fixed as k=3, which is the same as given in the original paper of LMNN proposed by Weinberger *et al*. in 2009 [23]. For the minority classes having fewer than 3 samples in the training set, all the samples in the training set are included. The number of neighbors is to be kept small, since a large number of distances, both inter- and intra-class, need to be calculated, which is computationally expensive. The final step would be the classification stage in which the class label of the test sample has to be determined. For each test sample, we transform it using both the distance metrics and in each case, compute the cosine similarity with each sample in the training set of the transformed input space. The two cosine similarity vectors are summed up and the training sample corresponding to the maximum aggregate cosine similarity is selected as the closest neighbor in the training space; its class label is assigned to the test sample.

3. Results

The experiments were performed on the publicly available dataset Labeled Faces in the Wild (LFW) developed in 2007 by Huang *et al.* [21]. It is today a benchmark in the field of facial recognition that is used for training several state-of-the-art pre-trained networks for face recognition. It is a highly imbalanced dataset consisting of 1680 celebrity classes with George W. Bush having the maximum number of samples (=530) and Michel Duclos having the minimum number of samples (=2). Only those celebs were selected who have two or more than two samples and the celebs with only one sample were discarded from our experiments. Out of the 1680 selected celebs, 1369 celebs have <=5 samples, as verified from the pie charts in Figure 4, which proves that the minority classes outnumber the majority classes in the LFW dataset.

We extracted the deep features using the pre-trained VGG-Face model, as discussed in Section 2. The gray-scale images were resized to dimensions 64x64. The pre-trained model generated the 2622-dimensional feature embeddings which were further fed to the learning module. The dataset was divided into majority and minority class subsets as explained in Section 2 and two-way metric learning was performed using these two subsets. The transformation matrices generated were used to transform the entire input space twice, separately. The cosine similarity measure was used to find the closeness of the test sample to a training sample in both the transformed spaces; this was followed by a simple summation of the cosine similarity measures.

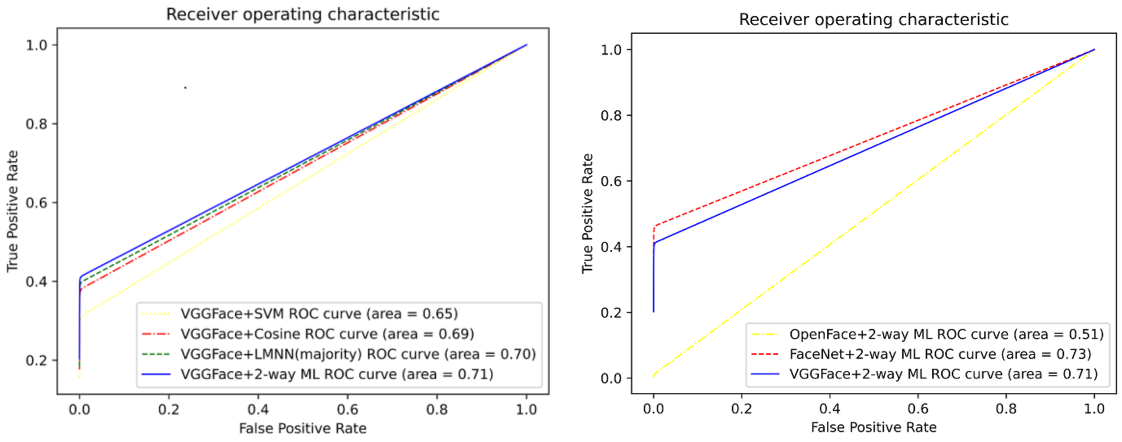
The dataset was divided into training and testing sets by alternate sampling. In case of odd number of samples n, the training set contained (n+1)/2 samples and the test set contained (n-1)/2 samples. Cross-validation is done by swapping the training and test sets. The results - Accuracy, F1-score and AUC scores obtained from ROC curves, are compiled in Table 1 for both Validation (V) and Cross-Validation (CV). We compared the performance of our method with that of existing methods: HOG + SVM [33], HOG+ Cosine similarity [34], HOG + Metric learning with majority class [2], VGG-Face + SVM [37], VGG-Face + Cosine similarity [38] and VGG-Face + Metric learning with majority class [36]. The proposed method outperformed all existing methods in terms of accuracy, F1-score and AUC scores as observed from Table 1. The scores are overall on the lower side due to the inclusion of the entire set of 1680 classes including the 779 minority classes with only 2 samples per class of which one sample is used for training and one sample for testing. Most of the earlier experiments on LFW dataset report results only for the majority classes that have at least 10 samples following the face verification protocol in the LFW technical report in [21]. The minority classes are excluded from previous works, since they contribute to class imbalance and deteriorate the overall performance. The state-of-the-art deep networks need a large number of training samples per class for efficient classification [35].

We also compared the current results with that of the weakly supervized metric learning scheme using majority classes [36] by substituting the traditional HOG features in [2] with VGG-Face deep features. We have used an Intel i5 dual core processor clocked at 2.7 GHz and python 3.7 software platform to perform the experiments. The system took half an hour each to learn the distance metrics for both majority and minority subsets. The classification took only a few minutes to execute. The deep features were found to outperform the HOG features, as observed from the classification scores in Table 1. The corresponding ROC curves are shown in Figure 6. A comparison of models that classify

VGG-Face deep features from the ROC graph in Figure 6 (a) reveals that two-way metric learning prior to classification improves the performance scores. Figure 6 (b) compares our method on two other deep features other than VGG-Face; i.e., FaceNet [24] and OpenFace [39]. The FaceNet model performs better than VGG-Face for the proposed two-way metric learning scheme.

Table 1. Performance comparison of various methods on the LFW dataset.

Method	AUC		F1-score		Accuracy	
	V	CV	V	CV	V	CV
HOG + SVM [33]	0.528	0.523	0.059	0.053	28.7%	27.4%
HOG + Cosine Similarity [34]	0.544	0.541	0.078	0.076	21.5%	19.1%
HOG + LMNN metric learning with majority subset [2]	0.556	0.554	0.1006	0.097	26.8%	24.6%
VGG-Face + SVM [37]	0.654	0.635	0.287	0.263	55.4%	51.1%
VGG-Face + Cosine similarity [38]	0.689	0.675	0.342	0.329	55.4%	51.9%
VGG-Face + LMNN metric learning with majority subset [36]	0.697	0.681	0.355	0.339	57%	53.6%
VGG-Face + Two-way metric learning (proposed)	0.705	0.689	0.372	0.356	58.5%	54.8%

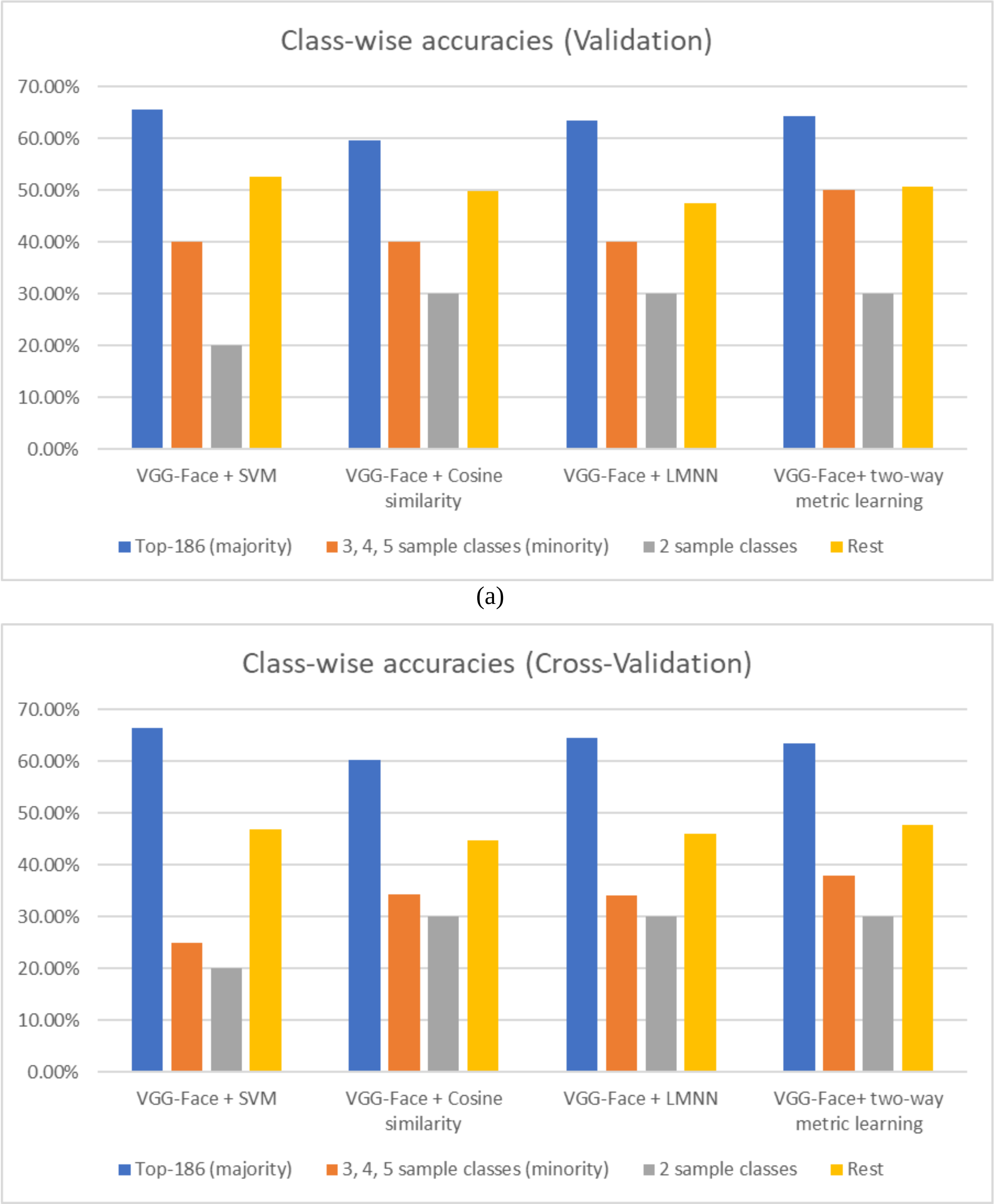


(a) (b)

Figure 6. ROC graphs for (a) classification based on VGG-Face deep features by various methods, (b) classification based on VGG-Face, FaceNet and OpenFace deep features for the proposed two-way metric learning scheme.

Some of the class-wise accuracies are shown in Figure 7 for the majority and minority classes to understand the impact of our two-way metric learning scheme as opposed to a scenario where there is no metric learning and the deep features extracted from VGG-Face are learned directly by the classification framework. The cosine similarity measure is the classifier. As observed, Figure 7 shows a consistent performance for all majority classes for the two-way metric learning scheme, which is at par with VGG-Face + SVM. However, for the minority classes with 3, 4, 5 samples per class, a significant improvement in accuracy was recorded, with the accuracies jumping from 0% to 50% and above, for most of the minority classes.

The 2-sample classes showed a higher performance than VGG-Face+SVM. The category of classes tagged as “Rest” contain about 6 to 8 samples each. The performance of this set of classes was found improved over VGG-Face+Cosine similarity and VGG-Face+LMNN, though the performance was marginally lower than that of VGG-Face+SVM.



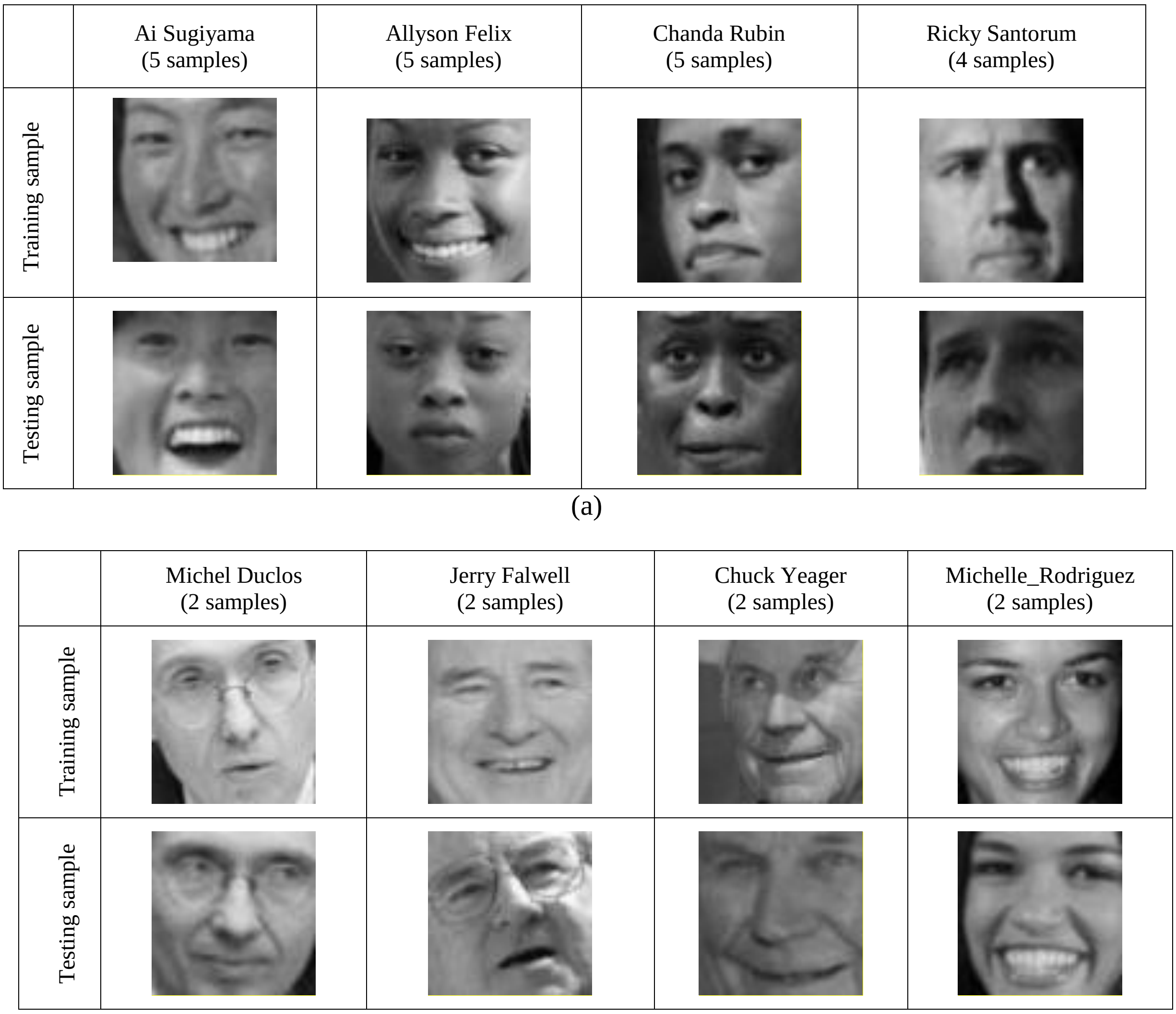
(a)

(b)

Figure 7. Performance comparison of majority class (top-186 classes), minority class (3, 4, 5 samples), 2-sample classes, rest of the classes (with samples in the range 6 to 8), with and without metric learning for (a) validation and (b) cross-validation experiments. (*LMNN was used as metric learning method).

Some success cases and failure cases for the proposed method are shown in Figure 8. The success cases shown are examples when metric learning proved to be useful for the classification. The failure cases are those, which were not classified by our method. Figure 9 shows the comparison between NCA, LMNN and MLKR metric learning schemes for the proposed methodology of two-way metric learning with deep features. The classification accuracies achieved for the top-10 majority classes are shown for all three metric learning schemes.

It is noted that LMNN significantly outperforms NCA and MLKR in terms of classification accuracies. LMNN involves minimization of the distance between each training sample and its k


Ai Sugiyama (5 samples)
Allyson Felix (5 samples)
Chanda Rubin (5 samples)
Ricky Santorum (4 samples)
Training sample
Testing sample
Michel Duclos (2 samples)
Jerry Falwell (2 samples)
Chuck Yeager (2 samples)
Michelle_Rodriguez (2 samples)
Training sample
Testing sample

(a)

(b)

Figure 8. (a) Some success cases of the proposed method, where VGG-Face features without metric learning could not classify the faces and (b) Some failure cases of the proposed method.

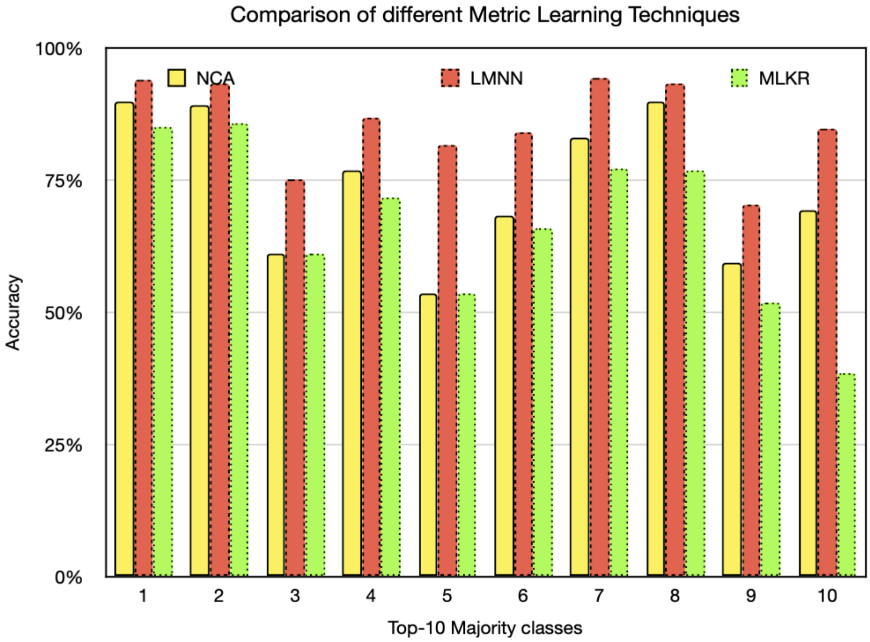


Figure 9. Performance comparison of top-10 majority classes of LFW for different metric learning techniques NCA, MLKR and LMNN.

nearest neighbors belonging to the same class while pushing the differently labelled samples farther apart. LMNN thus projects the input space into metric space in such a way that the inter-class similarities could be measured more accurately.

The primary contribution of our work as compared to our previous work and other works in literature is the improvement of accuracy of the minority classes. The challenge, here, was the existence of more than a thousand minority classes containing sparse samples, rendering metric learning a difficult task. On comparison with other methods, especially VGG-Face+SVM, we observe that though the accuracy of the majority class was comparable, the accuracy of the minority class significantly improved over all existing methods.

4. CONCLUSIONS

A novel learning methodology for large, extremely imbalanced face databases is proposed in this paper that involves deep features and two-way metric learning. LMNN is the metric learning scheme used. Deep features are extracted from the VGG-Face pre-trained model that is trained on two-million facial images. Majority and minority class subsets are identified based on the class population. Metric learning is applied twice, once for the majority subset and the second time for the minority subset. The closeness of the test sample from each training sample in the twice-transformed input space is measured using the sum of the cosine similarities computed in the two cases. The class of the closest training sample, in both the transformed spaces taken together, is assigned as the class of the test sample. Metric learning is known to transform the input space to bring samples of a class closer together. Two-way metric learning introduced in our scheme aims to improve the classification scores, especially for the minority classes, since it brings the few samples in the minority class closer together. Experiments were conducted on the LFW face dataset containing more than a thousand minority classes and the classification scores achieved indicate that the proposed learning technique is more effective than the existing methods for the classification of large, extremely imbalanced face datasets. The LFW dataset, to the best of our knowledge, is the largest extremely imbalanced dataset available for face recognition, which is the central theme of this paper. Our method can be easily adapted to other datasets having different scales of imbalance.

REFERENCES

[1] C. Huang, Y. Li, C. C. Loy and X. Tang, "Deep Imbalanced Learning for Face Recognition and Attribute Prediction," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 11, pp. 2781-2794, 2019.

[2] S. Susan and Ashu Kaushik, "Weakly Supervized Metric Learning with Majority Classes for Large Imbalanced Image Dataset," Proceedings of the 4th International Conference on Big Data and Internet of Things, pp. 16-19, DOI: 10.1145/3421537.3421549, 2020.

[3] H. He and E. A. Garcia, "Learning from Imbalanced Data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263-1284, 2009.

[4] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang and C. Jiang, "Random Forest for Credit Card Fraud Detection," Proc. of the 15th IEEE International Conference on Networking, Sensing and Control (ICNSC), pp. 1-6, Zhuhai, China, 2018.

[5] F. Zhang, G. Liu, Z. Li, C. Yan and C. Jiang, "GMM-based Undersampling and Its Application for Credit Card Fraud Detection," Proc. of the IEEE International Joint Conference on Neural Networks (IJCNN), pp. 1-8, Budapest, Hungary, 2019.

[6] S. Susan and A. Kumar, "DST-ML-EkNN: Data Space Transformation with Metric Learning and Elite K-nearest Neighbor Cluster Formation for Classification of Imbalanced Datasets," Proc. of Advances in Artificial Intelligence and Data Engineering, Part of the Advances in Intelligent Systems and Computing Book Series (AISC), vol. 1133, pp. 319-328, Springer, Singapore, 2021.

[7] H. Zhu, G. Liu, M. Zhou, Y. Xie, A. Abusorrah and Q. Kang, "Optimizing Weighted Extreme Learning Machines for Imbalanced Classification and Application to Credit Card Fraud Detection," Neurocomputing, vol. 407, pp. 50-62, DOI: 10.1016/j.neucom.2020.04.078, 2020.

[8] Z. Li, M. Huang, G. Liu and C. Jiang, "A Hybrid Method with Dynamic Weighted Entropy for Handling the Problem of Class Imbalance with Overlap in Credit Card Fraud Detection," Expert Systems with Applications, vol. 175, pp. 114750, DOI: 10.1016/j.eswa.2021.114750, 2021.

[9] S. Wang and X. Yao, "Multiclass Imbalance Problems: Analysis and Potential Solutions," IEEE Trans. on Systems, Man and Cybernetics, Part B (Cybernetics), vol. 42, no. 4, pp. 1119-1130, 2012.

[10] T. Hasanin, T. M. Khoshgoftaar, J. L. Leevy and R. A. Bauder, "Severely Imbalanced Big Data Challenges: Investigating Data Sampling Approaches," J. of Big Data, vol. 6, no. 1, pp. 1-25, 2019.

[11] B. Kulis, "Metric Learning: A Survey," Foundations and Trends in Machine Learning, vol. 5, no. 4, pp. 287-364, 2012.

[12] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems, vol. 25, pp. 1097-1105, 2012.

[13] A. Al-Shannaq and L. Elrefaei, "Age Estimation Using Specific Domain Transfer Learning," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 2, pp. 122-139, 2020.

[14] J. M. Johnson and T. M. Khoshgoftaar, "Survey on Deep Learning with Class Imbalance," Journal of Big Data, vol. 6, no. 1, pp. 1-54, 2019.

[15] R.-C. Chen and C.-Y. Liao, "Deep Learning to Predict User Rating in Imbalance Classification Data Incorporating Ensemble Methods," Proc. of the IEEE International Conference on Applied System Invention (ICASI), pp. 200-203, Chiba, Japan, 2018.

[16] N. Wang, X. Zhao, Y. Jiang and Y. Gao, "Iterative Metric Learning for Imbalance Data Classification," Proc. of the 27th International Joint Conference on Artificial Intelligence (IJCAI-18), pp. 2805-2811, [Online], available: https://www.ijcai.org/proceedings/2018/0389.pdf, 2018.

[17] L. Gautheron, A. Habrard, E. Morvant and M. Sebban, "Metric Learning from Imbalanced Data with Generalization Guarantees," Pattern Recognition Letters, vol. 133, pp. 298-304, 2020.

[18] S. Susan and A. Kumar, "Learning Data Space Transformation Matrix from Pruned Imbalanced Datasets for Nearest Neighbor Classification," Proc. of the IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 2831-2838, Zhangjiajie, China, 2019.

[19] S. Barua, Md. M. Islam, X. Yao and K. Murase, "MWMOTE - Majority Weighted Minority Oversampling Technique for Imbalanced Data Set Learning," IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 2, pp. 405-425, 2012.

[20] V. Ganganwar, "An Overview of Classification Algorithms for Imbalanced Datasets," International Journal of Emerging Technology and Advanced Engineering, vol. 2, no. 4, pp. 42-47, 2012.

[21] G. B. Huang, M. Mattar, T. Berg and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," Technical Report in Workshop on Faces in'Real-Life'Images: Detection, Alignment and Recognition, [Online], Available: http://vis-www.cs.umass.edu/lfw/lfw.pdf, 2008.

[22] O. M. Parkhi, A. Vedaldi and A. Zisserman, "Deep Face Recognition," Proc. of the British Machine Vision Conference (BMVC), pp. 41.1-41.12, [Online], Available: https://www.robots.ox.ac.uk/~vgg/ publications/2015/Parkhi15/parkhi15.pdf, Sep. 2015.

[23] K. Q. Weinberger and L. K. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," Journal of Machine Learning Research, vol. 10, no. 2, pp. 207-244, 2009.

[24] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 815-823, Boston, MA, USA, 2015.

[25] Y. Taigman, M. Yang, M. A. Ranzato and L. Wolf, "DeepFace: Closing the Gap to Human-level Performance in Face Verification," Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1701-1708, Columbus, OH, USA, 2014.

[26] Y. LeCun and Y. Bengio, "Convolutional Networks for Images, Speech and Time Series," The Handbook of Brain Theory and Neural Networks, vol. 3361, no. 10, pp. 1-14, 1995.

[27] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-scale Image Recognition," Proc. of ICLR 2015, arXiv preprint arXiv: 1409.1556, 2014.

[28] J. Goldberger, G. E. Hinton, S. T. Roweis and R. R. Salakhutdinov, "Neighborhood Components Analysis," Advances in Neural Information Processing Systems, pp. 513-520, [Online], Available: https://www.cs.toronto.edu/~hinton/absps/nca.pdf, 2005.

[29] K. Q. Weinberger and G. Tesauro, "Metric Learning for Kernel Regression," Artificial Intelligence and Statistics, pp. 612-619, [Online], Available: http://proceedings.mlr.press/v2/weinberger07a/Weinberger07a.pdf, 2007.

[30] J. V. Davis, B. Kulis, P. Jain, S. Sra and I. S. Dhillon, "Information-theoretic Metric Learning," Proc. of the 24th Int. Conf. on Machine Learning, pp. 209-216, DOI: 10.1145/1273496.1273523, ACM, 2007.

[31] E. P. Xing, M. I. Jordan, S. J. Russell and A. Y. Ng, "Distance Metric Learning with Application to Clustering with Side-information," Proc. of the 15th International Conference on Neural Information Processing Systems (NIPS'02), pp. 521-528, 2003.

[32] G.-J. Qi, J. Tang, Z.-J. Zha, T.-S. Chua and H.-J. Zhang, "An Efficient Sparse Metric Learning in High-dimensional Space *via* I_1-penalized Log-determinant Regularization," Proc. of the 26th Annual Int. Conf. on Machine Learning, pp. 841-848, DOI: 10.1145/1553374.1553482, ACM, 2009.

[33] H. S. Dadi and G. K. M. Pillutla, "Improved Face Recognition Rate Using HOG Features and SVM Classifier," IOSR Journal of Electronics and Communication Eng., vol. 11, no. 04, pp. 34-44, 2016.

[34] D. Chen, X. Cao, F. Wen and J. Sun, "Blessing of Dimensionality: High-dimensional Feature and Its Efficient Compression for Face Verification," Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3025-3032, Portland, OR, USA, 2013.

[35] Y. C. Wong, L. J. Choi, R. S. Sarban Singh, H. Zhang and A. R. Syafeeza, "Deep Learning-based Racing Bib Number Detection and Recognition," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 3, pp. 181-194, 2019.

[36] A. Kaushik and S. Susan, "Metric Learning with Deep Features for Highly Imbalanced Face Dataset," Proc. of the International Conference on Innovative Computing and Communications, Part of the Advances in Intelligent Systems and Computing Book Series, vol. 1394, pp. 639-646, 2022.

[37] B. Knyazev, R. Shvetsov, N. Efremova and A. Kuharenko, "Leveraging Large Face Recognition Data for Emotion Classification," Proc. of the 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), pp. 692-696, Xi'an, China, 2018.

[38] S. Karahan, M. K. Yildirum, K. Kirtac, F. S. Rende, G. Butun and H. K. Ekenel, "How Image Degradations Affect Deep CNN-based Face Recognition?," Proc. of the International Conference of the Biometrics Special Interest Group (BIOSIG), pp. 1-5, Darmstadt, Germany, 2016.

[39] T. Baltrušaitis, P. Robinson and L.-P. Morency, "OpenFace: An Open Source Facial Behavior Analysis Toolkit," Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1-10, Lake Placid, NY, USA, 2016.

ملخص البحث:

تقترح هذه الورقة منهجية تعلّم جديدة تتضمّن سِماتٍ عميقةٍ والتّعلُّم ذا الطّريقيْن لمجموعات البيانات المتعلّقة بتمييز الوجوه التي تتميّز بالضّخامة وشدّة عدم الاتّزان حيث عدد الأصناف الأقلية ومعدّل عدم الاتّزان كبيران جداً. وتبرز المشكلة من أنّ بعض الوجوه الأكثر شيوعاً متوافرة في وسائل التّواصل الاجتماعي والإنترنت، بينما وجوه الشّخصيات المعروفة بدرجةٍ أقلّ هي أقلّ عدداً. ولأنّ إعادة تشكيل العيّنات في ظلّ هذا السيناريو أمرٌ غير عمليّ، فإننا نقترح التّعلّم القياسي أداةً للتخفيف من مشكلة عدم الاتّزان في الأصناف قبل مرحلة التّصنيف. وللتقليل من تكلفة الحسابات المرتبطة بالتّعلّم القياسي، نقوم –بشكل منفصل- بإجراء التّعلّم القياسي المراقب بشكلٍ ضعيف على مجموعات البيانات الفرعية الأغلبية والأقلية، في عمليةٍ نسميها التّعلّم القياسي ذا الطريقيْن. وقد أثبتت التجارب على مجموعة البيانات المرجعية لتمييز الوجه LFW التي تحوي (1680) صنفاً من الوجوه أنّ المنهجية المقترحة كانت أكثر فاعلية مقارنةً بالطرق القائمة. حيث انّ دقّة التّصنيف للأصناف الأقليّة ارتفعت على نحوٍ خاص، وهو إنجاز نادر في طرق تصنيف مجموعات البيانات المعتمدة على التّعلّم غير المتوازن العميق.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 04, December 2021.

WORKFLOW SCHEDULING ACCORDING TO DATA DEPENDENCIES IN COMPUTATIONAL CLOUDS

Hamid Saadatfar[1] and Batoul Khazaie[2]

(Received: 19-Jul.-2021, Revised: 19-Sep.-2021, Accepted: 5-Oct.-2021)

ABSTRACT

The number of applications needing big data is on the rise nowadays, where big data processing tasks are sent as workflows to cloud computing systems. Considering the recent advances in the Internet technology, cloud computing has become the most popular computing technology. The scheduling approach in cloud computing environments has always been a topic of interest to many researchers. This paper proposes a new scheduling algorithm for data-intensive workflows based on data dependencies in computational clouds. The proposed algorithm tries to minimize the makespan by considering the details of the workflow structure and virtual machines. The concepts and details defined and considered in this study have received less emphasis in previous works. According to the results, the proposed algorithm reduced the duration of communication between tasks and runtimes by taking into account the features of data-intensive workflows and proper task assignment. Consequently, it reduced the total makespan in comparison with previous algorithms.

KEYWORDS

Bottleneck task, Computational volume, Data-intensive applications, Resource allocation, Sensitive task, Workflow scheduling.

1. INTRODUCTION

The current computing structures for today’s applications are growing in the form of a variety of networks, clusters and clouds, among which cloud computing has rapidly become a widely acceptable sample for scientific experiments, big data analysis and data-intensive applications. In such a distributed computing environment, data-intensive applications require high-performance computing resources to facilitate proper execution of tasks [1][2]. Scheduling is a very important problem in cloud computing systems, a goal of which is to make optimal and efficient use of resources and respond to users in real time. Due to the massive system scale and inherent complexity of applications, workflow scheduling is considered an NP-hard problem. Most of the big data applications contain hundreds of closely related tasks requiring to read or write massive amounts of data [3]. These applications which usually need to interact with their data are called data-intensive applications. The makespan of a data-intensive application is a very important efficiency criterion, which can be affected by many factors, such as the task scheduling mechanism, server load, communications and data access delay. For instance, the network performance can significantly affect data access delay and, as a result, the makespan [4][5][6]. The Directed Acyclic Graph (DAG) model is a popular, effective and common method of displaying complicated applications [7][8][9][10]. The workflows in data-intensive applications usually act as a series of interconnected tasks with data/control dependencies [11]. Increasing communication costs is evidently ideal for performance enhancement, especially for data-intensive applications. However, this is not always possible for various reasons. For example, data might be located in the local storage facilities, but the user is forced to outsource the computation to a cloud because of overloaded local processing nodes or needs to higher computing capacity [12]. Since the workflow scheduling of data-intensive applications has been considered a serious problem recently, this paper proposes a low-complexity heuristic algorithm to improve the runtime by considering data exchange, workload balance, properties of data-intensive applications and heterogeneity of machines.

The proposed algorithm attempts to reduce the average makespan by considering the properties of applications needing big data by defining new concepts, such as sensitive tasks and bottleneck tasks and scheduling based on considering machines’ characteristics (such as powerful or weak

[1] H. Saadatfar (corresponding author) is with Department of Computer Engineering, University of Birjand, Birjand, Iran. Email: saadatfar@birjand.ac.ir, ORCID: 0000-0002-6130-8450.

[2] B. Khazaie is with Department of Computer Engineering, Islamic Azad University, Birjand, Iran. Email: b.khazaie63@Gmail.com

communicating capabilities), noting the homogeneity and heterogeneity of tasks in the workflows of data-intensive applications and resources in the cloud environment. The proposed algorithm analyzes the workflow's graph structure and identifies the important subtasks (sensitive and bottleneck subtasks) considering the degree of vertices and the number of parents and children to reduce the runtime by striking balance in task mapping between machines. The algorithm consists of two steps to find the best machine for task assignment. The first step identifies and assigns sensitive tasks and the second step assigns insensitive and bottleneck tasks.

The rest of this paper is structured as follows: Section 2 reviews the literature. Section 3 presents the relevant concepts. In Section 4, the proposed algorithm and solution are introduced. Section 5 presents the simulation results. Finally, Section 6 reaches a conclusion and discusses future works.

2. RELATED WORKS

Cloud computing environments are nowadays used in nearly every field, such as engineering, mathematics, fundamental sciences and biotechnology. Significant studies have been conducted on management, scheduling and resource allocation and provisioning in cloud computing environments [3][7][13]. Workflow scheduling is used as an effective tool for system efficiency and performance enhancement. The workflows of data-intensive tasks (refer to large-scale data) are increasingly becoming more common in today's applications. Thus, scheduling algorithms should be modified and redesigned with respect to the specific features of these data-intensive workflows. Some of the studies on scheduling and resource allocation are reviewed in this section.

The paper presented by [14] introduced a workflow scheduling method for large-scale distributed systems from the perspective of Quality of Service (QoS) and data location analysis. The goal of this method is usually to provide a single relation and a scalable storage solution for cloud applications through storage on public services. The paper presented by [15] proposed an efficient algorithm for cloud workflow scheduling named Efficient Workflow Scheduling Algorithm (EWSA), which can manage a large number of applications simultaneously. The goal of this algorithm is to estimate the runtimes of all dynamic resources in order to maximize the use of resources and execute workflows in predetermined intervals. The paper presented by [16] introduced a Grouped Task Scheduling (GTS) algorithm by using QoS to estimate user requirements. This scheduler employs the min-min algorithm for the prioritization of batches. The authors in [17] employed the division algorithm for cloud computing scheduling with various databanks. A Divisible Load Theory (DLT) scheduling strategy is used for data-intensive computational loads in a heterogeneous cloud computing environment to minimize the total makespan and maximize the system efficiency by recovering a partitioned load from numerous databanks with respect to the distribution rate of databanks and the speed of each role. The paper presented by [18] analyzed task scheduling in the cloud environment by adopting a soft computing technique, in which the Genetic Algorithm (GA) was integrated with fuzzy sets for load balancing in the cloud environment. The final goal was to reduce the makespan in the cloud environment, so that computing resources would not be wasted. The paper presented by [19] introduced a Modified Genetic Algorithm (MGA) for resource-scheduling and proposed a workflow framework for cloud computing with a resource scheduling mechanism to improve the use of resources and system efficiency with respect to the QoS requirements.

The paper presented by [20] proposed an algorithm based on the Augmented Shuffled Frog Leaping Algorithm (ASFLA). The proposed algorithm determines the number of available virtual machines and employs the ASFLA to map the tasks in batches onto virtual machines for makespans. The paper presented by [21] introduced a workflow-scheduling technique aware of the data transfer duration and location based on the network bandwidth in a distributed environment with heterogeneous resources in addition to a local resource management method for data workflows in the cloud environment. The proposed algorithm employs the data workflow parallelization technique along the execution of tasks to reduce the execution cost of a workflow. In the paper presented by [22], a data-intensive application workflow-scheduling method was proposed for the heterogeneous computing environment (I-PDEA: Improved Partition-based Data-intensive Workflow Optimization Algorithm) to enhance the system throughput by partitioning data workflows and mapping each partition to the available heterogeneous resources having the minimum runtimes. A Partial Critical Path (PCP) algorithm was introduced in [11] to minimize the workflow execution costs while satisfying the makespan constraints. The Multi-Cloud Partial Critical Paths with Pre-treatment (MCPCPP) algorithm reduces the execution cost of a workflow in a makespan by finding multiple critical paths in that workflow and assigning them to the available computing services with the lowest cost. The paper presented by [23] proposed a scheduling algorithm, named the Granularity Score Scheduling (GSS) based on the details of the tasks in a workflow to minimize the runtime and maximize the efficiency of the workflow system in data-intensive applications. The proposed algorithm consists of three phases named level B (the maximum task length of an output task), task ranking and task mapping onto virtual machines. The paper presented by [24] introduced a scheduling algorithm which not only considered the computing capacity of existing virtual machines, but also the connection and access delays. The proposed algorithm also considered the different makespans of tasks to reduce the runtime by allocating tasks to more powerful machines. In addition, some of the subtasks can be executed simultaneously.

The paper presented by [25] introduced a scheduling algorithm on the volunteer computing systems. For this purpose, workflows were divided into sub-workflows to minimize data dependencies. The sub-workflows were then assigned to the distributed voluntary resources (executive nodes) with respect to the proximity of resources and load distribution policies. If the sub-workflows miss their sub-makespans due to a long waiting time, the scheduling will take place on the public cloud resources. The paper presented by [26] proposed a mechanism based on the workflow structure to identify the number of necessary virtual machines, configure these machines based on the aforementioned structure and optimize data transfer between tasks. This algorithm schedules tasks in a way that the number of executed tasks in each sample equals the number of virtual machines employed to run the workflow. The authors in [27] introduced a QoS-based workflow scheduling method and described the minimum effect imposed by the new input data to force rescheduling the workflow process. For this purpose, a database was used for data storage. The proposed algorithm is run through the Markov chain by scheduling the most complicated branch to the least complicated branch and allocating them on highly accessible machines with low costs. Given the importance of workflow scheduling in data-intensive applications and relevant studies conducted in recent years, most of the proposed approaches have focused on optimization of general scheduling and runtime minimization. None of the studies dealt with the different features of machines and the effects they might have on the execution of the DAG workflow. The different structures of machines have also been neglected.

The scheduling approach proposed in [28] considers the impact of communication between two tasks when building the schedule of a scientific workflow. It attempts to assign pairs of tasks with significant data transfers to the same computational node in order to minimize the overall communication cost. The authors in [29] proposed an immune particle swarm optimization algorithm (IMPSO) to solve the workflow scheduling problem with more speed and quality. In solving this problem, they have considered optimizing both execution time and cost criteria. In [30], the authors proposed a multi-objective workflow scheduler based on a prediction-based dynamic evolutionary algorithm. They also employed neural network to improve the answers' quality. They considered resources' failures to achieve more reliable task-resource mappings. They considered the estimated time to transfer data as a parameter in the objective function.

Researchers in [31] proposed a list-based scheduling algorithm called CAS-L1, which is a contention- aware algorithm. CAS-L1 is a heuristic scheduler based on lookahead technique, which schedules data transfers explicitly. Although this scheduler is a successful one for data-intensive workflows, it does not take into account the data exchanges of all workflows running at the same time and even other network traffic. The scheduler proposed in [32] called Minimum Communication Cost (MCC) algorithm focused on links’ available bandwidth and data files’ size that should be transformed. It can minimize communications effectively; however, lack of attention to the computing power of machines and their different ability to run different tasks is a point that has received less attention in this algorithm.

Compared to these previous works, we have defined and considered new concepts to focus on the data dependencies of a workflow that help map sub-tasks to a more appropriate machine. Also, the distinct ability of machines to perform different tasks has been considered in achieving better mappings. This paper tried to propose a novel approach by defining new concepts, considering different capabilities of machines to execute data-intensive tasks with respect to memory, processing speed, storage space, relevant effects on task execution, different structures of data-intensive workflows and complying with balance in existing machines and necessary data.

3. Relevant Concepts

3.1 Sensitive Task (T_S)

If the runtime of a task on some machines differs significantly from that on other machines, this task is called sensitive. In fact, when the runtimes of a task on different machines are sorted in an ascending order, the task is defined as sensitive if its runtime on one machine is at least two times longer than its runtime on the previous one [33]. This new concept has been defined in order to identify tasks the execution time of which has undergone many changes according to existing computing resources and to allocate resources with more attention to them.

3.2 Competent Machine

A machine or a series of machines can be considered competent to execute a task if the task's runtime on them is significantly shorter (at least a half) than the runtime on other machines. For instance, a machine with sufficient memory or high processing power is the competent machine for a task if it is capable of executing that task at a significantly shorter time interval compared to the other machines. Therefore, a competent machine is determined according to a specific task and the set of competent machines for a task includes all the machines from the whole cloud computing system that have a special ability to run that program and can do it at a run time at least a half of the run time on other machines.

Depending on the characteristics of the tasks and the computing power of the resources, the execution time of the tasks will vary on different computing machines. The more the capabilities of a resource are in line with the requirements of a task, the more execution time will be reduced. The concept of a competent machine is defined so that the same fact (the suitability of some resources to perform certain tasks) can be considered in the scheduling problem.

3.3 Bottleneck Task (Tbn)

A bottleneck task in a workflow is defined to be the one which has a considerably high data dependency on the other tasks of the workflow. Considering the graph structure of a workflow, a node is called a bottleneck if its degree is three times higher than the average degree of the total nodes in that graph. The existence of sub-tasks that create bottlenecks in the execution of the workflow is common in parallel and distributed processing and therefore, this new definition can help identify these tasks and allocate a better resource to them.

3.4 Machines with Fast and Slow Connections

A machine is called fast-connected if the average delay time of its directly-connected channels is fewer than a half of the average delay time of all other machines. However, if this delay time is two times higher compared to the other machines, it is called slowly-connected. Paying attention to resource characteristics can help better map tasks. Recognizing high-speed communication machines will provide good options for bottleneck tasks.

3.5 The Share of Each Virtual Machine from Computations

The computation share of a virtual machine from executing a workflow is calculated as follows:

$$CV = \frac{\sum_{j=1}^{n} \sum_{i=1}^{m} \frac{T_{ij}}{m}}{m} \qquad (1)$$

where *m* and *n* are parameters that indicate the number of machines and tasks, respectively and

T_{ij} represents the execution time of task j on machine i. Since a balanced division of tasks between machines can help increase efficiency, calculating the estimated share of each virtual machine in total workflow processing can be a good indicator of achieving this balance and determining the appropriate amount of load to be placed on each resource.

4. The Proposed Algorithm

4.1 Preliminary Definitions

A data-intensive workflow is represented by a directed acyclic graph (DAG). Each workflow consists of edges showing the dependency between tasks. A time limit is defined as a deadline for each workflow. In this case, a workflow like w contains n tasks $\{T_1, T_2, \ldots, T_n\}$. The system model includes m virtual machines $\{Vm_1, Vm_2, \ldots, Vm_m\}$ and E shows the edges in the workflow graph. Figure 1 indicates a data-intensive workflow. The edge between i and j means that task T_i is the prerequisite of task T_j. In other words, T_j can start only after T_i is completed. The weights of edges indicate the time relationship between every two tasks.

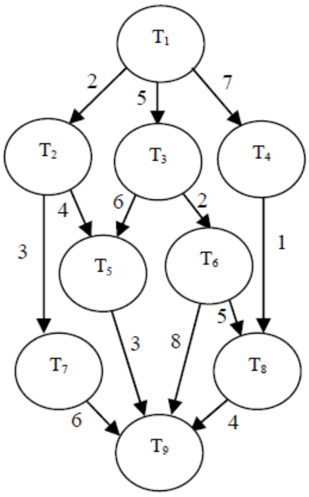


Figure 1. A DAG for a data-intensive application in the cloud computing environmen.

The makespan is the total runtime of all tasks in a workflow and it is defined as follows:

$$Ms=Mmax(Ms(VMj)) \tag{2}$$

Expected Time to Compute (ETC_{ij}) is a matrix with i rows and j columns ($1< i< n, 1< j< m$) which shows the runtimes of tasks on different machines:

$$ETC = \begin{array}{c} \\ T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_{n-1} \\ T_n \end{array} \left[\begin{array}{ccccc} Vm_1 & Vm_2 & Vm_3 & & Vm_{m-1} \\ ETC_{11} & ETC_{12} & ETC_{13} & \cdots & ETC_{1m} \\ ETC_{21} & ETC_{22} & ETC_{23} & & ETC_{2m} \\ ETC_{31} & ETC_{32} & ETC_{33} & & ETC_{3m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ ETC_{(n-1)1} & ETC_{(n-1)2} & ETC_{(n-1)3} & \cdots & ETC_{(n-1)m} \\ ETC_{n1} & ETC_{n2} & ETC_{n3} & & ETC_{mn} \end{array} \right] \tag{3}$$

Moreover, CT_{ij} ($1 < i, j< m$) is introduced as a matrix indicating the relationship between VM_i and VM_j based on the communication times:

$$CT = \begin{array}{c} \\ Vm_1 \\ Vm_2 \\ Vm_3 \\ \vdots \\ Vm_{m-1} \\ Vm_m \end{array} \left[\begin{array}{cccccc} Vm_1 & Vm_2 & Vm_3 & & Vm_{m-1} & Vm_m \\ 0 & CT_{12} & CT_{13} & \cdots & CT_{1(m-1)} & CT_{1m} \\ CT_{21} & 0 & CT_{23} & & CT_{2(m-1)} & CT_{2m} \\ CT_{31} & CT_{32} & 0 & & CT_{3(m-1)} & CT_{3m} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ CT_{(m-1)1} & CT_{(m-1)2} & CT_{(m-1)3} & & 0 & CT_{(m-1)m} \\ CT_{m1} & CT_{m2} & CT_{m3} & \cdots & CT_{m(m-1)} & 0 \end{array} \right] \tag{4}$$

As discussed earlier, the goal of scheduling in cloud computing is to reduce the runtime by assigning the tasks of a data-intensive application to machines such that the total makespan is minimized.

4.2 The Proposed Scheduling Algorithm

The proposed algorithm is a static workflow scheduling algorithm for the cloud computing environment. It is applied to the applications which require big data. It consists of two phases: sensitive tasks' scheduling and insensitive/bottleneck tasks' scheduling. The following pseudocode shows the details of the process.

Algorithm 1: Pseudocode of the Proposed Algorithm

```
Inputs: A DAG D = (T, E), an ETC matrix of size n × m and a CT matrix of size n × n
Output: Schedule S of DAG

Tsort =Sort ETC Matrix Rows in Ascending Order
sensitiveTasks=[ ]
BestMachines=[ ]
for i=1 to n
  for j=2 to m
    if Tsort(i,j)/Tsort(i,j-1) >=2
      Add i To sensitiveTasks set   // finding sensitive tasks
      BestMachines(i)=[$M_1$, $M_2$, ..., $M_j$]   //since the runtimes are sorted ascending
    end if
  End for j
End for i
AvgDeg=Calculate the average degrees of vertices of the DAG D
BottleNodes=[ ]
for i=1 to n
  if  degrees of vertice(i) >= 3* AvgDeg
    Add i To BottleNodes set   // finding bottleneck tasks
  end if
end for i
AvgCommunications = Calculate the total average communication time based on CT matrix
FastConnectionMachines=[ ]
SlowConnectionMachines=[ ]
For i=1 to n
  if  average time of communication for machine(i) <= AvgCommunications/2 then
    Add i To FastConnectionMachines set   // finding fast connected machines
  end if
  if  average time of communication for machine(i) >= AvgCommunication*2 then
    Add i To SlowConnectionMachines set
  end if
end for i
Sort sensitiveTasks by runtime in Descending Order
sensitiveTasks = sensitiveTasks - BottleNodes
ComputationalVolume = Sum(Average(runtime Tasks on machines)) /m
for i=1 to m
  Capacity(i)= ComputationalVolume   // setting the computation share of each machine
end for i
Assign=[ ]
for i=1 to length(sensitiveTasks)   // first phase
  for j=1 to length(BestMachines for task i)
    if (Capacity(BestMachines(j)>=ETC(i, BestMachines(j))
      Assign(i)= BestMachines(j)
      Capacity(BestMachines(j))= Capacity(BestMachines(j))- ETC(i,BestMachines(j))
    else if among the tasks of BestMachines(j), one can be moved to free space for task i
      move that task to another BestMachine
      Assign(i)= BestMachines(j)
      Capacity(BestMachines(j))= Capacity(BestMachines(j))- ETC(i,BestMachines(j))
  end for j
  if the task i has not yet been mapped to a machine
    assign the machine with minimum runtime for task i and update the Capacity
  end if
end for i
```

```
51  for each task i not assigned to a machine yet   // second phase
52    candidate_1= Machine with max(capacity)
53    candidate_2= Machine with min(ETC(task i))
54    candidate_3= Machine with the most parent or child tasks for the task i based on DAG D
55    if task i is a bottleneck one
56      candidate_4= Machine with fastest network connections
57      if candidate_1 is a SlowConnectionMachine
58        candidate_1=[ ]
59      end if
60      if candidate_2 is a SlowConnectionMachine
61        candidate_2=[ ]
62      end if
63    else
64      candidate_4=[ ]
65    end if
66    RT1 = Makespan of DAG D if  If the task i is mapped to candidate_1
67    RT2 = Makespan of DAG D if  If the task i is mapped to candidate_2
68    RT3 = Makespan of DAG D if  If the task i is mapped to candidate_3
69    RT4 = Makespan of DAG D if  If the task i is mapped to candidate_4
70    Assign task i to the machine with Min(RT1, RT2, RT3, RT4) and update its Capacity
71  end for
```

As can be seen, the proposed method considers the characteristics of resources and tasks, identifies sensitive and bottleneck tasks of the DAG and pays attention to the communicating capability of the machines. The resource allocation procedure is explained in more detail below.

On Lines (1-11), the algorithm finds sensitive tasks and their competent machines. As discussed in the previous section, the rows of the ETC matrix are sorted out in an ascending order first. Every row is then checked to see whether the runtime (*t*) of a task on a machine is twice or greater than that on the previous machine (Line 6). If true, that task is regarded as a sensitive task (T_S) and the previous machines (from the point on which the runtime is doubled) are considered competent for that task.

Lines (12-18) show how the average degrees of graph nodes are determined and how the bottleneck tasks are found. On Line 12, the average degree of the graph nodes is calculated. Line 15 determines whether there is a node the degree of which is three times or more than the average degree of the graph nodes. If there is a node matching the description, it is regarded as a bottleneck task (T_{bn}).

On Lines (19-29), the algorithm finds and determines fast and slow machines. In this section, the machine communication time matrix (CT_{ij}) is employed to determine the average time of communications between machines (Line 19). It is then checked whether the runtime of a machine is shorter than a half of the total average and if this applies, the machine is considered to be fast-connected. On the other hand, if the runtime is longer than twice the total average time, the machine is slowly-connected.

The next steps deal with how each task is allocated to the best machine such that the total runtime of the graph is reduced. The algorithm instructions are discussed in the following sub-sections. The main algorithm consists of two phases, the first of which is the allocation of sensitive tasks and the second phase is about the allocation of insensitive and bottleneck tasks.

4.2.1 First Phase: Allocation of Sensitive Tasks

First, the sensitive tasks are sorted in a descending order based on the average runtime of each task on Line 30. If also included among the bottleneck tasks, that sensitive task is not considered in this step (Line 31). This is because proper scheduling of bottleneck tasks is a key factor in reducing runtime in data-intensive workflows. As discussed earlier, since the balance between machines is a condition for the execution speed of tasks, Line 32 determines the computational load share of each machine to comply with the balance. In the next step, the sensitive tasks should be allocated to the best machine, which is performed in Lines 37-50. All of the sensitive tasks are sorted out in a descending queue and then the largest one is selected and allocated to the most competent machine. Since the computational capacity (the computation share) of each machine was determined earlier, the runtime of a task should be subtracted from the computational load of that machine after the task is allocated, so that the available time of each machine is obtained to observe the balance condition (Line 41). If a common machine is calculated to be competent for several sensitive tasks, then the best case should be selected for allocation. For instance, if multiple sensitive tasks are already running on a machine, when a new task is to be assigned to the machine to complete the computational capacity of that machine, some of the previous tasks should be allocated to their other competent machines in the following way:

Regarding every sensitive task on that machine, it should be checked whether there is another competent machine. If so, the reduction in the runtime should be determined. After determining the reductions in runtimes for all sensitive tasks on that machine, the machine offering the smallest runtime is selected and the sensitive task experiencing the smallest reduction is allocated (moved) to its other competent machine. After performing this process, all of the sensitive tasks are scheduled on their competent machines. When all the sensitive tasks are finished, the remaining tasks are allocated in the next phase.

4.2.2 Second Phase: Allocation of Insensitive Tasks and Bottlenecks

Regarding every insensitive task, three machines are selected from the existing available machines: 1- the emptiest machine (the machine with the lowest allocated computational load), 2- the fastest machine (the machine which performs the task in the shortest period) and 3- the machine that hosts the largest number of parents or children of the intended task (Lines 52-54).

If the task is a bottleneck, there is also a fourth option: 4- the machine with fastest communications (Line 56). If a task is a bottleneck and the main share (more than a half) in its degree is due to its children, this task should be assigned to a machine with high communication capacity. Therefore, if the fastest and the emptiest candidate machines are slow in terms of communication, these two machines will be excluded from the list of candidates (57-62). Lines 66-70 determine on what candidates the insensitive and bottleneck nodes have the minimum runtime. The best machine is found as follows:

The graph is analyzed from the first node. Every insensitive and bottleneck task would be assigned to a candidate; i.e., the emptiest machine, the fastest machine, the machine with the largest number of parents and children or the machine with fast communications to determine the total runtime of the graph. The task is then assigned to the candidate offering the minimum makespan. The following steps are taken to determine the total runtime of a graph for each task on the candidate machines:

1. If tasks are already assigned to a machine on both sides of an edge, both the runtime and connection delay will be known.
2. If only one of the two-sided tasks of an edge is mapped, the average runtime of all machines for the unmapped task will be considered as an estimate for its runtime. Also, the average communication time between the mapped task’s machine and other machines will be considered as an estimate for communication delay for this edge.
3. If no tasks are assigned on both sides of an edge, the runtime is considered the average runtime of all machines and the connection time is the average connection time of the entire graph.

The same procedure is followed to assign all tasks to their corresponding proper machines offering the minimum workflow makespan. In this study, fast-connected machines were employed to execute the bottleneck tasks given their large number of connections required; i.e., parents and children, so that the runtime could be optimized by reducing the communication delays. The makespan of the workflow is determined by finding the critical path of the graph.

4.2.3 Time Complexity of the Algorithm

Based on the pseudocode provided, it can be stated that most of the computations occur in nested loops of lines 37 to 50. In the worst case, if all the tasks are sensitive and all the machines are available for mapping, according to the search performed in the body of these loops, we can say that the algorithm of the proposed method has a complexity of the order of $O(m \times n \times \log n)$. Parameter m indicates the number of available machines and parameter n indicates the number of jobs.

5. Simulation Results

This section describes the performance evaluation of the proposed method. First, the benchmark workflows used in the experiments are described and then the system simulation conditions and configuration are detailed. Finally, the results of evaluation and performance analysis of the proposed method in comparison with GSS [23], CAS-L1 [31] and MCC [32] are described.

5.1 Benchmark Workflows

The workflows used in the experiments are generated based on real applications' structures. There are four types of workflow applications named Cybershake, Epigenomics, Montage and Inspiral (LIGO), which are used extensively for generating synthetic workflows in previous related research. Figure 2 shows these workflows:

1. Cybershake: This application is utilized to describe the hazards of earthquakes in a specific region by using the seismic hazard curve.

2. Epigenomics: This application is employed to show the epigenetic status of human cells on a large scale in genome.

3. Montage: This application was developed by NASA/IPAC to generate the aerial input images in a customized tiled format.

4. Inspiral (LIGO): This application was based on Einstein's theory for the detection of gravity waves.

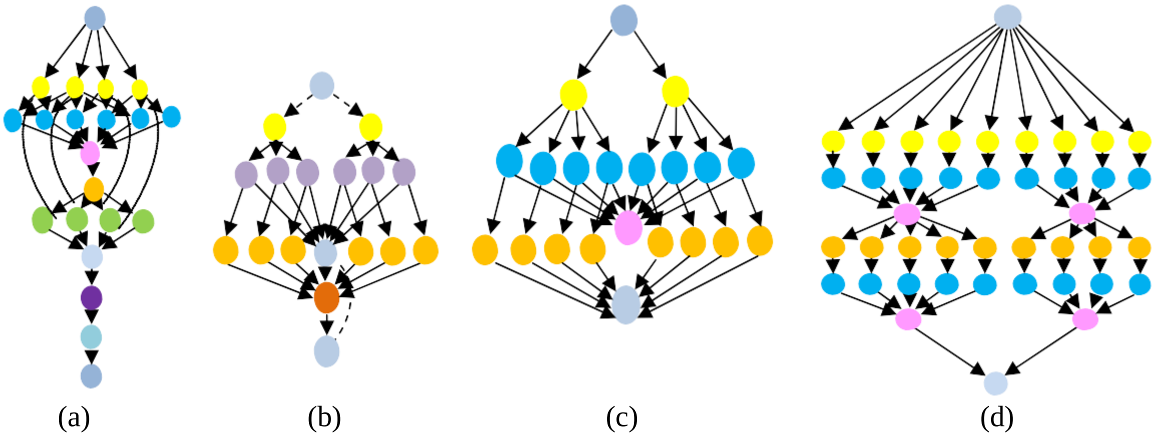


Figure 2. The structures of benchmark workflow applications [24].

More details of these applications can be found in [34][35]. Based on these structures, 400 synthetic workflows are generated using [35], 100 instances of each type. Since these structures are derived from real applications, it helps to have a fairer evaluation than using graphs that are made completely at random. Since this research focuses on data-intensive workflows and to better evaluate the proposed method, these workflows have been created from various CCR (communication to computation ratio) values (from 0.3 to 0.7). Higher values of the communication to computation ratio correspond to more intensives workflows. Task compute amounts and data transfer sizes are generated randomly with uniform distribution.

5.2 Simulated System Configuration

Cloudsim simulator [36] is used to model a realistic computing system and network. The system includes a cluster of 10 processing elements each one having 8 computing cores.

Therefore, 8 virtual machines can be run in parallel on each physical host. The computation power of these machines is set based on the characteristics of real-world cloud systems (i.e., Amazon's EC2). These machines are connected with Gigabit Ethernet network. However, communication links are considered to have different delays, which are determined randomly in the range of 10 to 100 milliseconds.

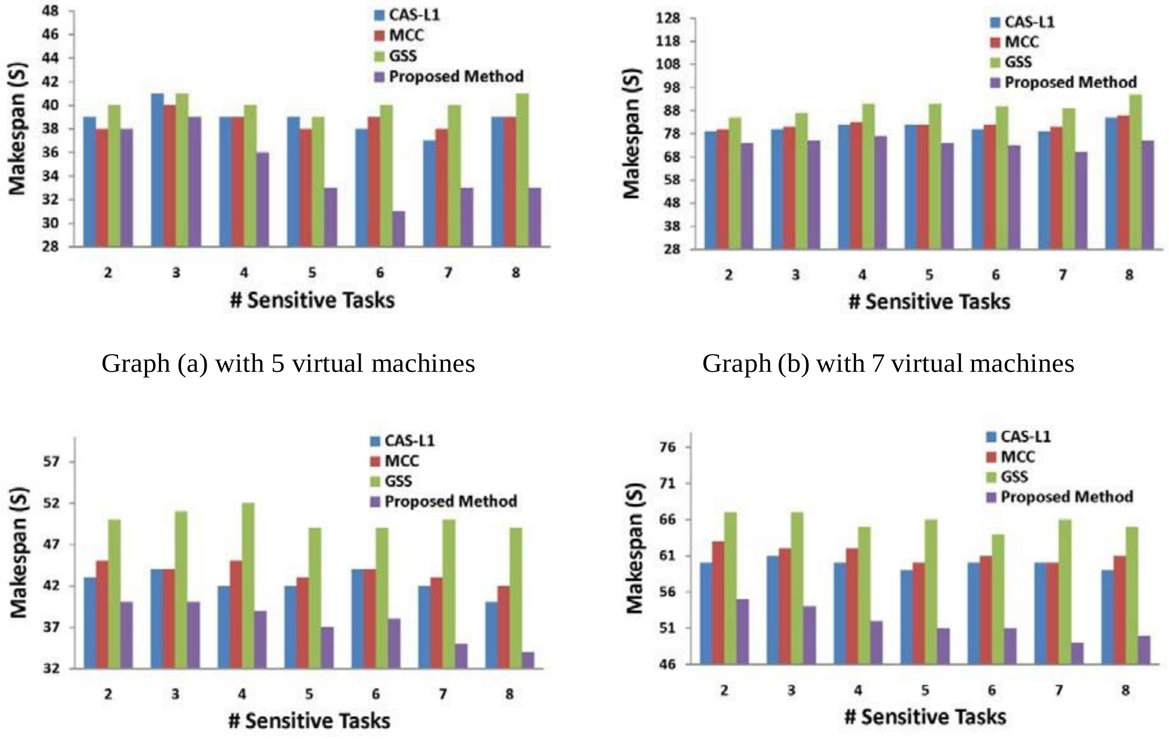
In the following part, the experiments performed are described and their results are analyzed. In order to examine the effect of defined concepts, such as sensitive work and bottlenecks, an attempt has been made to examine the performance of the proposed method in different cases of the number of sensitive tasks and the heterogeneity of resources in the system. The experiments described in the next sub-section were repeated 10 times and the mean values obtained were reported as a more reliable result.

5.3 Experimental Results

As discussed earlier, paying attention to the structure of workflows and considering the fact that one machine may be good for running one program while being not well for another, the proposed scheduling method is distinguished from previous ones. In fact, looking at the features of the task and the machine separately cannot always lead to proper mapping and one must look at the program and machine together. The concept of sensitive task defined in this research wants to say that among the available resources, some machines may be much more suitable for performing some tasks than others and this issue should be a priority in mapping tasks to resources. Three factors are considered in the performance of the proposed method and its performance is compared with previous work in terms of: the number of sensitive tasks, the number of employed machines and the heterogeneity or homogeneity of resources.

5.3.1 Different Numbers of Sensitive Tasks

In the first scenario, the number of sensitive tasks of each graph was changed in the workflows when the number of machines was constant. These changes can be controlled when setting the runtimes of each workflow on each virtual machine (the ETC matrix). Figure 3 shows the experimental results for performance evaluation of the proposed method.



Graph (a) with 5 virtual machines

Graph (b) with 7 virtual machines

Graph (c) with 8 virtual machines

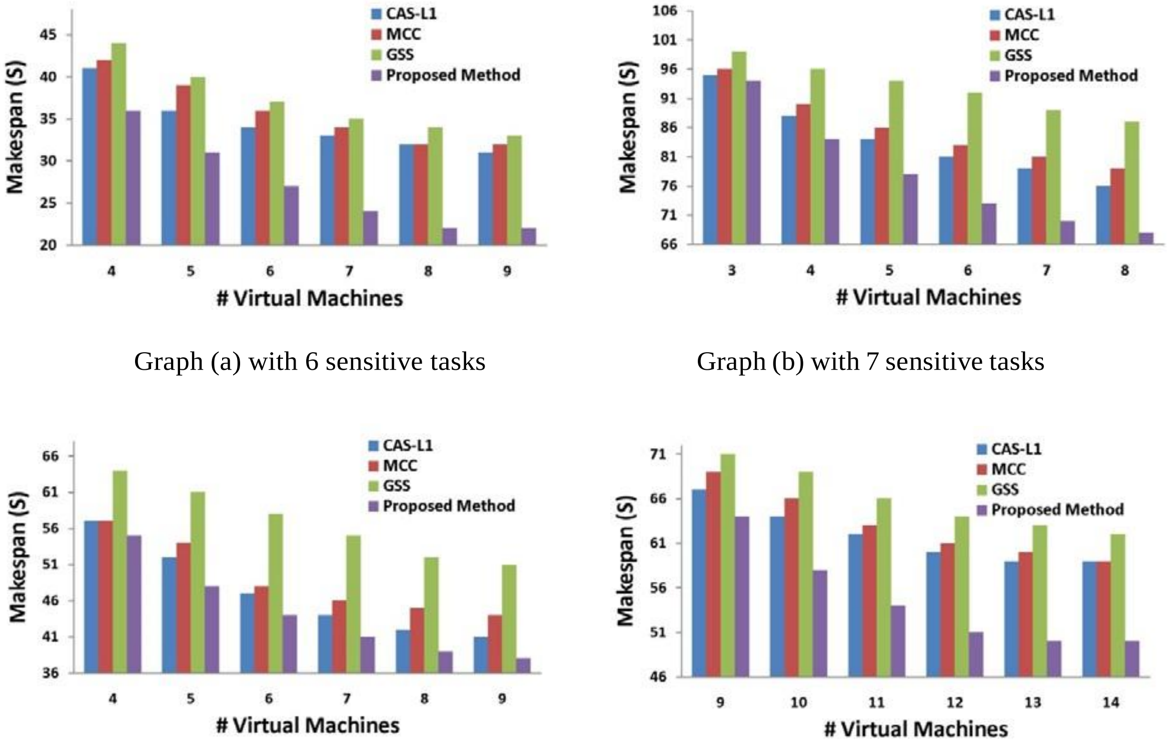
Graph (d) with 12 virtual machines

Figure 3. The performance of the proposed method in comparison with previous related works for different numbers of sensitive tasks in the workflows.

According to the results, a reduction in the execution time of workflows for the proposed method is visible compared to other methods. Paying attention to sensitive and vulnerable tasks and the priority in scheduling these tasks has led to this improvement. As the number of sensitive tasks in the workflow increases, the superiority of the proposed method in improving execution time becomes more apparent.

5.3.2 Different Numbers of Virtual Machines

In the second scenario, the performance of the proposed method in comparison with other methods is evaluated for different numbers of virtual machines. Other factors such as the number of sensitive tasks and the data-intensive nature of workflows are considered the same. Figure 4 shows the results for each type of workflow structure.



Graph (a) with 6 sensitive tasks

Graph (b) with 7 sensitive tasks

Graph (c) with 4 sensitive tasks

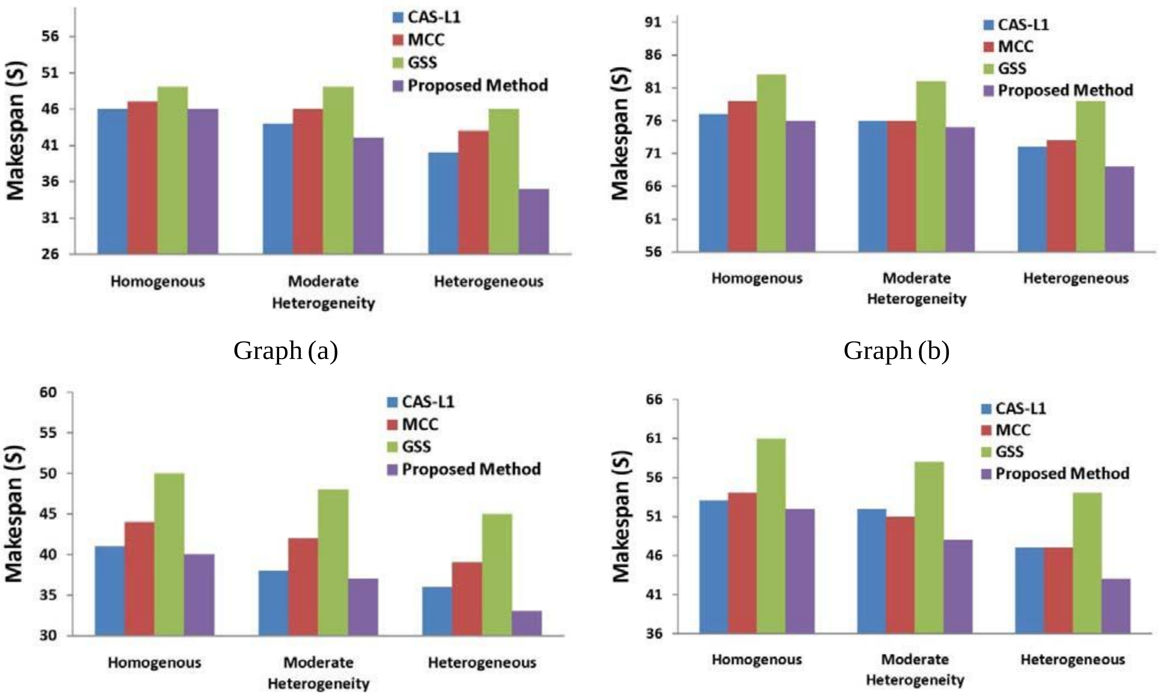
Graph (d) with 6 sensitive tasks

Figure 4. The performance of the proposed method in comparison with previous related works for different numbers of virtual machines.

Based on the results, it can be said that regardless of the number of virtual machines used, the proposed method shows better performance compared to similar previous methods. This may be due to the fact that the proposed method pays special attention to the structure of workflows, bottlenecks and sensitive tasks. The increase in the number of virtual machines in general has further improved the performance of the proposed method and this may be due to more options for scheduling decisions.

5.3.3 The Degree of Resource Heterogeneity

In the third scenario, an attempt has been made to evaluate the performance of the proposed method at different levels of resource heterogeneity. The results are presented in three categories: homogeneous, heterogeneous and moderate heterogeneity. It is expected that as the heterogeneity in resources increases, the performance of the proposed method will also improve. Since the proposed method pays attention to the fact that some resources have special capabilities to run some tasks, so increasing heterogeneity while increasing the options for scheduling, the chances of such capabilities among the available machines for execution of a task also increase.



Graph (a)

Graph (b)

Graph (c)

Graph (d)

Figure 5. The performance of the proposed method in comparison with previous related works at different levels of resource heterogeneity.

As can be seen from the results, the proposed method has been successful in reducing the execution time of data-intensive workflows compared to similar previous works. Attention to the graph structure of workflows, detection of bottlenecks and sensitive programs and allocation of appropriate resources to them with higher priority, attention to different features of machines in communication delays and balancing computational load can be mentioned as reasons for this performance.

6. CONCLUSIONS AND FUTURE WORK

Applications requiring big data have now gained more importance as their usage continues to grow. Therefore, it is necessary to focus on their scheduling methods and consider their differences to manage them better. In the proposed method, a heuristic scheduling algorithm with a low time complexity was introduced to minimize the workflow makespan by analyzing workflows' graph structure and the tasks' runtime on machines of different physical capabilities. This procedure also considered that all the data required by a task might be read from the database in a single attempt. Additionally, the connections between machines were taken into account to find fast-connected machines for those tasks requiring more data transformation; i.e., bottleneck tasks, in order to reduce the makespan of data-intensive workflows. The proposed method, based on the results, has had a more successful performance in reducing the total time required to execute workflows compared to other previous methods.

In future work, it is possible to deal with the interference of virtual machines located on one host on each other's performance and by modeling these effects to incorporate them into decisions. Also, considering the chance of machine failure when scheduling to increase system reliability can be another effective factor in scheduling decisions. Reducing energy consumption along with workflow execution time can turn the scheduling problem into a two-criterion optimization problem.

REFERENCES

[1] Y. Ahn and Y. Kim, "Auto-scaling of Virtual Resources for Scientific Workflows on Hybrid Clouds," Proc. of the 5th ACM Workshop on Scientific Cloud Computing (ScienceCloud '14), pp. 47-52, DOI: 10.1145/2608029.2608036, June 2014.

[2] L. F. Bittencourt and E. R. M. Madeira, "HCOC: A Cost Optimization Algorithm for Workflow Scheduling in Hybrid Clouds," Journal of Internet Services and Applications, vol. 2, pp. 207-227, 2011.

[3] S. Sagiroglu and D. Sinanc, "Big Data: A Review," Proc. of the IEEE International Conference on Collaboration Technologies and Systems (CTS), pp. 42-47, San Diego, CA, USA, July 2013.

[4] K. Wang, K. Qiao, I. Sadooghi, X. Zhou, T. Li, M. Lang et al., "Load-balanced and Locality-aware Scheduling for Data-intensive Workloads at Extreme Scales," Concurrency and Computation: Practice and Experience, vol. 28, pp. 70-94, 2016.

[5] M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker and I. Stoica, "Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling," Proc. of the 5th European Conf. on Computer Systems (EuroSys '10), pp. 265-278, DOI: 10.1145/1755913.1755940, April 2010.

[6] M. Zaharia, A. Konwinski, A. D. Joseph, R. H. Katz and I. Stoica, "Improving MapReduce Performance in Heterogeneous Environments," Proc. of the 8th USENIX Conference on Operating Systems Design and Implementation (OSDI'08), vol. 8, no. 4, pp. 29-42, December 2008.

[7] B. Lin, W. Guo and X. Lin, "Online Optimization Scheduling for Scientific Workflows with Deadline Constraint on Hybrid Clouds," Concurrency and Computation: Practice and Experience, vol. 28, pp. 3079-3095, August 2016.

[8] N. Xiong, X. Jia, L. T. Yang, A. V. Vasilakos, Y. Li and Y. Pan, "A Distributed Efficient Flow Control Scheme for Multirate Multicast Networks," IEEE Transactions on Parallel and Distributed Systems, vol. 21, no. 9, pp. 1254-1266, September 2010.

[9] J. Yin, X. Lu, X. Zhao, H. Chen and X. Liu, "BURSE: A Bursty and Self-similar Workload Generator for Cloud Computing," IEEE Trans. on Parallel and Distributed Sys., vol. 26, no. 3, pp. 668-680, 2015.

[10] Y. E. M. Hamouda, "Modified Random Bit Climbing (λ -mRBC) for Task Mapping and Scheduling in Wireless Sensor Networks," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 1, pp. 17-32, April 2019.

[11] B. Lin, W. Guo, N. Xiong, G. Chen, A. V. Vasilakos and H. Zhang, "A Pretreatment Workflow Scheduling Approach for Big Data Applications in Multicloud Environments," IEEE Transactions on Network and Service Management, vol. 13, no. 3, pp. 581-594, September 2016.

[12] A. N. Toosi, R. O. Sinnott and R. Buyya, "Resource Provisioning for Data-intensive Applications with Deadline Constraints on Hybrid Clouds Using Aneka," Future Generation Computer Systems, vol. 79, no.2, pp. 765-775, February 2018.

[13] M. Sohani and S. C. Jain, "Fault Tolerance Using Self-healing SLA and Load Balanced Dynamic Resource Provisioning in Cloud Computing," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 07, no. 02, pp. 206-222, June 2021

[14] G. L. Stavrinides, F. R. Duro, H. D. Karatza, J. G. Blas and J. Carretero, "Different Aspects of Workflow Scheduling in Large-scale Distributed Systems," Simulation Modeling Practice and Theory, vol. 70, pp. 120-134, January 2017.

[15] M. Adhikari and T. Amgoth, "Multi-objective Accelerated Particle Swarm Optimization Technique for Scientific Workflows in IaaS Cloud," Proc. of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1448-1454, Bangalore, India, September 2018.

[16] H. G. E. D. H. Ali, I. A. Saroit and A. M. Kotb, "Grouped Tasks Scheduling Algorithm Based on QoS in Cloud Computing Network," Egyptian Informatics Journal, vol. 18, no. 1, pp. 11-19, March 2017.

[17] S. Suresh, H. Huang and H. J. Kim, "Scheduling in Compute Cloud with Multiple Data Banks Using Divisible Load Paradigm," IEEE Transactions on Aerospace and Electronic Systems, vol. 51, no. 2, pp. 1288-1297, 2015.

[18] M. Kowsigan and P. Balasubramanie, "Scheduling of Jobs in Cloud Environment Using Soft Computing Techniques," Int. Journal of Applied Engineering Research, vol. 10, no. 38, pp. 28640-28645, 2015.

[19] W. Yan, W. Jinkuan and H. Yinghua, "Cloud Computing Workflow Framework with Resource Scheduling Mechanism," Proc. of the IEEE Chinese Guidance, Navigation and Control Conference (CGNCC), pp. 342-345, Nanjing, China, August 2016.

[20] P. Kaur and S. Mehta, "Resource Provisioning and Workflow Scheduling in Clouds Using Augmented Shuffled Frog Leaping Algorithm," Journal of Parallel and Distributed Computing, vol. 101, pp. 41-50, 2017.

[21] J. Shamsi, M. A. Khojaye and M. A. Qasmi, "Data-intensive Cloud Computing: Requirements, Expectations, Challenges and Solutions," Journal of Grid Computing, vol. 11, pp. 281-310, April 2013.

[22] S. G. Ahmad, C. S. Liew, M. M. Rafique and E. U. Munir, "Optimization of Data-intensive Workflows in Stream-based Data Processing Models," The Jour. of Supercomputing, vol. 73, pp. 3901-3923, 2017.

[23] M. S. Kumar, I. Gupta, S. K. Panda and P. K. Jana, "Granularity-based Workflow Scheduling Algorithm for Cloud Computing," The Journal of Supercomputing, vol. 73, pp. 5440-5464, June 2017.

[24] F. Xiong, C. Yeliang, Z. Lipeng, H. Bin, D. Song and W. Dong, "Deadline Based Scheduling for Data-Intensive Applications in Clouds," The Journal of China Universities of Posts and Telecommunications, vol. 23, no. 6, pp. 8-15, December 2016.

[25] T. Ghafarian and B. Javadi, "Cloud-aware Data Intensive Workflow Scheduling on Volunteer Computing Systems," Future Generation Computer Systems, vol. 51, no. C, pp. 87-97, October 2015.

[26] K. Kanagaraj and S. Swamynathan, "Structure Aware Resource Estimation for Effective Scheduling and Execution of Data Intensive Workflows in Cloud," Future Generation Computer Systems, vol. 79, no. P3, pp. 878-891, February 2018.

[27] S. Esteves and L. Veiga, "WaaS: Workflow-as-a-Service for the Cloud with Scheduling of Continuous and Data-intensive Workflows," The Computer Journal, vol. 59, no. 3, pp. 371-383, March 2016.

[28] I. Pietri and R. Sakellariou, "Scheduling Data-intensive Scientific Workflows with Reduced Communication," Proc. of the 30th International Conference on Scientific and Statistical Database Management (SSDBM '18), pp. 1-4, DOI: 10.1145/3221269.3221298, July 2018.

[29] P. Wang, Y. Lei, P. R. Agbedanu and Z. Zhang, "Makespan-driven Workflow Scheduling in Clouds Using Immune-based PSO Algorithm," IEEE Access, vol. 8, pp. 29281-29290, February 2020.

[30] G. Ismayilov and H.R. Topcuoglu, "Neural Network Based Multi-objective Evolutionary Algorithm for Dynamic Workflow Scheduling in Cloud Computing," Future Generation Computer Systems, vol. 102, pp. 307-322, January 2020.

[31] O. Sukhoroslov, "Toward Efficient Execution of Data-intensive Workflows," The Journal of Supercomputing, vol. 12, pp. 7989-8012, 2021.

[32] F. Li, "A Novel Scheduling Algorithm for Data-intensive Workflow in Virtualized Clouds," International Journal of Networking and Virtual Organizations, vol. 20, no. 3, pp. 284-300, June 2019.

[33] H. Saadatfar and H. Deldari, "A Study on Combinational Effects of Job and Resource Characteristics on Energy Consumption," Multiagent and Grid Systems, vol. 9, no. 4, pp. 301-314, January 2014.

[34] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta and K. Vahi, "Characterizing and Profiling Scientific Workflows," Future Generation Computer Systems, vol. 29, no. 3, pp. 682-692, March 2013.

[35] Confluence, "Workflow Generator," [Online], Available: https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator.

[36] T. Goyal, A. Singh and A. Agrawal, "Cloudsim: Simulator for Cloud Computing Infrastructure and Modelling," Procedia Engineering, vol. 38, pp. 3566-3572, DOI: 10.1016/j.proeng.2012.06.412, 2012.

ملخص البحث:

تقترح هذه الورقة خوارزمية جدولةٍ جديدةً لتدفّقات العمل كثيفة البيانات بناءً على اعتمادية البيانات في السّحابات الحاسوبية. وتحاول الخوارزمية المقترحة التقليل الى الحدّ الأدنى من نطاق العمل مع اعتبار تفاصيل بُنْية تدفّق العمل والآلات الافتراضية. وتجدر الإشارة الى أنّ المفاهيم والتفاصيل التي تمّ تعريفها وأخذها بعين الاعتبار في الخوارزمية المقترحة لم تنلْ ما يكفي من الاهتمام في أبحاثٍ سابقةٍ مماثلة.

وبناءً على النتائج، فقد قلّلت الخوارزمية المقترحة زمن الاتّصال بين المهمات وفترات التشغيل عن طريق أخذ سِمات تدفّقات العمل كثيفة البيانات والتّخصيص المحْكم للمهمات بعين الاعتبار. وبالنتيجة، خفّضت الخوارزمية المقترحة النّطاق الإجمالي للعمل مقارنةً بالخوارزميات الواردة في أعمالٍ سابقةٍ حول الموضوع.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 04, December 2021.

BRAIN-INSPIRED SPIKING NEURAL NETWORKS FOR WI-FI BASED HUMAN ACTIVITY RECOGNITION

Yee Leong Tan, Yan Chiew Wong and Syafeeza Ahmad Radzi

(Received: 17-Aug.-2021, Revised: 12-Oct.-2021, Accepted: 31-Oct.-2021)

ABSTRACT

Human activities can be recognized through reflections of wireless signals which solve the problem of privacy concerns and restriction of the application environment in vision-based recognition. Spiking Neural Networks (SNNs) for human activity recognition (HAR) using Wi-Fi signals have been proposed in this work. SNNs are inspired by information processing in biology and processed in a massively parallel fashion. The proposed method reduces processing resources while still maintaining accuracy through using frail but robust to noise spiking signals information transfer. The performance of HAR by SNNs is compared with other machine learning (ML) networks, such as LSTM, Bi-LSTM and GRU models. Significant reduction in memory usage while still having accuracy that is on a par with other ML networks has been observed. More than 70% saving in memory usage has been achieved in SNNs compared with the other existing ML networks, making SNNs a potential solution for edge computing in industrial revolution 4.0.

KEYWORDS

Human activity recognition, Wi-Fi signals, Spiking neural network.

1. INTRODUCTION

The area of human activity recognition (HAR) has become one of the most popular study fields due to the increment in computer vision technology and the availability of sensors at cheap prices and small sizes. With the help of technology, the data can be collected easily through wearable sensors, images or video frames [1]. However, there have been some problems in image processing using data collected from video or images. Firstly, privacy is a big concern and secondly, there are line of sight issues where the object needs to be inside the camera capturing area for a model to learn.

Indoor HAR using Wi-Fi has its benefits compared to the traditional technique where no extra wearable device is needed and there is no line of sight limitation. Received Signal Strength (RSS) has been used in human movement tracking due to its simplicity and easiness of measurement [2]. However, RSS-based algorithms have several limitations, due to the multipath and random noise in the enclosed environment havings a significant impact on RSS [3]. Therefore, a modified Wi-Fi system called channel state information (CSI) is used, which enables more information to be extracted from the signal captured. There will be a reflection in the wireless signal which will cause variation in CSI when detecting the movement of the human body. Unlike RSS, CSI consists of two types of information which are amplitude and phase information. Authors in [4] state that there have been some sources, such as carrier frequency offset (CFO) and sampling frequency offset (SFO), which often exacerbate the phase information. However, the amplitude information of CSI is more stable and has commonly been used for HAR. There have been several types of machine learning (ML) methods, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). Both CNNs and RNNs are famous image-processing methods that result in high accuracy and had been used in many areas, such as functional magnetic resonance imaging (MRI) for autism spectrum disorder detection [5]-[6], racing bib number recognition [7] and HAR [8]. However, more hardware resources are needed for these ML methods. To solve this, a new type of neural network called Spiking Neural Networks (SNNs) can be applied to this Wi-Fi CSI-based HAR.

2. RELATED WORK IN HUMAN ACTIVITY RECOGNITION (HAR) USING CSI

The Wi-Fi system can be built up with a transmitter and a receiver. The router sends out the signal with a certain frequency and the frequency is received by a laptop with a CSI reader, such as NIC

Y. L. Tan, *Y. C. Wong (ORCID: 0000-0003-2483-9962) and A. R. Syafeeza are with Universiti Teknikal Malaysia Melaka, Malaysia. Emails: petertanyeeleong@gmail.com, *ycwong@utem.edu.my and syafeeza@utem.edu.my.

5300 chips to record the CSI readings on the other side in the line-of-sight (LOS), as shown in Figure 1. When there is a movement of a human in the LOS of the Wi-Fi system, it can cause a variation in the Wi-Fi signal which is recorded as amplitude and phase information in CSI. The dataset will have 30 subcarriers for each receiver or 30 packets received by each receiver. There are three receivers in NIC 5300 chip, thus 90 subcarriers for each amplitude and phase information will be gathered. The movement of humans will cause different signal reflections and this means that the signals can be used for recognizing human activities through the ML method.

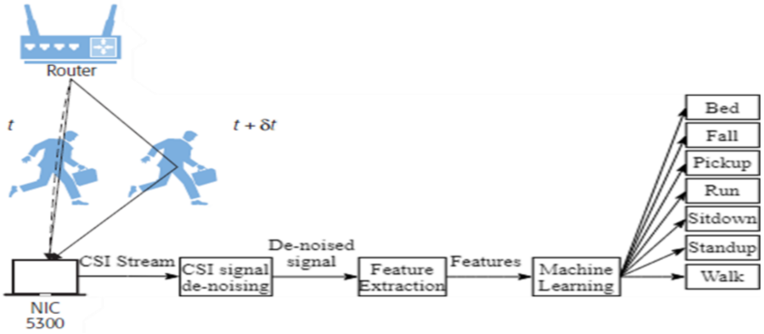
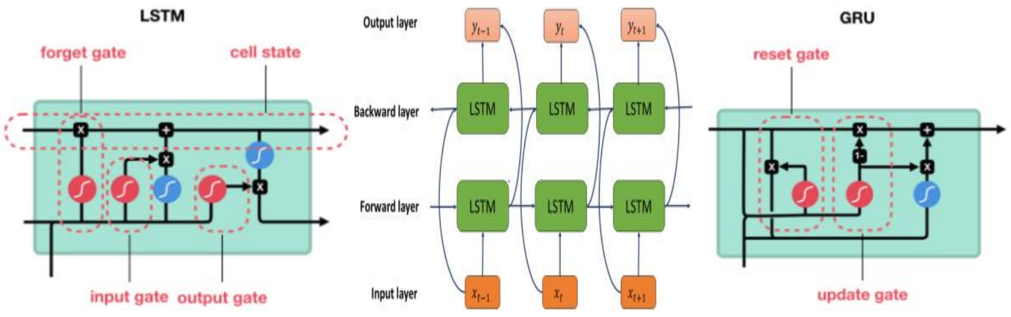


Figure 1. Human activity recognition using channel state information extracted from Wi-Fi.

2.1 Existing Machine Learning (ML) Models

CSI data has been used for location-oriented HAR using the moving variance thresholding method in E-eyes [9]. CARM [10] is a model that uses the relationship between Wi-Fi signal dynamics theory and human activities in activities classification. Beside these non-neural network methods, RNN has also been used for HAR. Several famous RNN methods have been proposed by other researchers to solve HAR using CSI data, such as Long Short-Term Memory (LSTM) [11], Bidirectional LSTM (Bi-LSTM) [4] and Gated Recurrent Units (GRU) [12] model. The structure of these models is shown in Figure 2. LSTM is a kind of RNN that is capable of learning long-term relationships, notably in sequence prediction issues. LSTM has two different states called cell state and hidden state which carry long- and short-term memory, respectively, between the cells. Bi-LSTM is an upgraded version of LSTM, where each training sequence will go through the model in both forward and reverse orders within two hidden layers, but with the same output layer. GRU model is a model that is significantly less complicated than an LSTM, since it has only two gates (update gate and reset gate), while LSTM has three gates (forget, input and output gates) in each cell. GRU can perform similarly to LSTM with a shorter training time. However, when having a bigger size of datasets, LSTM is said to have a better result than GRU. Since these three models are a kind of RNN model, they can extract the features of the datasets during training with all the CSI datasets inserted into the model.



a) b) c)

Figure 2. Machine learning models: a) LSTM, b) Bi-LSTM and c) GRU.

When compared with the Random Forest (RF) and Hidden Markov model (HMM), LSTM in [11] had obtained a high accuracy in classification. Authors in [4] proposed an attention-based Bi-LSTM model, which has a higher accuracy, but the time consumed is twice that of the LSTM model. The deep GRU algorithm in [12] achieves a high average accuracy of 98.12% when compared with RF and LSTM algorithms.

Although RNN-based models can achieve a high accuracy, they need a high memory bandwidth on demand to ensure that the classification can take place with the required low latency in edge computing. Memory bandwidth and latency are critical concerns in edge computing. The quantity of data that may be transferred to or from a particular destination is referred to as bandwidth, while latency is the time for an operation to complete. Edge computing is the science of allowing data to be processed at the edge of the network without having to send it to a central server. Therefore, low bandwidth and latency are the keys that enable edge devices to analyze the data in near real-time. Spiking Neural Networks (SNNs), a novel type of neural network, can be used to overcome this problem.

2.2 Spiking Neural Networks (SNNs)

SNNs represent the third generation of neural network models if neural network models are classified based on their computational units [13]-[14]. SNNs use spikes for information encoding and use spiking neurons or integrate-and-fire neurons as computational units. The computational units of the first generation are based on McCulloch-Pitts neurons or in other words, perceptron and threshold gates. For the second generation of the neural network models, an activation function, such as sigmoid and linear saturated functions, is applied to the output of every possible output value.

SNNs are a type of promising artificial neural network (ANN). Their outputs were presented in trains of spikes, having three new dimensions in the structure and the functionality of ANNs which are time, phase and frequency [15]. Not like ANNs, SNNs can be considered time-dependent during computation since the single-bit impulses or the so-called spikes will be sent out by neurons to synapses, potentially in a non-zero traveling time. SNNs are highly inspired by natural brain computation and recent developments in neuroscience [16]. With the involvement of spike firing timing, the strength and interest of SNNs are gained from the precise modeling of neuronal synaptic connections. Hence, the computational power of the 1st and 2nd generations was overcome by SNNs.

SNNs have been said to be the artificial neural network type that is closest to human neurons, because it transfers the data information in spike, having timing characteristics. In a human neuron, the postsynaptic neuron receives the pulse (spike) from other presynaptic neurons and action potential is produced when the membrane potential accumulated reached the threshold voltage. The action potential produced will be the input pulse for the next postsynaptic neuron. The spiking neuron model has also been called as biological neuron model, because it can simulate the motion of activation function in the human neurons.

There have been several existing spiking models such as Hodgkin-Huxley model, the earliest model that was introduced in the year 1952 by Alan Hodgkin and Andrew Huxley, Integrate-and-Fire (IF) model, a model that uses a basic mathematical equation to explain the motion of activation function in the neurons and also the Izhikevich model, a model having the characteristics of both Hodgkin-Huxley and IF models. Figure 3 shows the motion of an action potential, where a spike is emitted when the voltage has reached the threshold.

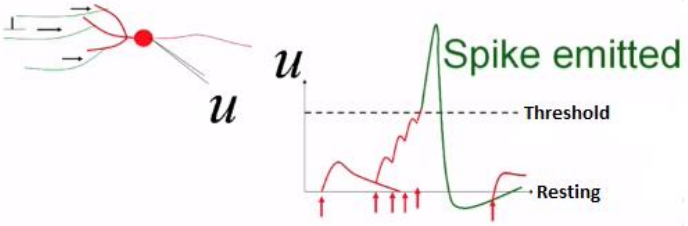


Figure 3. Spike generated with the motion of action potential exceeding the threshold in the neuron.

SNNs had shown their flexibility in image processing works and great work in energy saving, since they are promising high importance to biological systems. The energy-saving of the spiking model has been achieved by more than 90%, where energy consumption is reduced from 1.69W-28W to 0.136W compared to the Support Vector Machine and CNN model in [17]. In [18], 95% of energy saving has been achieved, where energy is reduced from 657.5µJ to 31.5µJ compared with the fully connected deep neural network model. Several types of SNNs are proposed by other researchers, such as fuzzy SNNs [19], artificial SNNs [20] and reservoir-based convolution SNNs [21], in their works. Along with that, there have also been learning rules proposed by other researchers, such as synaptic weight association training (SWAT) [22], Synaptic Efficacy Function-based leaky-integrate-and-fire neuRON (SEFRON) [23] and spike-timing-dependent-plasticity (STDP) with dynamic threshold neurons [24].

3. Methodology

In this paper, a spiking model based on Synaptic Efficacy Function-based leaky-integrate-and-fire neuRON (SEFRON) [23] has been used to classify human activities from the CSI dataset. The SNN model trains to classify the data with seven different human activities, which are ‘Bed’, ‘Fall’, ‘Pickup’, ‘Run’, ‘Sitdown’, ‘Standup’ and ‘Walk’ on MATLAB software.

3.1 Data Pre-processing

The dataset used is based on the Kazuki dataset having three receivers (30 subcarriers for each receiver) with a 50 Hz sampling frequency for 19.8s (990 data for each subcarrier) [25]. Since SNN is a time-dependent neural network, only the data of the first receiver and the first subcarrier is used in this paper for the classification method. MATLAB smooth function is used as the denoise method to remove noise from data. With the default setting, the function smooths the response data using a moving average low-pass filter with the filter coefficient equal to the reciprocal of the span. Span is the number of data points used for calculating the smoothed value and the number is 5 in default. An overview of data selection and pre-processing is shown in Figure 4. For the LSTM, Bi-LSTM and GRU models with a total of 90 input neurons, all the 90 subcarriers are used, where each subcarrier inserts to each input neuron for a time range of 990 data without any data pre-processing.

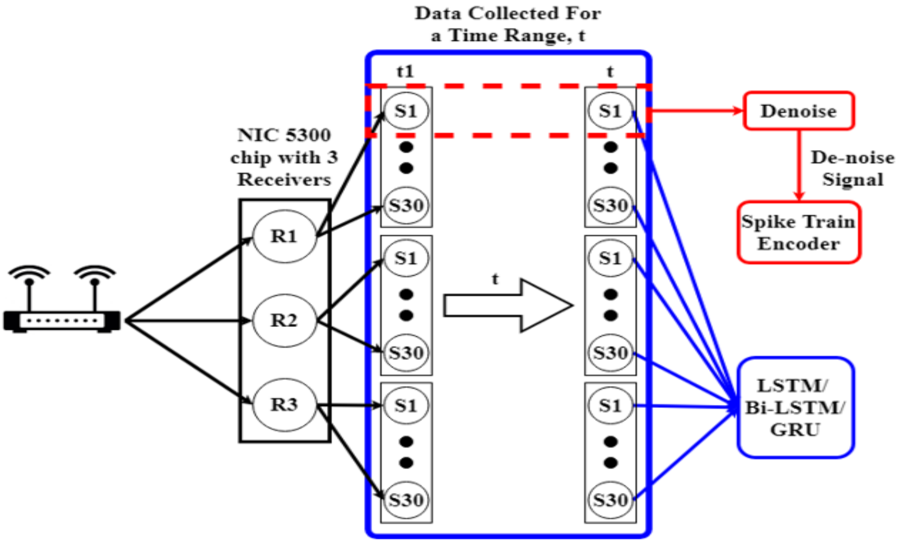


Figure 4. Overview of data pre-processing and data selection for Spike Train Encoder, LSTM, Bi-LSTM and GRU.

3.2 Spike Train Encoder

The data consists of analogue values and needs to be converted into the spike train pattern that is needed by the Spiking Neuron Model. The population encoding method is useed to convert the analogue CSI data into the spike train needed by SNNs. By setting the number of receptive field neurons, each real value converts the presynaptic spike along with the presynaptic spike time interval. In this paper, each spike train is designed to have a presynaptic spike along with the time interval of [0,2.5] ms. Since the precision of the time step is set as 0.01, the time step interval is [0.250]. An overview of the spike train encoder is shown in Figure 5.

3.3 SNN Model

In this paper, each analogue value in the spike train pattern is the input synapse for the SNN model. Input presynaptic neurons are multiplied with their time-varying weight and summed up at the output postsynaptic neuron. There are 7 layers of time-varying synaptic weight and output postsynaptic

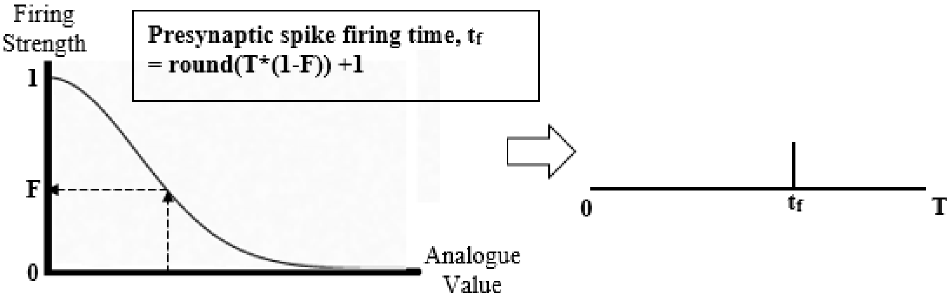


Figure 5. Overview of Spike Train Encoder.

Neurons, since there are 7 classes of activities. The synaptic weight in this model can be either positive or negative along with the time interval. At the output postsynaptic neuron, a postsynaptic spike is fired when the voltage reaches its threshold at time t. The postsynaptic spike time interval is set to [0,4] ms with time step interval of [0,400] to allow the model to capture the postsynaptic spike after time T.

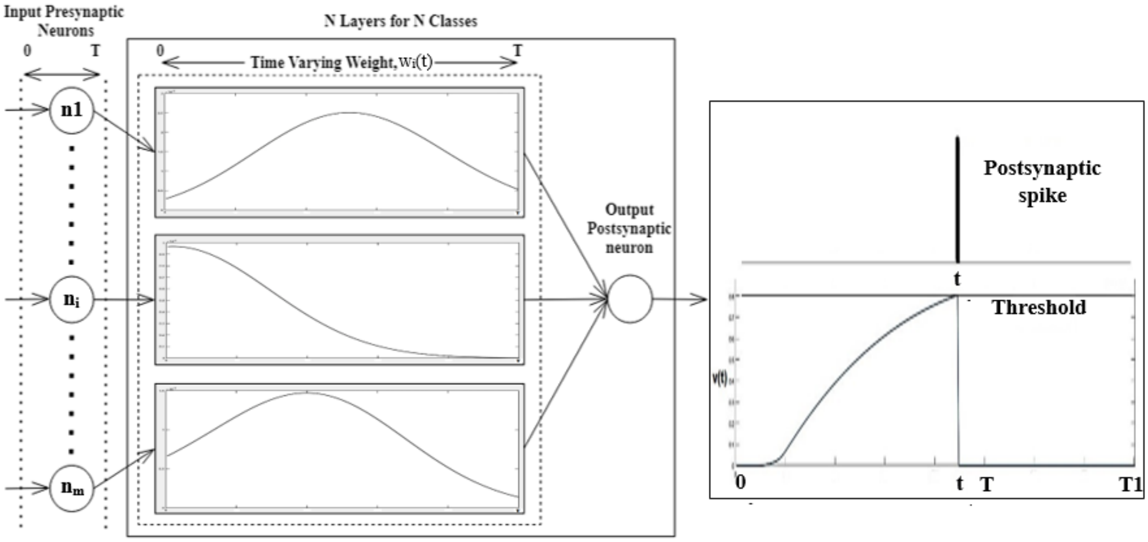

Input Presynaptic Neurons
0
T
n1
ni
nm
N Layers for N Classes
0
Time Varying Weight, wi(t)
T
Output Postsynaptic neuron
Postsynaptic spike
t
Threshold
v(t)
0
t
T
T1

Figure 6. A model with m number of input presynaptic neurons. Spike train input pattern in a time interval of [0, T] and postsynaptic output spike time interval of [0, T1].

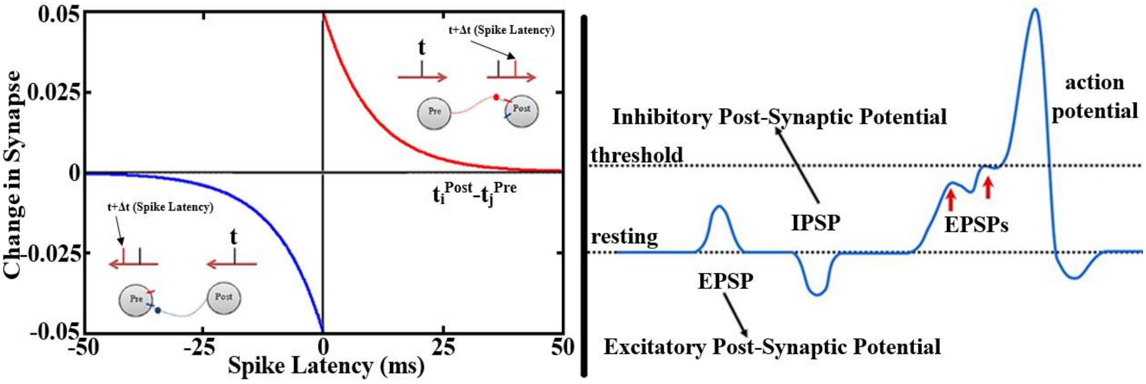
3.4 Model Training

In training, only the first actual postsynaptic spike is important and is used for the classification method. If the postsynaptic spike does not fire, the firing time is taken as the end of the simulation time. Since only the first postsynaptic spike is used by the SNN model, the membrane potential, or the postsynaptic spikes after the first postsynaptic spike, is not considered. The target class postsynaptic neuron is designed to fire a postsynaptic spike at the desired postsynaptic firing time of 2ms. Weight update with a 0.5 learning rate occurs when the target class postsynaptic neuron is fired outside the firing time or another class fires in the desired postsynaptic firing time range of 2±0.05 ms. The output postsynaptic neuron that is fired first is the output class for the data.

Normalized STDP learning rule, with a learning window size of 1.5 ms, computes the results of postsynaptic potential due to fractional contribution ($V_{STDP}(t)$) in both actual and desired postsynaptic spike firing times. Then, by minimizing an error function in the ratio of threshold and $V_{STDP}(t)$, it calculates the change in the time-varying weight on both actual and desired postsynaptic spike firing times. Finally, the learning rule uses a Gaussian distribution function centred at the current presynaptic spike timing to regulate the change in synaptic time-varying weight.

3.5 Spike-Timing-Dependent-Plasticity (STDP) Learning Rules

SNNs can be trained for solving pattern classification problems, using normalized STDP supervised learning rules to reduce the error by computing the changes in the weight. STDP is a neuron's rule strengthening or weakening the connections between neurons based on the degree of synchronous firing. When the postsynaptic spike timing is after the presynaptic spike timing, the synaptic weight will be strengthened in a positive value if the spike latency is near zero. On the other hand, for the postsynaptic spike that spikes before the presynaptic spike, the synaptic weight will be strengthened in a negative value for the spike latency that is near to zero as shown in Figure 7 a). The effect of these positive and negative weight values is shown in Figure 7 b), where a positive weight value will cause an excitatory postsynaptic potential (EPSP) and a negative weight value will cause an inhibitory postsynaptic potential (IPSP) in the postsynaptic neurons.



a) b)

Figure 7. a) STDP learning rule and b) Excitatory and inhibitory Postsynaptic potential.

4. RESULTS

By adjusting the desired postsynaptic firing time parameter, the threshold value for each class to fire is influenced. A lower threshold indicates that it is easier to have a postsynaptic spike, while a higher threshold indicates that more presynaptic spikes are needed to fire a spike on the postsynaptic neuron. Table 1 shows the threshold values for each class with different desired postsynaptic firing time values. The effect of low and high thresholds in training spike time is shown in Figure 8. It seems that the model is firing earlier with a lower firing threshold than the model with a higher firing threshold. The classes which do not fire are considered fired at the end of the simulation time.

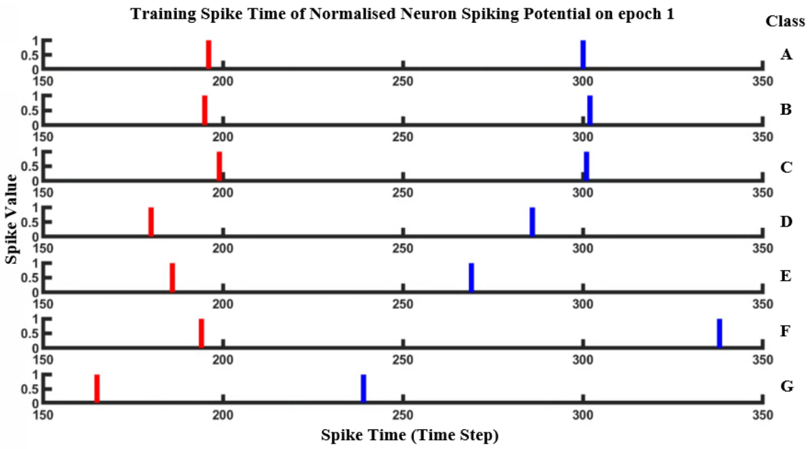


Figure 8. Effect of threshold in training postsynaptic spike time with a lower threshold (red colour) and a higher threshold (red colour).

Table 1. Threshold values for postsynaptic in different classes.

PARAMETER	CLASS						
Desired postsynaptic firing time step	A-Bed	B-Fall	C-Pickup	D-Run	E-Sitdown	F-Standup	G-Walk
150	0.6044	0.5822	0.6965	0.4908	0.6058	0.6485	0.6085
220	0.8500	0.8374	0.8967	0.7884	0.8480	0.8728	0.8516

Figure 9 shows the example increase of membrane potential due to a spike in the spike train fired as a postsynaptic spike at a time of 263ms when reaching the threshold. Since only the first postsynaptic spike is used by the SNN model, the membrane potential, or the postsynaptic spikes after the first postsynaptic spike, was not considered. The postsynaptic spike firing output for class G from both training and testing datasets for all epochs is shown in Figure 10.

Table 2 shows the comparison of the results of SNNs with other ML models, such as the Bi-LSTM, LSTM and GRU models with 100 hidden neurons each. SNN model achieves 85% classification accuracy on the testing data which is higher than that of the Bi-LSTM model. Figure 11 shows the memory profiling summary graph for all the models in Table 2, with the stacks representing the functions that are running in that period. Comparing the memory needed by the model, SNNs only need 13620kb of memory size to run; however, the Bi-LSTM model needs the largest memory size which is 108520kb. In this case, the SNNs model shows a saving in the memory usage which saved 74%, 87% and 72% of memory when compared with the LSTM, Bi-LSTM and GRU models, respectively.

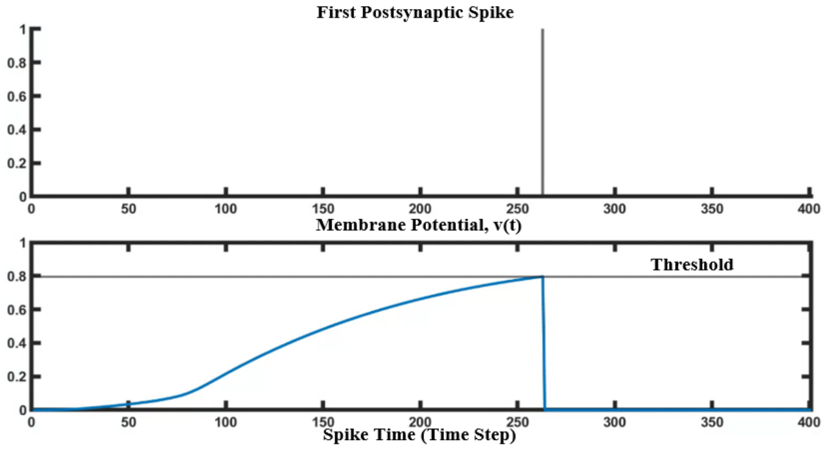


Figure 9 . Postsynaptic spike caused by membrane potential.

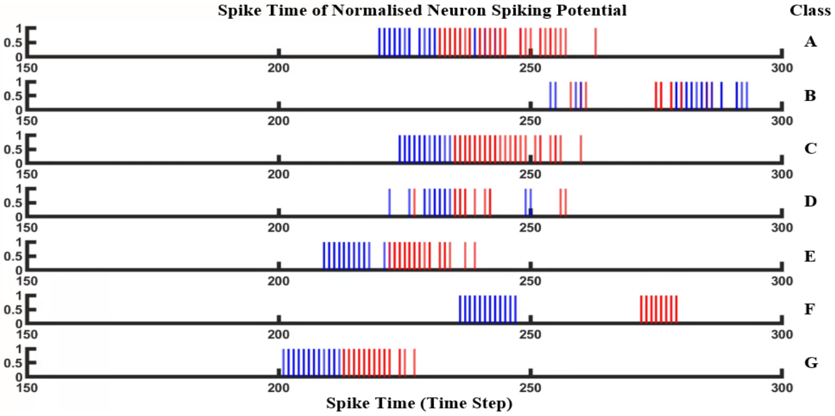


Figure 10. Example of firing output for one of the class G training (blue colour) and testing datasets (red colour) for all epochs.

Table 2. Results of SNN model and other ML models.

Model	SNNs	LSTM	Bi-LSTM	GRU
Number of inputs	990	90	90	90
Number of hidden neurons	-	100	100	100
Overall training accuracy (%)	100	100	100	100
Overall testing accuracy (%)	84.8073	85.71	80.00	88.57
Learning rate	0.5	0.001 (Adam optimizer)		
Time for training and testing all dataset (s)	71	84	184	73
Peak memory (kb)	13620	54252	108520	49100

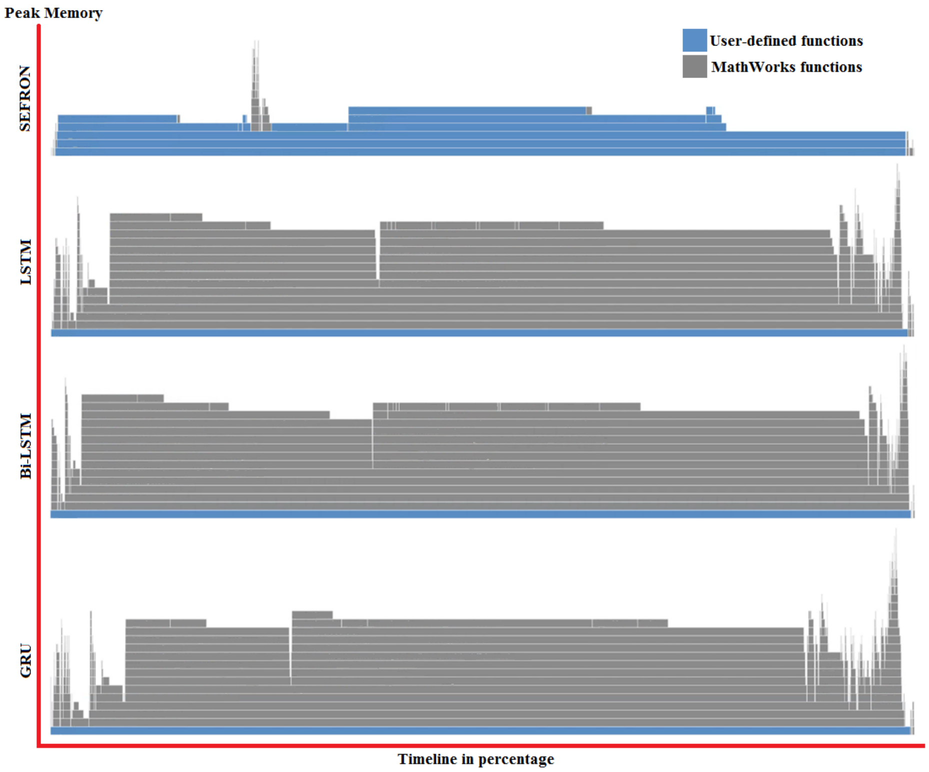


Figure 11. Graph of memory profiling summary.

Table 3 shows the confusion table of testing data for the SNN model and the other compared models. All the test data for activities ‘A-Bed’, ‘B-Fall’, ‘C-Pickup’ and ‘D-Run’ has been predicted correctly by the SNN model. On the other hand, the accuracy to predict activities of ‘E-Sitdown’ and ‘F-Standup’ is very low. This may be due to the threshold for these activities that are very close to each other and have a miss-postsynaptic spike at wrong classes. Besides that, similar or less clear features among the datasets of different classes may cause a decrease in the classification accuracy.

Table 3. Confusion table for testing data where activity A to activity G refer to ‘Bed’, ‘Fall’, ‘Pickup’, ‘Run’, ‘Sitdown’, ‘Standup’ and ‘Walk’.

SNNs							
Actual	Predicted						
	A	B	C	D	E	F	G
A	1.0	0	0	0	0	0	0
B	0	1.0	0	0	0	0	0
C	0	0	1.0	0	0	0	0
D	0	0	0	1.0	0	0	0
E	0.4	0.2	0.2	0	0.2	0	0
F	0.4	0	0.2	0	0	0.4	0
G	0	0	0	0.2	0	0	0.8

LSTM							
Actual	Predicted						
	A	B	C	D	E	F	G
A	0.6	0.2	0	0.2	0	0	0
B	0	1.0	0	0	0	0	0
C	0	0	1.0	0	0	0	0
D	0	0	0	1.0	0	0	0
E	0	0	0	0.6	0.4	0	0
F	0	0	0	0	0	1.0	0
G	0	0	0	0	0	0	1.0

Bi-LSTM							
Actual	Predicted						
	A	B	C	D	E	F	G
A	1.0	0	0	0	0	0	0
B	0.4	0.6	0	0	0	0	0
C	0	0	1.0	0	0	0	0
D	0	0	0	1.0	0	0	0
E	0.6	0	0	0	0.4	0	0
F	0.4	0	0	0	0	0.6	0
G	0	0	0	0	0	0	1.0

GRU							
Actual	Predicted						
	A	B	C	D	E	F	G
A	0.6	0.2	0	0.2	0	0	0
B	0	1.0	0	0	0	0	0
C	0	0	1.0	0	0	0	0
D	0	0	0	1.0	0	0	0
E	0.4	0	0	0	0.6	0	0
F	0	0	0	0	0	1.0	0
G	0	0	0	0	0	0	1.0

5. CONCLUSIONS

HAR using Wi-Fi signals is very useful in many areas, such as keeping track of elderly people and creating a smart home environment. The existing ML models require a high demand on hardware resources, which is less advantageous for mobile edge computing applications. As a result, an SNN method is proposed and this is the first CSI-based Wi-Fi signal HAR using SNNs. In this paper, the SNN model has achieved lower memory usage while maintaining high overall accuracy of 84.8073%. The proposed SNN model has saved 87% memory when compared with the Bi-LSTM model. The memory needed by the SNN model is as low as 13620kb, while the Bi-LSTM model needed 108520kb of memory. More than 70% of memory saving running in SNNs compared to other models demonstrates that the SNN model is the preferred computational model to exploit machine intelligence while keeping computational accuracy, thereby avoiding expensive memory access operations.

ACKNOWLEDGEMENTS

The authors acknowledge the technical and financial support by Universiti Teknikal Malaysia Melaka (UTeM) and the Ministry of Higher Education, Malaysia, under the research grant no. FRGS/1/2020/FKEKK-CETRI/F00421.

REFERENCES

[1] C. Jobanputra, J. Bavishi and N. Doshi, "Human Activity Recognition: A Survey," Procedia-Computer Science, vol. 155, no. 2018, pp. 698–703, DOI: 10.1016/j.procs.2019.08.100, 2019.

[2] S. Yousefi, H. Narui, S. Dayal, S. Ermon and S. Valaee, "A Survey on Behaviour Recognition Using WiFi Channel State Information," IEEE Communications Magazine, vol. 55, no. 10, pp. 98–104, 2017.

[3] J. Yang, Y. Liu, Z. Liu, Y. Wu, T. Li and Y. Yang, "A Framework for Human Activity Recognition Based on WiFi CSI Signal Enhancement," International Journal of Antennas and Propagation, vol. 2021, DOI: 10.1155/2021/6654752, 2021.

[4] Z. Chen, L. Zhang, C. Jiang, Z. Cao and W. Cui, "WiFi CSI Based Passive Human Activity Recognition Using Attention Based BLSTM," IEEE Transactions on Mobile Computing, vol. 18, no. 11, pp. 2714–2724, DOI: 10.1109/TMC.2018.2878233, 2019.

[5] R. N. S. Husna, A. R. Syafeeza, N. A. Hamid, Y. C. Wong and R. A. Raihan, "Functional Magnetic Resonance Imaging for Autism Spectrum Disorder Detection Using Deep Learning," Jurnal Teknologi, vol. 83, no. 3, pp. 45–52, DOI: 10.11113/JURNALTEKNOLOGI.V83.16389, 2021.

[6] D. Azzouz and S. Mazouzi, "A Hyper-surface-based Modeling and Correction of Bias Field in MR Images," Jordanian Jour. of Computers and Information Technology (JJCIT), vol. 7, no. 3, p. 223, 2021.

[7] Y. C. Wong, L. J. Choi, R. S. S. Singh, H. Zhang and A. R. Syafeeza, "Deep Learning-based Racing BIB Number Detection and Recognition," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 3, pp. 181–194, DOI: 10.5455/JJCIT.71-1562747728, Dec. 2019.

[8] D. Singh et al., "Human Activity Recognition Using Recurrent Neural Networks," Proc. of the Int. Cross-domain Conf. for Machine Learning and Knowledge Extraction (CD-MAKE 2017), pp. 267–274, vol. 10410 LNCS, DOI: 10.1007/978-3-319-66808-6_18, 2017.

[9] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang and H. Liu, "E-eyes: Device-free Location-oriented Activity Identification Using Fine-grained WiFi Signatures," Proceedings of the 20th Annual International Conference on Mobile Computing and Networking (MobiCom), pp. 617–628, DOI: 10.1145/2639108.2639143, 2014.

[10] W. Wang, A. X. Liu, M. Shahzad, K. Ling and S. Lu, "Device-free Human Activity Recognition Using Commercial WiFi Devices," IEEE Journal on Selected Areas in Communications, vol. 35, no. 5, pp. 1118–1131, DOI: 10.1109/JSAC.2017.2679658, 2017.

[11] Z. Wang et al., "A Survey on Human Behavior Recognition Using Channel State Information," IEEE Access, vol. 7, no. October, pp. 155986–156024, DOI: 10.1109/ACCESS.2019.2949123, 2019.

[12] S. M. Bokhari, S. Sohaib, A. R. Khan, M. Shafi and A. R. Khan, "DGRU Based Human Activity Recognition Using Channel State Information," Measurement, vol. 167, p. 108245, 2021.

[13] W. Maass, "Networks of Spiking Neurons: The Third Generation of Neural Network Models," Neural Networks, vol. 10, no. 9, pp. 1659–1671, 1997.

[14] S. Ghosh-Dastidar and H. Adeli, "Spiking Neural Networks," International Journal of Neural Systems, vol. 19, no. 4, pp. 295–308, DOI: 10.1142/S0129065709002002, Aug. 2009.

[15] H. Hazan, D. J. Saunders, H. Khan, D. Patel and D. T. Sanghavi, "BindsNET: A Machine Learning-oriented Spiking Neural Networks Library in Python," Frontiers. Neuroinformatics, vol. 12, no. December, pp. 1–18, DOI: 10.3389/fninf.2018.00089, 2018.

[16] B. Meftah, O. Lézoray, S. Chaturvedi, A. A. Khurshid and A. Benyettou, "Image Processing with Spiking Neuron Networks," Stud. Comput. Intell., vol. 427, pp. 525–544, DOI: 10.1007/978-3-642-29694-9_20, 2013.

[17] W. N. Lo and Y. C. Wong, "Spiking Neural Network for Energy Efficient Learning and Recognition," International Journal of Scientific & Technology Research, vol. 9, no. 11, pp. 166–174, 2020.

[18] M. Alawad, H. J. Yoon and G. Tourassi, "Energy Efficient Stochastic-based Deep Spiking Neural Networks for Sparse Datasets," Proc. of the IEEE Int. Conf. on Big Data (Big Data), vol. 2018-Jan., pp. 311–318, DOI: 10.1109/BigData.2017.8257939, Boston, MA, USA, 2017.

[19] T. Obo, N. Kubota and B. Hee Lee, "Localization of Human in Informationally Structured Space Based on Sensor Networks," Proc. of the IEEE International Conference on Fuzzy Systems, DOI: 10.1109/FUZZY.2010.5584888, Barcelona, Spain, 2010.

[20] A. Antonietti, C. Casellato, J. A. Garrido, E. D'Angelo and A. Pedrocchi, "Spiking Cerebellar Model with Multiple Plasticity Sites Reproduces Eye Blinking Classical Conditioning," Proc. of the 7th Int. IEEE/EMBS Conf. on Neural Eng. (NER), vol. 2015-July, pp. 296–299, Montpellier, France, 2015.

[21] A. M. George, D. Banerjee, S. Dey, A. Mukherjee and P. Balamurali, "A Reservoir-based Convolutional Spiking Neural Network for Gesture Recognition from DVS Input," Proc. of the IEEE International Joint Conference on Neural Networks (IJCNN), DOI: 10.1109/IJCNN48605.2020.9206681, Glasgow, UK, 2020.

[22] J. J. Wade, L. J. McDaid, J. A. Santos and H. M. Sayers, "SWAT: A Spiking Neural Network Training Algorithm for Classification Problems," IEEE Transactions on Neural Networks, vol. 21, no. 11, pp. 1817–1830, DOI: 10.1109/TNN.2010.2074212, 2010.

[23] A. Jeyasothy, S. Sundaram and N. Sundararajan, "SEFRON: A New Spiking Neuron Model with Time-varying Synaptic Efficacy Function for Pattern Classification," IEEE Transactions on Neural Networks Learn. Syst., vol. 30, no. 4, pp. 1231–1240, DOI: 10.1109/TNNLS.2018.2868874, 2019.

[24] T. J. Strain, L. J. McDaid, T. M. McGinnity, L. P. Maguire and H. M. Sayers, "An STDP Training Algorithm for a Spiking Neural Network with Dynamic Threshold Neurons," International Journal of Neural Systems, vol. 20, no. 6, pp. 463–480, DOI: 10.1142/S0129065710002553, 2010.

[25] GitHub, "GitHub-Hirokazu-Narui/LSTM_wifi_activity_recognition," [Online], Available: https://github .com/Hirokazu-Narui/LSTM_wifi_activity_recognition, (Accessed ON Jan. 03, 2021).

ملخص البحث:

في هذا البحث تمّ اقتراح شبكات عصبيّة ناتئة (SNNs) لتمييز الأنشطة البشرية باستخدام إشارات (واي-فاي). وتجدر الإشارة الى أن الشبكات العصبية النّاتئة مستوحاة من معالجة المعلومات في علم الأحياء التي تجري معالجتها بطريقة متوازية الى حدّ كبير. وتعمل الطريقة المقترحة على تقليل الموارد الخاصة بالمعالجة في الوقت الذي تحافظ فيه على الدّقّة من خلال استخدام نقل المعلومات بطريقةٍ سهلة لكنها متينة بالنسبة الى الضّجيج النّاجم عن الإشارات النّاتئة. كذلك تمت مقارنة أداء تمييز الأنشطة البشرية باستخدام الشّبكات العصبية النّاتئة بأداء شبكات أخرى من شبكات تعلّم الآلة (ML)، مثل نماذج LSTM و Bi-LSTM و GRU. وقد اتّضح تحقيق وَفْرٍ في استخدام الذاكرة مع الحفاظ على الدّقّة على نحوٍ متقارب لشبكات تعلّم الآلة الأخرى. فقد تمّ تحقيق وَفْرٍ نسبته 70% في استخدام الذّاكرة في الشّبكات العصبية النّاتئة. وهذا من شأنه أن يجعل من الشّبكات العصبية النّاتئة حلاً محتملاً لحساب الحواف في الثّورة الصناعية الرابعة.

NETWORK INTRUSION DETECTION SYSTEMS USING SUPERVISED MACHINE LEARNING CLASSIFICATION AND DIMENSIONALITY REDUCTION TECHNIQUES: A SYSTEMATIC REVIEW

Zein Ashi, Laila Aburashed, Mahmoud Al-Qudah and Abdallah Qusef

(Received: 21-Aug.-2021, Revised: 4-Nov.-2021, Accepted: 14-Nov.-2021)

ABSTRACT

Protecting the confidentiality, integrity and availability of cyberspace and network (NW) assets has become an increasing concern. The rapid increase in the Internet size and the presence of new computing systems (like Cloud) are creating great incentives for intruders. Therefore, security engineers have to develop new technologies to match growing threats to NWs. New and advanced technologies have emerged to create more efficient intrusion detection systems using machine learning (ML) and dimensionality reduction techniques, to help security engineers bolster more effective NW Intrusion Detection Systems (NIDSs). This systematic review provides a comprehensive review of the most recent NIDS using the supervised ML classification and dimensionality reduction techniques, it shows how the used ML classifiers, dimensionality reduction techniques and evaluating metrics have improved NIDS construction. The key point of this study is to provide up-to-date knowledge for new interested researchers.

KEYWORDS

Network intrusion detection, Machine learning, Supervised learning, Dimensionality, Systematic review.

1. INTRODUCTION

With the new development in NWs and communications, cybersecurity has become a vital requirement to defend new cyber-attacks [1]. Recently, IDSs in general and NIDSs in particular, have been increasingly used as tools to constantly monitor NW traffic and provide desired security protection against cyber-attacks [2]. The earliest IDS was produced in 1980 by Jim Anderson and since then, such systems have continuously developed and improved, to keep pace with the rapid growth in the NW and communication fields [3]. The growth of cyberspace has introduced the Big Data concept to the IDS field, in which massive volumes of data are continually generated around the Internet. Security engineers have used this Big Data with different ML techniques for further IDS improvements [1]. Supervised ML NIDS depends on pre-collected datasets to learn how to distinguish between normal and abnormal NW traffic, to be able to detect any intrusions in the future [3].

The main purpose of this systematic review is to provide a broad analysis of developments in modern supervised ML NIDSs. The core idea is to provide updated information on supervised ML NIDSs to provide a starting point for new researchers who want to explore this field. This study undertakes three main objectives to contribute to existing knowledge: (1) To conduct a systematic review of selected research papers concerned with supervised ML NIDS published during 2017 and until March 2021 in Science-Direct (Elsevier), Springer-Link (Springer) and IEEE-Explore (IEEE) libraries; (2) To review each research paper extensively and discuss its used ML classifiers, dimensionality reduction algorithms and evaluation metrics; and (3) To highlight recent trends in using such technologies for building NIDSs and various future challenges.

There are many survey papers in the literature providing reviews on NIDSs, but this study is unique in applying a systematic approach to collect more relevant research papers on NIDSs designed by supervised ML classification and dimensional minimization techniques. This study reviews the most recent research papers from the past three years, providing up-to-date knowledge for researchers.

Section 2 reviews related studies in this area to present background information about IDSs and Section 3 details IDS categorization. Section 4 explains the research methodology, followed by the application

Z. Ashi, L. Aburashed, M. Al-Qudah and Abdallah Qusef (Corresponding Author, ORCID: 0000-0003-4769-6992) are with the King Hussein School of Computing Science, Princess Sumaya University for Technology, Amman, Jordan. Emails: zai20198121@std.psut.edu.jo, lay20188108@std.psut.edu.jo, mah20208173@std.psut.edu.jo and a.qusef@psut.edu.jo

of supervised ML and dimensionality reduction techniques in Section 5. Section 6 presents the evaluation metrics. Section 7 discusses the salient findings and identified challenges, while Section 8 concludes the paper.

2. RELATED WORKS

Numerous researchers have taken an interest in NIDSs and machine learning and a variety of surveys and systematic reviews have summarized previous studies in this field. Zebari *et al.* [29] conducted a comprehensive review of dimensionality reduction techniques used in the previous IDSs. For each study, they provided some details about the algorithms used, datasets, dimensionality reduction techniques (categorized into feature selection and feature extraction) and they summarized the achieved results. Although they analyzed recent studies (between 2018 and 2020), they did not follow a systematic approach, unlike the current study (which provides a systematic approach to collect the analyzed research papers, to make the data collection more accurate and comprehensive).

Martins *et al.* [56] presented a systematic review of ML-based systems to detect intrusion and malware scenarios. They reviewed 20 research papers from multiple scientific e-libraries and compared them based on attack techniques, used algorithms, datasets, evaluation metrics and their results. The limitation of their study was that they did not provide details about their systematic approach and did not mention whether the analyzed studies were recent or not. In our study, we provide a detailed description for our followed approach.

Ahmad *et al.* [1] reviewed recent studies (from 2017 to 2020) that generally used machine learning and deep-learning techniques. Their review was notable in identifying the strengths and weaknesses for each reviewed study, which we have also applied in the current work.

Some studies that introduced the software system IDS were analyzed by Ramaki *et al.* [57]. They limited their study to ML techniques that used “Hidden Markov” models and did not provide a systematic approach for collecting research papers for analysis. This systematic review spans a wider domain, including NIDSs based on supervised ML techniques. Gonzalez *et al.* [58] developed a method for improving security inside secure military self-protected software and comprehensively analyzed software present position and potential responses to threats. Their method consisted of three stages: user detection, analysis of current situation and reactive action. The detection phase consists of analyzing location, timing at present location and identifying user type (friend or foe). The analysis phase entails determining whether self-protected software should be present at the current site, predicting future locations and analyzing the level of hazard at the current location. Analytical results showed that self-protected software that incorporates user detection provides higher protection than self-protected software that does not contain such detection capability.

Nassif *et al.* [59] analyzed ML approaches utilized to detect cloud system attacks with a detailed systematic review for 63 relevant research papers from 2004 to 2021. For each study, they identified the related security threats, ML techniques used and evaluation metrics’ results. This systematic review provides more comparison criteria between the analyzed research papers.

3. IDS CATEGORIZATION

An IDS system identifies abnormal events by constantly monitoring network traffic, keeping a network log and alerting the network administrator in the event of any intrusion. IDS copies the network traffic for read-only analysis, to detect any suspicious events and notify the administrators about what is going on (to take manual responsive actions). IDS is implemented outbound of the network line, without affecting the network data flow [4]. IDS can be categorized based on its monitoring environment and detection approach. Types of IDS according to monitoring environment are host-based (HIDS), NW-based (NIDS) and hybrid IDS, while according to their detection approaches they can be classified as signature-based, anomaly-based and hybrid [5] (Figure 1).

3.1 IDSs According to Monitoring Environment

HIDS operates in a local machine that detects local abnormal behaviours; any changes to the host registry, unauthorized access attempts or attacks cannot be detected by firewalls [5]. IDS is considered

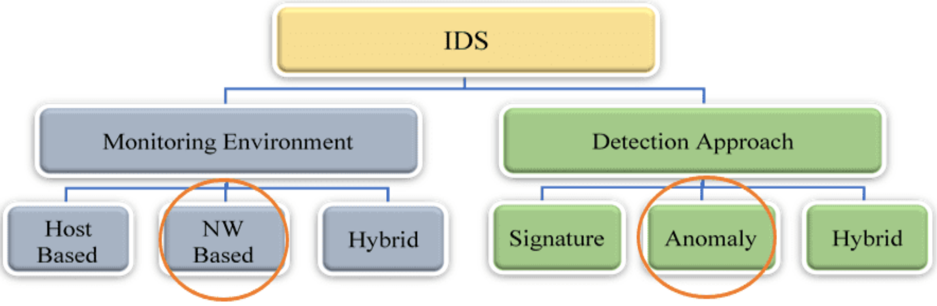


Figure 1. IDS categories and types.

a reliable system, because it analyzes the log files so that it can efficiently determine whether an attack is active or not [6]. NIDS operates in an NW node used to monitor and analyze network traffic on a single network node to detect any abnormal traffic [7]. Some NIDSs are created by analyzing the payload of an NW packet (packet level) or analyzing only that packet's header (flow level) [7]. Hybrid IDS integrates HIDS and NIDS in an effective way [2].

3.2 IDSs According to Detection Approach

Signature-based IDS (also called "misuse detection IDS" or "knowledge-based IDS") uses a blacklist of predefined intrusions and attacks. When any intrusion in the blacklist occurs, this IDS can detects it accurately, with no false alarms [8]. The disadvantages of this type are the required storage size and that it cannot detect any novel predefined intrusion on its blacklist. This blacklist requires constant updates to be able to detect any new intrusions [2].

Anomaly-based IDS (also called "behaviour-based IDS") uses the definition of the normal NW traffic and any deviation of that normality is detected as an intrusion. It compares the actual NW traffic with the predefined characteristics of normal traffic to detect any intrusions [9]. It can detect any novel intrusion, but it suffers high false alarms, as it is difficult to define uniform traffic among all NWs [2].

Hybrid IDS efficiently combines signature-based and anomaly-based approaches, to detect known attacks in the blacklist while simultaneously detecting new ones [2]. Anomaly-based NIDS is the main focus of this study, developed using supervised, unsupervised or reinforcing ML techniques [7].

4. Research Methodology

The methodology used in this study is adapted from [10], collecting papers on NIDSs built with ML techniques published from 2017 to March 2021 in the Science-Direct (Elsevier), Springer-Link (Springer) and IEEE-Explore (IEEE) libraries. Search keywords have been used to achieve results related to the search questions, according to the inclusion and exclusion criteria phases (Figure 2).

4.1 Research Questions

RQ1. What are the proposed supervised ML classification and dimensionality reduction techniques used to build the NIDS?

This question describes the supervised ML classification and dimensionality reduction techniques which have been used in previous studies to build NIDS against cyber-attacks, to investigate the popular techniques used for more enhancement in this domain.

RQ2. What are the evaluation metrics used to evaluate the proposed NIDS?

RQ3. What are the best supervised ML classification and dimensionality reduction techniques used to build the NIDS?

The main purpose of NIDSs is to detect intrusions in real time, with high sensitivity and low false alarms. This question explores whether the built NIDS provides a noticeable enhancement in this domain, as well as to identify techniques that enhance NIDS sensitivity without increasing processing overhead or affecting real time detection.

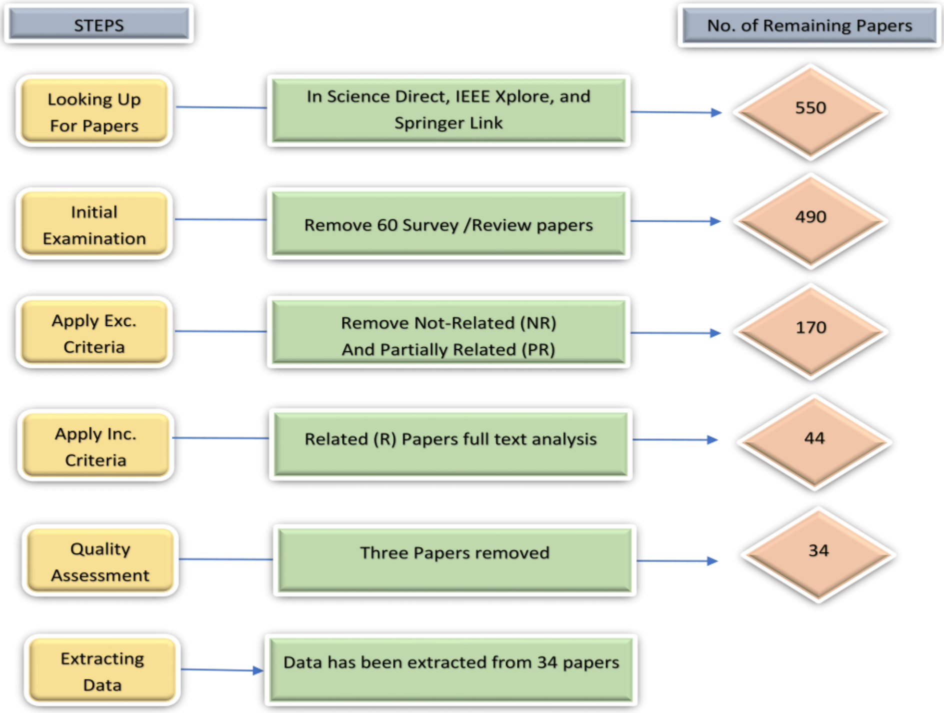

STEPS
No. of Remaining Papers
Looking Up For Papers
In Science Direct, IEEE Xplore, and Springer Link
550
Initial Examination
Remove 60 Survey /Review papers
490
Apply Exc. Criteria
Remove Not-Related (NR) And Partially Related (PR)
170
Apply Inc. Criteria
Related (R) Papers full text analysis
44
Quality Assessment
Three Papers removed
34
Extracting Data
Data has been extracted from 34 papers

Figure 2. Flow process of inclusion and exclusion of papers.

4.2 Research Process

4.2.1 E-Library Search Phase

Three e-libraries were selected to conduct this systematic review; Science-Direct (Elsevier); Springer-Link (Springer); and IEEE-Explore (IEEE). All are Scopus-indexed, constituting the biggest database of peer-reviewed research papers. The search was conducted directly in the selected e-libraries during 2017-2021, using the search keywords shown in Table 1.

Table 1. Keyword searching.

Keywords	Close Keywords	AND/OR Combination
NIDS	Network Intrusion Detection System	NIDS AND machine learning NIDS AND dimensionality reduction NIDS AND machine learning AND (feature selection OR feature extraction) NIDS AND machine learning AND dimensionality reduction
Machine Learning	Artificial Intelligence Techniques	
Feature Selection	Optimization Algorithms	
Feature Transformation	Feature Extraction	
Dimensionality Reduction	------	

4.2.2 Selecting Pre-processing Phase

The initial search process using the chosen keywords resulted in many initial hits, the titles of which were then cross-checked with the research questions and inclusion and exclusion criteria, to eliminate 550 papers not directly related to machine learning-based NIDSs. All authors independently scan the resulted 550 research papers (titles and abstracts). The resulting group was categorized into unrelated research papers (NR), partially related (PR) and related research papers (R). In this stage, the process of exclusion was performed on research papers the abstracts of which did not mention any techniques for supervised ML NIDS classification, feature selection, feature transformation and dimensionality reduction. A total of 170 research papers were marked NR and PR by the first review and then another review was performed on the unmarked set to judge 44 R papers. Further reviewing, to avoid any bias, was conducted. All reviewers met later to verify the exclusion of research papers deemed NR and PR. The final set of research papers was approved by all reviewers as related to this study, as shown in Figure 2. A total of 34 research papers remained and finally, the quality assessment criteria were followed again during the final full-text analysis for a total of 34 research papers.

4.2.3 Quality Assessment

The quality assessment eliminated bias in research papers selection and ensured that clear criteria were used to determine the quality of the selected research papers, as shown in Table 2. Scores for quality relied upon the following criteria: score 1 indicates that a research paper explicitly follows the assessment criteria, score 0 indicates that a research paper no doubt did not meet the criteria and research papers suspected to be related that necessitated more analysis and clarification or which did not fully meet the criteria were scored 0.5. Section 5 analyzes the papers that achieved over 50% in the quality assessment in detail.

Table 2. Quality assessment criteria.

	Assessment Question	**Assessment**
Q1	Does the paper topic cover NIDS domain?	1/ Zero/ 0.5
Q2	Does the paper use “machine learning techniques” or “machine learning and optimization techniques” or “machine learning and dimensionality reduction techniques”?	1/ Zero/ 0.5
Q3	Is the proposed methodology fully defined?	1/ Zero/ 0.5
Q4	Are the research results verified by clearly defined evaluation metrics?	1/ Zero/ 0.5

4.2.4 Information Extraction

The research questions require extracting information from the selected research papers, such as the use of: ML classification algorithms; dimensionality reduction techniques; and evaluation metrics and their results.

5. Supervised ML and Dimensionality Reduction Techniques

Answering RQ1 requires a complete analysis of the most popular supervised ML techniques (their implementation and algorithms) used to build NIDSs and detailed analysis of the dimensionality reduction techniques used.

5.1 Building Supervised ML NIDSs

Supervised ML provides an intelligence technique to extract patterns from previously labelled datasets [11], learning from previous datasets to predict future values [12]. Studies built NIDSs through several phases, including data pre-processing, training and testing and evaluation.

5.1.1 Data Pre-processing Phase

Dataset intensive care is required in supervised ML NIDSs to achieve the highest prediction accuracy rate and the most efficient performance in real-time intrusion detection; higher data quality indicates more NIDS efficiency [1]. Data pre-processing stages depend on dataset and ML algorithm requirements and researcher experience [1], [13]-[15]. In dataset cleaning, all duplicated or missing values are handled; duplicate values are deleted and rows with missing values may be deleted or filled with median, mean or most frequent corresponding values. Tables 3-5 show the research paper results for the ScienceDirect, IEEE and Springer-Link databases (respectively).

Table 3. ScienceDirect research paper results.

Research Paper	**Year**	**Q1**	**Q2**	**Q3**	**Q4**	**Total of 4**	**Percentage**
Nazir and Khan [32]	2021	1	1	1	0.5	3.5	87.5%
Mohammadi et al. [38]	2019	1	1	0.5	1	3.5	87.5%
Mazini *et al.* [33]	2019	1	1	0.5	1	3.5	87.5%
Alzahrani *et al.* [4]	2019	0.5	1	1	1	3.5	87.5%
Aljawarneh *et al.* [19]	2017	1	1	1	1	4	100%
Verma and Ranga [54]	2018	1	0.5	1	0.5	3	75%
Dwivedi *et al.* [39]	2020	0	1	1	1	3	75%
Shekhawat *et al.* [8]	2019	0.5	1	1	0.5	3	75%
Dahiya and Srivastava [10]	2018	1	1	1	1	4	100%
Kanimozhi and Jacob [40]	2020	1	1	1	0.5	3.5	87.5%
Hamamoto *et al.* [35]	2019	1	1	0.5	1	3.5	87.5%
Torres *et al.* [36]	2021	1	1	1	1	4	100%

Table 4. IEEE research paper results.

Research Paper	Year	Q1	Q2	Q3	Q4	Total of 4	Percentage
Vijayanand and Devaraj [18]	2020	1	1	1	1	4	100%
Stiawan *et al.* [16]	2020	1	1	1	1	4	100%
Jiang *et al.* [41]	2019	0.5	1	0.5	1	3	75%
Xue and Wu [17]	2020	1	1	1	0	3	75%
Ding *et al.* [42]	2020	1	1	1	0.5	3.5	87.5%
Nagaraja *et al.* [43]	2020	1	1	1	1	4	100%
Chang *et al.* [25]	2017	0.5	1	0.5	1	3	75%
Matel *et al.* [26]	2019	1	1	0.5	1	3.5	87.5%
Sun *et al.* [27]	2018	0.5	1	1	1	3.5	87.5%
Sakr *et al.* [34]	2019	1	1	1	1	4	100%

Table 5. Springer-Link research paper results.

Research Paper	Year	Q1	Q2	Q3	Q4	Total of 4	Percentage
Ghazy *et al.* [44]	2018	0.5	0.5	1	1	3	75%
Kunhare *et al.* [45]	2020	0.5	1	1	0.5	3	75%
Kasongo and Sun [46]	2020	0.5	1	1	0.5	3	75%
Bindra and Sood [47]	2019	0.5	1	1	1	3.5	87.5%
Rajadurai and Gandhi [48]	2020	0.5	1	1	0.5	3	75%
Alamiedy *et al.* [49]	2019	0.5	0.5	1	1	3	75%
Zhu and Zheng [50]	2019	0.5	0.5	1	1	3	75%
Sebbar *et al.* [51]	2020	0.5	1	1	1	3.5	87.5%
Thakur and Kumar [52]	2020	0.5	1	1	1	3.5	87.5%
Abhale and Manivannan [53]	2020	0.5	1	1	1	3.5	87.5%
Verma and Ranga [54]	2019	1	1	1	1	4	100%
Moon *et al.* [55]	2017	0.5	0.5	1	1	3	75%

All string values are transformed into numeric values, to be in a suitable format for the classification algorithm. For feature selection, unnecessary features are dropped either manually [14] or automatically (using dimensionality reduction techniques, as explained below). Some ML algorithms require normalization (data scaling) to ensure a uniform range between values (e.g. K-NN algorithm). Data splitting is applied by splitting datasets' columns and rows: columns are split into X, comprising all columns with independent variables; and Y is the column of the dependent variable that classifies rows (normal or abnormal traffic), called the "label column," which is the key data classification element in supervised learning.

5.1.2 Training Phase

To make the supervised ML algorithm goes through the learning experiment; it needs a partition of the dataset, called the training set [16]. The supervised ML classifier is fed with the independent (X) and dependent (Y) variables in the training set, to be able to predict Y values on its own in the future [12]. The size of the training set is important to help the ML algorithm learn efficiently with a highly accurate prediction rate in the least amount of time [17]. Most commonly, the training set consists of 70-80% of the original dataset, with the remainder for the testing set [16].

5.1.3 Testing and Evaluation Phase

The testing set is fed to the trained ML algorithm with only the X values, to test its ability to predict Y values. Predicted and actual Y values are then compared using evaluation metrics [16] (Section 6), to measure the trained ML algorithm's prediction ability and test its suitability with real NW traffic [18]. Supervised ML classification algorithms thus use independent (X) and dependent (Y) values and learn how they relate to each other in the training phase, then the trained algorithm is provided X values to evaluate performance in predicting Y values in the testing phase. Finally, the predicted results are evaluated using evaluation metrics [12].

5.2 Supervised ML Classifiers

5.2.1 Decision Tree (DT)

DT algorithm represents the feature values as nodes in a hierarchal tree, to divide the classification problems into sub-sets [19]. DT consists of nodes that represent features, branches represent roles and leaves represent a class value (e.g. malicious or normal traffic) [12]. DT algorithm forecasts class values based on learning decision rules extracted from features [20]. DT algorithm may be implemented by C4.5 (J48), an open-source Java implementation [21]; ID3, an extension of the former and REP-Tree [22], another open-source implementation for DT [23]. Aljawarneh *et al.* [19] proposed anomaly ML NIDS using REPTree classifier, pre-processed with feature selection using Vote scheme, training and testing phases. Their proposed NIDS obtained highly accurate results for detecting NW intrusions.

5.2.2 K-Nearest Neighbour (K-NN)

K-NN algorithm represents the given training data as neighbour points in a graph and assigns the new data point to the nearest specified K neighbour points. Figure 3 shows K-NN performance with K= 5. The distance between the new data point (X1, Y2) and any other neighbour point (X2, Y2) is calculated using Manhattan (Eq. 1) or Euclidean (Eq. 2) equations [1]. After calculating the distances, the new data point is classified according to the closest points [12].

$$|X2 - X1| + |Y2 - Y1| \qquad (1)$$

$$\sqrt{((X2 - X1)^2 + (Y2 - Y1)^2)} \qquad (2)$$

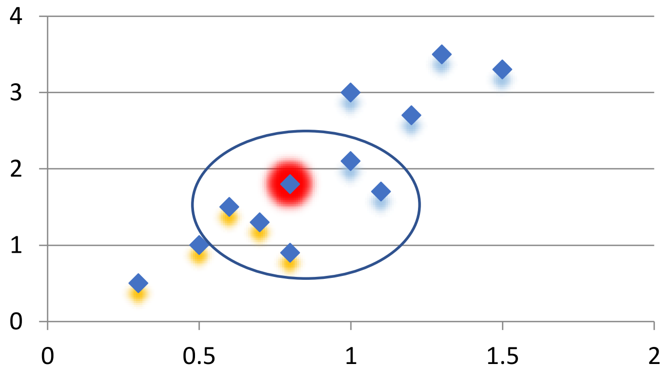


Figure 3. K-NN graph.

Verma and Ranga [7] used CIDDS-001 dataset to build their ML NIDS, labelled based on flow level and having 14 features; only 12 features were manually selected for the supervised training phase. Using the Weka tool, a K-NN classifier was implemented for multi-class classification. During the experiment, several K-NN iterations were conducted with different values for the number of neighbours (NN), identifying the best classification performance for NN = 2, with accuracy of 100% and no false positives.

5.2.3 Naïve Bayes (NB)

It is one of the most common machine learning classifiers in general. NB classification is based on Bayes' theorem [24]. NB measures the likelihood of a given prediction based on available features, as each feature independently contributes to predicting unknown data [2].

5.2.4 Support Vector Machine (SVM)

SVM algorithm combines statistical theory with supervised learning by finding the best way to split data into two classes by adding a boundary between them, regardless of whether the data can be divided linearly or not [8]. Essentially, this algorithm finds the best possible boundaries in the data collection to distinguish between classes [24].

5.2.5 ML Ensemble Methods

Ensemble supervised ML classifiers are integrated to solve a complex problem and increase accuracy by pooling individual classifiers' strengths [20]. For example, some algorithms may perform well in detecting a certain type of attack, but poorly in detecting others, thus combinations form a stronger classifier [25]. Several ML techniques (Random Forest (RF), Ada-Boost, XG-Boost, …etc.) use ensemble method to enhance performance. RF classifier integrates many DT classifiers, instead of depending on a single decision tree, taking predictions from each tree to forecast final performance based on the majority vote of predictions [17]. AdaBoost improves the performance of binary classifiers by employing an iterative approach, learning from the errors of weak classifiers and transforming them into strong ones [26]. XG-Boost consists of multiple DTs to solve a wide range of data-mining problems quickly and accurately [27].

5.3 Dimensionality Reduction Techniques

Supervised ML and Big Data mining techniques are very complex and require high computational costs due to the voluminous data processed [1]. Real-time detection and accurate detection rates in NIDSs are major concerns in relation to the "dimensional curse," referring to ML model complexity due to a large number of both necessary and unnecessary features, with high dimensionality [28]. Dimensionality reduction techniques seek to reduce the number of features processed by selecting or extracting only relevant ones from the feature set, excluding irrelevant, noisy or redundant ones [29]. For dimensionality reduction, several algorithms reduce feature space either by removing features that do not provide important information or extracting relationships between available features to produce less space with new features [30]. This reduces complexity, increases understanding of data, facilitates easier analysis, improves visualization and reduces processing costs and storage space requirements [6], [29]. The ML model learning process is thus enhanced, resulting in higher performance and prediction accuracy rates, providing real-time prediction results [30]. Dimensionality reduction can be conducted by two approaches.

First, feature transformation/ extraction transforms the available features into more beneficial ones using optimization algorithms [28]. The most common methods used to conduct feature extraction are Principal Component Analysis (PCA), Multi-Dimensional Scaling (MDS), Isometric Mapping (ISOMAP), Locally Linear Embedding (LLE), Linear Discriminant Analysis (LDA), Canonical-Correlation-Analysis (CCA), Latent Semantic Indexing (LSI) and clustering methods [29].

Second, feature selection approach selects features according to their relevance and effectiveness related to the classification problem [29], without changing representation [31].

Researchers can choose one of four methods to implement their feature selection approach, which differ in how the ML algorithm functions [31] (Figure 4), as discussed below.

5.3.1 Filter Method

Weights are assigned to features to determine their relevance and essence (dependency, consistency, …etc.) using statistical standards, without involving the ML algorithm [29]. Depending on the assigned weights, features are either discarded or retained [31]. Filter method has been found to outperform other feature selection methods, with less computational costs, more scalability in high-dimensional datasets and more efficiency [29], [32]. Its drawbacks are that it does not integrate between the selected subset and the ML algorithm [29] and it is only suited to independent features [32].

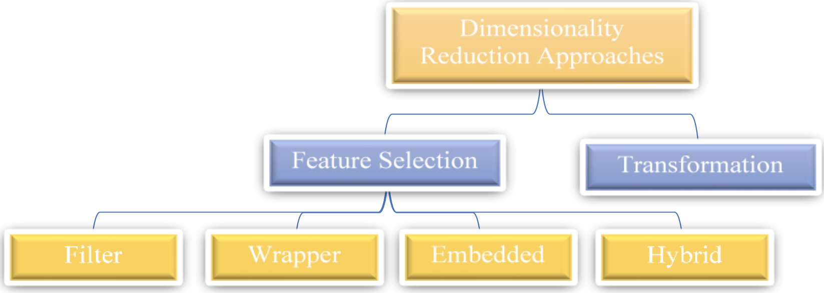


Figure 4. Dimensionality reduction techniques.

5.3.2 Wrapper Method

Wrapping creates an interaction between the ML algorithm later used for classification and each selected feature subset. The ML algorithm is used with each subset designated as a black box, to evaluate prediction accuracy and determine which subset has the fewest errors [30]. It is thus accurate and efficient [31], but is time-consuming, as selected subsets work only with particular ML algorithms, which may cause over-fitting, as well as being expensive [29].

5.3.3 Embedded Method

Embedding feature selection with the ML algorithm assigns weights independently and the highly weighted features are recursively used to construct subsets until finding the optimal one; its prediction accuracy outperforms others with the ML algorithm [30], [31]. The embedded method reduces the computational cost and the possibility of over-fitting [29].

5.3.4 Hybrid Method

The hybrid combination of filter and wrapper methods is the most commonly used solution, accruing the constituent advantages to achieve better performance [29]. This systematic review noticed that adopted dimensionality reduction techniques vary according to the research paper problem. In some problems, feature selection was forbidden, as removing features from the dataset would be misleading. Others preferred feature selection techniques, to keep meaningful original features and shorten the dimensionality reduction techniques within the selected features. Mazini *et al.* [33] proposed hybrid anomaly NIDS to detect attacks that threaten network activities. They mentioned that data-mining techniques were implemented to get rid of imbalanced database disadvantages and the complicity of feature values. Furthermore, to reach the best performance of the AdaBoost classification algorithm, they used the Artificial Bee Colony algorithm (the wrapper method) for feature selection. Selecting the most significant features to learn the classifier increases accuracy detection rate and reduces false alarms.

5.4 RQ1 Analysis and Results

Answering RQ1 entails understanding ML techniques, the most commonly used supervised ML classifiers and dimensionality reduction techniques, as summarized in Tables 6-8. Some of the selected research papers used ML classification algorithms to build their NIDS, while others additionally used dimensionality reduction technique, to improve prediction results and increase NIDS sensitivity and accuracy.

Table 6. ML methods and dimensionality reduction techniques employed – ScienceDirect studies.

Research Paper	**Training ML Algorithms**	**Feature Selection**			**FT**	**Dataset**
		Fil	Wra	Emb		
Nazir and Khan [32]	RF		✓			UNSW-NB15
Mohammadi *et al.* [38]	DT, Least Square SVM	✓	✓			KDD CUP 99
Mazini *et al.* [33]	Adaboost		✓			NSL-KDD, ISCXIDS2012
Alzahrani *et al.* [4]	SVM		✓			NSL-KDD
Aljawarneh *et al.* [19]	RF, J48, AdaBoost, NB	✓				NSL-KDD
Verma and Ranga [54]	K-NN					CIDDS-001
Dwivedi *et al.* [39]	SVM	✓	✓			ISCX 2012, NSL-KDD, CIC-IDS2017
Shekhawat *et al.* [8]	RF, SVM, XG-boost					CTU-13, Malware Capture Facility Project dataset
Dahiya and Srivastava [10]	RF, REP TREE, NB				✓	UNSW-NB15
Kanimozhi and Jacob [40]	RF, SVM, NB, K-NN, AdaBoost with DT		✓			CSE-CIC-IDS2018
Hamamoto *et al.* [35]	RF					Private dataset
Torres *et al.* [36]	RF	✓	✓			Private dataset

Fil: filter; Wra: wrapper; Emb: embedded; FT: feature transformation.

Table 7. ML methods and dimensionality reduction techniques employed – IEEE studies.

Research Paper	Training ML Algorithms	Feature Selection			FT	Dataset
		Fil	Wra	Emb		
Vijayanand and Devaraj [18]	SVM, RF				✓	CICIDS2017, ADFA-LD
Stiawan *et al.* [16]	J48			✓		ITD-UTM
Jiang *et al.* [41]	XG-Boost, RF				✓	KDD99, NSL-KDD
Xue and Wu [17]	SVM, XG-Boost					Private Dataset
Ding *et al.* [42]	SVM, NB				✓	UNSW_NB15
Nagaraja *et al.* [43]	J48	✓				NSL-KDD
Chang *et al.* [25]	SVM, RF				✓	KDD99
Matel *et al.* [26]	SVM		✓			DARPA KDD CUP 99
Sun *et al.* [27]	SVM			✓		KDD CUP 99
Sakr *et al.* [34]	SVM				✓	NSL-KDD

Filt: filter; Wra: wrapper; Emb: embedded; FT: feature transformation.

Table 8. ML methods and dimensionality reduction techniques employed – Springer-Link studies.

Research Paper	Training ML Algorithms	Feature Selection			FT	Dataset
		Fil	Wra	Emb		
Ghazy *et al.* [44]	RF		✓		✓	NSL-KDD
Kunhare *et al.* [45]	RF			✓		NSL-KDD
Kasongo and Sun [46]	XG-Boost- DT	✓				UNSW-NB15
Bindra and Sood [47]	RF				✓	CIC IDS 2017
Rajadurai and Gandhi [48]	Ensemble Gradient descent, RF			✓		NSL-KDD
Alamiedy *et al.* [49]	RF		✓			NSL–KDD
Zhu and Zheng [50]	SVM					Private dataset
Sebbar *et al.* [51]	RF	✓				Private dataset
Thakur and Kumar [52]	RF	✓	✓			Private Dataset
Abhale and Manivannan [53]	SVM					NSL-KDD
Verma and Ranga [54]	DT				✓	RPL-NIDDS17
Moon *et al.* [55]	DT					Private dataset

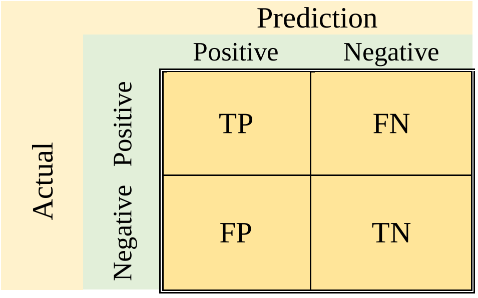
Filt: filter; Wra: wrapper; Emb: embedded; FT: feature transformation.

6. Evaluation Metrics

Answering RQ2 and RQ3 requires a complete analysis of evaluation metrics used to evaluate the proposed NIDS in each research paper.

6.1 Evaluation Metrics

During the building of any ML model, particularly in the testing phase, many metrics are used to evaluate performance [7], [27], [32]. Most of these measures are derived from the confusion matrix, which consists of two columns displaying predicted values and two rows displaying the actual values. In NIDS, predicted or actual values are positive if NW traffic is positive or negative if normal, as shown in Figure 5 [1].



		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Figure 5. Confusion matrix.

Each intersection between the columns and rows contains the following values [7]: True Positives (TP): the number of values predicted as attacks that are attacks; False Negatives (FN): the number values predicted as normal traffic that are attacks; False Positives (FP): the number of values predicted as attacks that are normal traffic; and True Negatives (FN): the number of values predicted as normal that are normal traffic. The evaluation metrics from the confusion matrix used to evaluate the proposed NIDS varied between those discussed below.

Accuracy Rate (AR) and Error Rate (ER) recognize intrusions, indicated by the ratio of correctly predicted values (TP and TN) to all other values (Eq. 3.a) [35]. ER is calculated depending on the AR, as shown in Eq. 3.b.

$$Accuracy\ rate\ (AR) = \frac{TP+TN}{TP+FN+FP+TN} \tag{3.a}$$

$$Error\ rate\ (ER) = 100 - AR \tag{3.b}$$

Recall Value (Re-V) (detection rate) measures NIDS sensitivity [12], [36]. It is the ratio of correctly predicted values as attacks (TP) to all other values that are in fact attacks (Eq. 4) [1].

$$Recall\ value = \frac{TP}{TP+FN} \tag{4}$$

Precision value (PV) indicates the reliability of the NIDS [12]. It is the ratio of correctly predicted values as attacks (TP) to all other predicted values as attacks (Eq. 5) [1].

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

False alarm rate is the ratio of incorrectly predicted values as attacks (FP) to all other normal values (Eq. 6) [1].

$$False\ alarm\ rate\ = \frac{FP}{FP+TN} \tag{6}$$

True Negative Rate (TNR) measures NIDS specificity [13]; it is the ratio of correctly predicted values as normal traffic (TN) to all other normal values (Eq. 7) [1].

$$True\ negative\ rate\ = \frac{TN}{FP+TN} \tag{7}$$

F Measure (F1) represents NIDS accuracy in terms of precision and recall values (Eq. 8) [13].

$$F\ Measure\ = 2\left(\frac{Precession\ X\ Recall}{Precession + Recall}\right) \tag{8}$$

Receiver Operating Character – Area Under the Curve (Roc-Auc) rate is the area under the curve that virtualizes the relation between the True Positive Rate (TPR) and False Positive Rate (FPR) for every confusion matrix, resulting from every threshold in binary classification [8], [37]. The higher the TPR and the lower the FPR, the higher the Roc-Auc score [13]. For further evaluation of the NIDS performance, researchers calculate the time consumed in the training and testing phases (Tr-T and Ts-T, respectively), so the NIDS is lightweight and easy to install and provides real-time detection of NW intrusions [12]. Tables 9-11 show that most researchers relied on AR, DR and FPR to evaluate their proposed NIDS, so these metrics are considered to answer RQ3 in the next section.

Table 9. Results of evaluation metrics for each research paper – ScienceDirect.

Research Paper	**Evaluation Metrics**							
	AR %	Re-V/DR %	PV	FPR%	F1	Auc-Roc	Tr-T (sec)	Ts-T (sec)
Nazir and Khan [32]	83.12			3.7				
Mohammadi *et al.* [38]	95.03	95.23		1.65				
Mazini *et al.* [33]	98.9	99.61		1				
Alzahrani *et al.* [4]		99.21					0.6385	
Aljawarneh *et al.* [19]	99.81			0.3				
Verma and Ranga [54]	100							
Dwivedi *et al.* [39]	99.63	99.71		8.5				
Shekhawat *et al.* [8]	100					99.88		
Dahiya and Srivastava [10]	93.56	0.843	84.2	2.1		96.1	5.74	
Kanimozhi and Jacob [40]	99.96	99.88	99.96		99.92	99.88		
Hamamoto *et al.* [35]	96.53			0.56				
Torres *et al.* [36]	93			7.5	93	97		

Table 10. Results of evaluation metrics for each research paper – IEEE.

Research Paper	**Evaluation Metrics**							
	AR %	Re-V/DR %	PV	FPR%	F1	Auc-Roc	Tr-T (sec)	Ts-T (sec)
Vijayanand *et al.* [18]	95.91			4			4959	4960
Stiawan *et al.* [16]	99.87						0.996	0.830
Jiang *et al.* [41]	94	0.75	81.0		0.71			
Xue and Wu [17]	99.68							
Ding *et al.* [42]	99.41	99.64	99.04	0.0077			144.3	2.39
Nagaraja *et al.* [43]	99.44	87.6	92.5			.981		
Chang *et al.* [25]	93			3				
Matel *et al.* [26]	96.122			3.878				
Sun *et al.* [27]	91.686							
Sakr *et al.* [34]	98.04	97.55		1.4			5.16	10.18

Table 11. Results of evaluation metrics for each research paper – Springer-Link.

Research Paper	**Evaluation Metrics**							
	AR %	Re-V/DR %	PV	FPR%	F1	Auc-Roc	Tr-T (sec)	Ts-T (sec)
Ghazy *et al.* [44]			0.99	0.001				
Kunhare *et al.* [45]	99.32	99.26	99.37	0.62	99.31			
Kasongo and Sun [46]	90.85		80.33		87.42			
Bindra and Sood [47]	96					0.99		
Rajadurai and Gandhi [48]	91.06	99.77						
Alamiedy *et al.* [49]	93.64							
Zhu and Zheng [50]	99.31							
Sebbar *et al.* [51]	97.4	98.9	94.7		96.7	99		
Thakur and Kumar [52]	99.1							
Abhale and Manivannan [53]	84.0	0.86	0.87	0.8	0.87	0.85		
Verma and Ranga [54]	94.07			3.80				
Moon *et al.* [55]	89.1	84.7						

6.2 RQ2 and RQ3 Results

For RQ2, after analyzing the used evaluation metrics to determine the highest evaluation results achieved by the selected research papers, Tables 12-14 show that most researchers relied on AR, DR and FPR to evaluate their proposed NIDS, so these metrics are considered to answer RQ3. Determining the best ML and dimensionality reduction techniques used to build NIDS requires summarizing all techniques used in the selected research papers, showing their AR, DR and FPR (Tables 12-14).

Table 12. Summary of all techniques used in the selected research papers – ScienceDirect.

Research Paper	**ML Algorithms**	**Feature Selection**			**FT**	**Evolution Metric**		
		Fil	Wra	Emb		AR	DR	FPR
Nazir and Khan [32]	RF	✓	✓			83.12		3.7
Mohammadi *et al.*	DT	✓	✓			95.03	95.23	1.65
Mazini *et al.* [33]	Ada-Boost		✓			98.9	99.61	1
Alzahrani *et al.* [4]	SVM		✓				99.21	
Aljawarneh *et al.* [19]	RF, J48, Ada-Boost, NB	✓				99.81		0.3
Verma and Ranga [54]	K-NN					100		
Dwivedi *et al.* [39]	SVM	✓	✓			99.63	99.71	8.5
Shekhawat *et al.* [8]	RF, SVM, XG-boost					100		
Dahiya and Srivastava	RF, REP TREE, NB				✓	93.56	84.3	2.1
Kanimozhi and Jacob	RF, SVM, NB,		✓			99.96	99.88	
Hamamoto *et al.* [35]	RF					93		7.5
Torres *et al.* [36]	RF	✓	✓			98.92		

Fil: filter; Wra: wrapper; Emb: embedded; FT: feature transformation.

The ML algorithm (marked in red) indicates its adoption in the research paper, achieving the best evaluation results. The tables also show that most researchers relied on AR, DR and FPR to evaluate their proposed NIDS, so these metrics are considered to answer RQ3.

Table 13. Summary of all techniques used in the selected research papers – IEEE.

Research Paper	**ML Algorithms**	**Feature Selection**			**FT**	**Evolution Metric**		
		Fil	Wra	Emb		AR	DR	FPR
Vijayanand *et al.* [18]	SVM, RF				✓	95.91		4
Stiawan *et al.* [16]	J48			✓		99.87		
Jiang *et al.* [41]	XGBoost, RF				✓	94	0.75	
Xue and Wu [17]	SVM, XGBoost				✓	99.68		
Ding *et al.* [42]	SVM, NB	✓				99.41	99.64	0.77
Nagaraja *et al.* [43]	J48				✓	99.44	87.6	
Chang *et al.* [25]	SVM, RF		✓					3
Matel *et al.* [26]	SVM			✓		93		3.878
Sun *et al.* [27]	SVM				✓	96.122		
Sakr *et al.* [34]	SVM	✓	✓		✓	91.686	97.55	1.4

Filt: filter; Wra: wrapper; Emb: embedded; FT: feature transformation.

Table 14. Summary of all techniques used in the selected research papers – Springer-Link.

Research Paper	**ML Algorithms**	**Feature Selection**			**FT**	**Evolution Metric**		
		Fil	Wra	Emb		AR	DR	FPR
Ghazy *et al.* [44]	RF		✓		✓			0.001
Kunhare *et al.* [45]	RF			✓		99.32	99.26	0.62
Kasongo and Sun [46]	XGBoost- DT	✓				90.85		
Bindra and Sood [47]	RF				✓	96		
Rajadurai and Gandhi [48]	Ensemble Gradient descent, RF			✓		91.06	99.77	
Alamiedy *et al.* [49]	RF		✓			93.64		
Zhu and Zheng [50]	SVM					99.31		
Sebbar *et al.* [51]	RF	✓				97.4	98.9	
Thakur and Kumar [52]	RF	✓	✓			99.1		
Abhale and Manivannan [53]	SVM					84.0	0.86	0.8
Verma and Ranga [54]	DT				✓	94.07		3.80
Moon *et al.* [55]	DT					89.1	84.7	

Filt: filter; Wra: wrapper; Emb: embedded; FT: feature transformation.

7. Discussion and Future Challenges

7.1 Main Findings

Figure 6 shows how many supervised ML classifiers are used in the selected research papers. It can be observed that RF classifier is generally preferred, due to its accurate classification performance (i.e., ability to detect zero-day attacks) and low computational costs in real time (Table 6). From the selected research papers, feature selection is the most used dimensionality reduction technique for the proposed NIDS (Figure 7). These techniques reduce feature dimensionality to reduce the complexity of the training and testing phases, ultimately ensuring real-time detection, but at the cost of more computational resources. Figure 8 shows that the most used evaluation metrics are AR, DR and FPR. Efficient NIDS requires high AR and DR, with low FPR. Thus, to evaluate the efficiency of the NIDS, these values must be calculated.

7.2 Research Challenges

Most of the proposed NIDSs were constructed in laboratory conditions (not in a real environment), using predefined datasets and there is no proof of their efficiency in real-world implementations. Testing NIDS effectiveness in real NW traffic remains a research challenge. The proposed NIDS is complex and its computational and time costs are considerable, which may affect real-time detection. Although dimensionality reduction techniques are being used for this purpose, more improvement is still needed in the field.

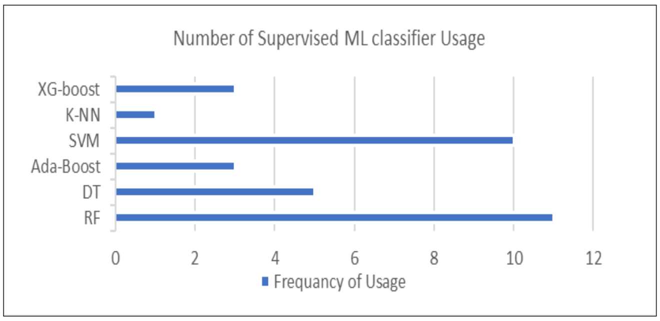
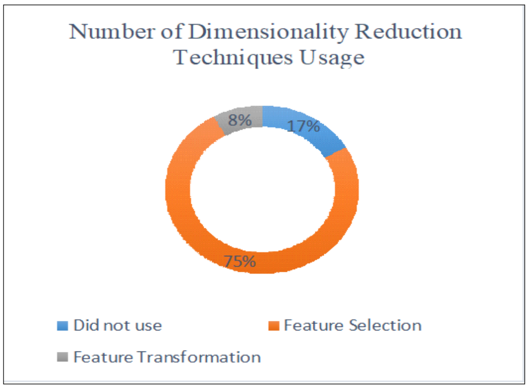


Figure 6. Use of supervised ML classifier.



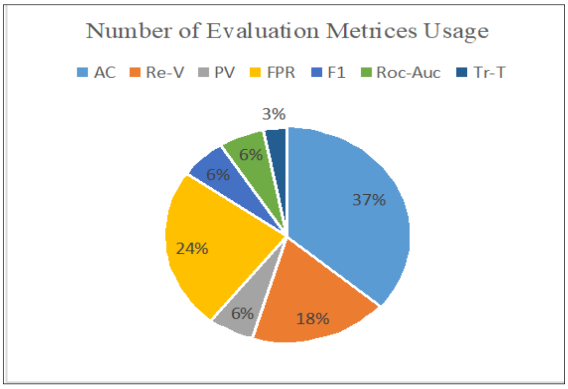


Figure 7. Use of dimensionality reduction techniques.

Figure 8. Use of evaluation metrics.

8. Conclusions

This systematic review extensively analyzed NIDSs based on supervised ML classifiers and dimensionality reduction techniques to provide updated knowledge for new interested researchers in this field. A systematic approach was adopted to select relevant research papers to answer the RQs. According to the results, RF is the most supervised ML classifier, due to its accurate classification performance and low computational costs. Feature selection techniques are the most used for dimensionality reduction in recently proposed NIDSs. These techniques reduce feature dimensionality to reduce the complexity of the training and testing phases and eventually ensure accurate real-time detection, but they need more computational resources. The most commonly used metrics are AR, DR and FPR. An efficient NIDS requires high AR and DR, with low FPR; these values must be determined for NIDS efficiency evaluation. This systematic review concludes that despite all efforts in the ML NIDS field, there are still some challenges facing interested researchers, including proving the effectiveness of the proposed ML NIDS implementation in a real NW traffic environment and reducing its complexity to ensure real-time detection. This systematic review is limited by being restricted to only 34 research papers within the domain of supervised anomaly ML-based NIDSs. Future work needs to address more research papers in a broader domain, including ML and deep learning techniques.

References

[1] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah and F. Ahmad, "Network Intrusion Detection System: A Systematic Study of Machine Learning and Deep Learning Approaches," Transactions on Emerging Telecom. Technologies, vol. 32, no. 1, pp. 1–29, DOI: 10.1002/ett.4150, Oct. 2021.

[2] P. Spadaccino and F. Cuomo, "Intrusion Detection Systems for IoT: Opportunities and Challenges Offered by Edge Computing," *arXiv*, Art no. 2012.01174v1, Dec. 2020.

[3] D. Anderson, T. Frivold and A. Valdes, "Next-generation Intrusion Detection Expert System (NIDES): A Summary," Computer Science Laboratory Rep. SRI-CSL-95-07, [Online], Available: http://merlot.usc.edu/cs530-s04/papers/Anderson95a.pdf, May 1995.

[4] A. S. Alzahrani, R. A. Shah, Y. Qian and M. Ali, "A Novel Method for Feature Learning and Network Intrusion Classification," Alexandria Engineering Journal, vol. 59, no. 3, pp. 1159–1169, Jun. 2020.

[5] W. A. Gould, "Spoilage of Canned Tomatoes and Tomato Products," in: Tomato Production, Processing and Technology, Ch. 25, pp. 419–431, 3rd Ed., Sawston, U.K.: Woodhead Publishing, DOI: 10.1533/9781845696146.3.419, 1992.

[6] S. M. Othman, F. Mutaher Ba-Alwi, N. T. Alsohybe and A. T. Zahary, "Survey on Intrusion Detection System Types," Int. J. Cyber-Security Digit. Forensics, vol. 7, no. 4, pp. 444–462, Dec. 2018.

[7] A. Verma and V. Ranga, "Statistical Analysis of CIDDS-001 Dataset for Network Intrusion Detection Systems Using Distance-based Machine Learning," Procedia Computer Science, vol. 125, pp. 709–716, DOI: 10.1016/j.procs.2017.12.091, Dec. 2018.

[8] A. S. Shekhawat, F. Di Troia and M. Stamp, "Feature Analysis of Encrypted Malicious Traffic," Expert Systems with Applications, vol. 125, pp. 130–141, DOI: 10.1016/j.eswa.2019.01.064, Feb. 2019.

[9] T. Aldwairi, D. Perera and M. A. Novotny, "An Evaluation of the Performance of Restricted Boltzmann Machines As a Model for Anomaly Network Intrusion Detection," Computer Networks, vol. 144, pp. 111–119, DOI: 10.1016/j.comnet.2018.07.025, Oct. 2018.

[10] P. Dahiya and D. K. Srivastava, "Network Intrusion Detection in Big Dataset Using Spark," Procedia Computer Science, vol. 132, pp. 253–262, DOI: 10.1016/j.procs.2018.05.169, 2018.

[11] A. Singh, N. Thakur and A. Sharma, "A Review of Supervised Machine Learning Algorithms," Proc. of the 3rd IEEE International Conference on Computing for Sustainable Global Development (INDIACom), pp. 1310–1315, New Delhi, India, Mar. 2016.

[12] S. Khatri, A. Arora and A. P. Agrawal, "Supervised Machine Learning Algorithms for Card Fraud Detection: A Comparison," Proc. of the 10th IEEE Int. Conf. on Cloud Computing, Data Science &Amp; Engineering (Confluence), pp. 680–683, Noida, India, Mar. 2020.

[13] M. Qasaimeh, R. Turab and R. S. Al-Qassas, "Authentication Techniques in Smart Grid: A Systematic Review," Telkomnika, vol. 17, no. 3, pp. 1584–1594, DOI: 10.12928/TELKOMNIKA.V17I3.11437, Jun. 2019.

[14] Z. Ashi, L. Aburashed, M. Al-Fawa'reh and M. Qasaimeh, "Fast and Reliable DDoS Detection Using Dimensionality Reduction and Machine Learning," Proc. of the 15th IEEE Int. Conf. for Internet Technology and Secured Transactions (ICITST), DOI: 10.23919/ICITST51030.2020.9351347, London, UK, Dec. 2020.

[15] S. B. Kotsiantis, D. Kanellopoulos and P. E. Pintelas, "Data Preprocessing for Supervised Learning," International Journal of Computer and Information Engineering's, vol. 1, no. 12, pp. 4104–4110, 2007.

[16] D. Stiawan et al., "An Approach for Optimizing Ensemble Intrusion Detection Systems," IEEE Access, vol. 9, pp. 6930–6947, DOI: 10.1109/ACCESS.2020.3046246, Dec. 2021.

[17] W. Xue and T. Wu, "Active Learning-based XGBoost for Cyber Physical System against Generic AC False Data Injection Attacks," IEEE Access, vol. 8, pp. 144575–144584, Aug. 2020.

[18] R. Vijayanand and D. Devaraj, "A Novel Feature Selection Method Using Whale Optimization Algorithm and Genetic Operators for Intrusion Detection System in Wireless Mesh Network," IEEE Access, vol. 8, pp. 56847–56854, DOI: 10.1109/ACCESS.2020.2978035, Mar. 2020.

[19] S. Aljawarneh, M. Aldwairi and M. B. Yassein, "Anomaly-based Intrusion Detection System through Feature Selection Analysis and Building Hybrid Efficient Model," J. of Computational Science, vol. 25, no. October, pp. 152–160, DOI: 10.1016/j.jocs.2017.03.006, Mar. 2018.

[20] C. T. Tran, M. Zhang, P. Andreae and B. Xue, "Bagging and Feature Selection for Classification with Incomplete Data," Proc. of the European Conference on the Applications of Evolutionary Computation (EvoApplications 2017), Part of the Lecture Notes in Computer Science Book Series, vol. 10199, Cham: Springer, DOI: 10.1007/978-3-319-55849-3_31, 2017.

[21] R. Ihya, A. Namir, S. El Filali, M. Ait Daoud and F. Z. Guerss, "J48 Algorithms of Machine Learning for Predicting a User's Acceptance of an E-orientation Systems," Proceedings of the 4th ACM International Conference on Smart City Applications (SCA '19), Article no. 20, pp. 1-8, DOI: 10.1145/3368756.3368995, Oct. 2019.

[22] F. Alam and S. Pachauri, "Comparative Study of J48, Naive Bayes and One-R Classification Technique for Credit Card Fraud Detection Using WEKA," Journal of Advanced Computer Science & Technology, vol. 10, no. 6, pp. 1731–1743, 2017.

[23] R. Harode, "XGBoost: A Deep Dive into Boosting," SFU/// Professional Master's Program in Computer Science, [Online], Available: https://medium.com/sfu-cspmp/xgboost-a-deep-dive-into-boosting-

f06c9c41349 (accessed on Oct. 15, 2021).

[24] M. Guia, R. R. Silva and J. Bernardino, "Comparison of Naive Bayes, Support Vector Machine, Decision Trees and Random Forest on Sentiment Analysis," Proc. 11th Int. Jt. Conf. Knowl. Discov. Knowl. Eng. Knowl. Manag. (KDIR 2019), vol. 1, pp. 525–531, DOI: 10.5220/0008364105250531, Nov. 2019.

[25] Y. Chang, W. Li and Z. Yang, "Network Intrusion Detection Based on Random Forest and Support Vector Machine," Proc. of 2017 IEEE Int. Conf. on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), vol. 1, pp. 635–638, DOI: 10.1109/CSE-EUC.2017.118, Guangzhou, China, Jul. 2017.

[26] E. C. Matel, A. M. Sison and R. P. Medina, "Optimization of Network Intrusion Detection System Using Genetic Algorithm with Improved Feature Selection Technique," Proc. of the IEEE 11th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag. (HNICEM), Article no. 19556390, Laoag, Philippines, DOI: 10.1109/HNICEM48295.2019.9073439, 2019.

[27] S. Sun, Z. Ye, L. Yan, J. Su and R. Wang, "Wrapper Feature Selection Based on Lightning Attachment Procedure Optimization and Support Vector Machine for Intrusion Detection," Proc. of the 4th IEEE Int. Symposium on Wireless Systems within the Int. Conf. on Intelligent Data Acquisition and Advanced Computing Systems, pp. 41–46, Lviv, Ukraine, Sep. 2018.

[28] M. Pechenizkiy, A. Tsymbal and S. Puuronen, "PCA-based Feature Transformation for Classification: Issues in Medical Diagnostics," Proc. of the 17th IEEE Symposium on Computer-based Medical Systems, vol. 17, pp. 535–540, DOI: 10.1109/cbms.2004.1311770, Bethesda, MD, USA, Jul. 2004.

[29] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari and J. Saeed, "A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction," Journal of Applied Science and Technology Trends, vol. 1, no. 2, pp. 56–70, DOI: 10.38094/jastt1224, May 2020.

[30] D. Mladenić, "Feature Selection for Dimensionality Reduction," Proc. of Subspace, Latent Structure and Feature Selection, vol. 3940, C. Saunders, M. Grobelnik, S. Gunn and J. Shawe-Taylor, Eds., Berlin: Springer, pp. 84–102, DOI: 10.1007/11752790_5, 2006.

[31] M. Masaeli, G. Fung and J. G. Dy, "From Transformation-based Dimensionality Reduction to Feature Selection," Proceedings of the 27th International Conference on Machine Learning, pp. 751–758, DOI: 10.5555/3104322.3104418, Haifa, 2010.

[32] A. Nazir and R. A. Khan, "A Novel Combinatorial Optimization based Feature Selection Method for Network Intrusion Detection," Computers & Security, vol. 102, Article no. 102164, DOI: 10.1016/j.cose.2020.102164, Mar. 2021.

[33] M. Mazini, B. Shirazi and I. Mahdavi, "Anomaly Network-based Intrusion Detection System Using a Reliable Hybrid Artificial Bee Colony and AdaBoost Algorithms," Journal of King Saud University - Computer and Information Sciences, vol. 31, no. 4, pp. 541–553, Oct. 2019.

[34] M. M. Sakr, M. A. Tawfeeq and A. B. El-Sisi, "Filter *versus* Wrapper Feature Selection for Network Intrusion Detection System," Proc. of the IEEE 9th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS), pp. 209–214, DOI: 10.1109/ICICIS46948.2019.9014797, Cairo, Egypt, Dec. 2019.

[35] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrão and M. L. Proença, "Network Anomaly Detection System Using Genetic Algorithm and Fuzzy Logic," Expert Systems with Applications, vol. 92, no. C, pp. 390–402, DOI: 10.1016/j.eswa.2017.09.013, Feb. 2018.

[36] J. L. G. Torres, C. A. Catania and E. Veas, "Active Learning Approach to Label Network Traffic Datasets," Journal of Information Security and Applications, vol. 49, Article no. 102388, 2019.

[37] B. Anderson and D. McGrew, "Machine Learning for Encrypted Malware Traffic Classification: Accounting for Noisy Labels and Non-stationarity," Proc. of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17), vol. F1296, pp. 1723–1732, DOI: 10.1145/3097983.3098163, Aug. 2017.

[38] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee and H. Karimipour, "Cyber Intrusion Detection by Combined Feature Selection Algorithm," Journal of Information Security and Applications, vol. 44, pp. 80–88, DOI: 10.1016/j.jisa.2018.11.007, Feb. 2019.

[39] S. Dwivedi, M. Vardhan and S. Tripathi, "An Effect of Chaos Grasshopper Optimization Algorithm for Protection of Network Infrastructure," Computers Networks, vol. 176, Article no. 107251, DOI: 10.1016/j.comnet.2020.107251, May 2020.

[40] V. Kanimozhi and T. P. Jacob, "Artificial Intelligence Outflanks All Other Machine Learning Classifiers in Network Intrusion Detection System on the Realistic Cyber Dataset CSE-CIC-IDS2018 Using Cloud Computing," ICT Express, vol. 7, no. 3, pp. 366–370, DOI: 10.1016/j.icte.2020.12.004, Sep. 2021.

[41] H. Jiang, Z. He, G. Ye and H. Zhang, "Network Intrusion Detection based on PSO-Xgboost Model," IEEE Access, vol. 8, pp. 58392–58401, DOI: 10.1109/ACCESS.2020.2982418, Mar. 2020.

[42] P. Ding, J. Li, M. Wen, L. Wang and H. Li, "Efficient BiSRU Combined with Feature Dimensionality Reduction for Abnormal Traffic Detection," IEEE Access, vol. 8, pp. 164414–164427, DOI: 10.1109/ACCESS.2020.3022355, Sep. 2020.

[43] A. Nagaraja, U. Boregowda, K. Khatatneh, R. Vangipuram, R. Nuvvusetty and V. Sravan Kiran, "Similarity-based Feature Transformation for Network Anomaly Detection," IEEE Access, vol. 8, pp. 39184–39196, DOI: 10.1109/ACCESS.2020.2975716, Feb. 2020.

[44] R. A. Ghazy, E. S. M. EL-Rabaie, M. I. Dessouky, N. A. El-Fishawy and F. E. Abd El-Samie, "Efficient Techniques for Attack Detection Using Different Features Selection Algorithms and Classifiers," Wireles Personal Communication, vol. 100, no. 4, pp. 1689–1706, DOI: 10.1007/s11277-018-5662-0, May 2018.

[45] N. Kunhare, R. Tiwari and J. Dhar, "Particle Swarm Optimization and Feature Selection for an Intrusion Detection System," Sadhana, vol. 45, Article no. 109, DOI: 10.1007/s12046-020-1308-5, May 2020.

[46] S. M. Kasongo and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," Journal of Big Data, vol. 7, Article no. 105, DOI: 10.1186/s40537-020-00379-6, Nov. 2020.

[47] N. Bindra and M. Sood, "Detecting DDoS Attacks Using Machine Learning Techniques and Contemporary Intrusion Detection Dataset," Automatic Control and Computer Sciences, vol. 53, no. 5, pp. 419–428, DOI: 10.3103/S0146411619050043, Nov. 2019.

[48] H. Rajadurai and U. D. Gandhi, "A Stacked Ensemble Learning Model for Intrusion Detection in a Wireless Network," Neural Comp. and App., vol. 5, DOI: 10.1007/s00521-020-04986-5, May 2020.

[49] T. A. Alamiedy, M. Anbar, Z. N. M. Alqattan and Q. M. Alzubi, "Anomaly-based Intrusion Detection System Using Multi-objective Grey Wolf Optimization Algorithm," Journal of Ambient Intelligence and Humanized Computing, vol. 11, pp. 3735–3756, DOI: 10.1007/s12652-019-01569-8, Nov. 2019.

[50] Y. Zhu and Y. Zheng, "Traffic Identification and Traffic Analysis Based on Support Vector Machine," Neural Comput. Appl., vol. 32, pp. 1903–1911, DOI: 10.1007/s00521-019-04493-2, Sep. 2020.

[51] A. Sebbar, K. Zkik, Y. Baddi, M. Boulmalf and M. D. E. C. El Kettani, "MitM Detection and Defense Mechanism CBNA-RF Based on Machine Learning for Large-scale SDN Context," J. of Ambient Intelligence and Humanized Comp., vol. 11, pp. 5875–5894, DOI: 10.1007/s12652-020-02099-4, 2020.

[52] K. Thakur and G. Kumar, "Nature Inspired Techniques and Applications in Intrusion Detection Systems: Recent Progress and Updated Perspective," Archives of Computational Methods in Engineering, Article no. 0123456789, DOI: 10.1007/s11831-020-09481-7, Aug. 2020.

[53] A. B. Abhale and S. S. Manivannan, "Supervised Machine Learning Classification Algorithmic Approach for Finding Anomaly Type of Intrusion Detection in Wireless Sensor Network," Optical Memory and Neural Networks, vol. 29, pp. 244-256, DOI: 10.3103/S1060992X20030029, 2020.

[54] A. Verma and V. Ranga, "Evaluation of Network Intrusion Detection Systems for RPL Based 6LoWPAN Networks in IoT," Wireless Personal Communications, vol. 108, pp. 1571–1594, DOI: 10.1007/s11277-019-06485-w, Oct. 2019.

[55] D. Moon, H. Im, I. Kim and J. H. Park, "DTB-IDS: An Intrusion Detection System Based on Decision Tree Using behavior Analysis for Preventing APT Attacks," Journal of Supercomputing, vol. 73, pp. 2881–2895, DOI: 10.1007/s11227-015-1604-8, Jul. 2017.

[56] N. Martins, J. M. Cruz, T. Cruz and P. H. Abreu, "Adversarial Machine Learning Applied to Intrusion and Malware Scenarios: A Systematic Review," IEEE Access, vol. 8, pp. 35403–35419, Feb. 2020.

[57] A. A. Ramaki, A. Rasoolzadegan and A. J. Jafari, "A Systematic Review on Intrusion Detection Based on the Hidden Markov Model," Statistical Analysis and Data Mining, vol. 11, pp. 111–134, Apr. 2018.

[58] C. Gonzalez, "Increasing Security in Military Self-protected Software," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 7, no. 3, pp. 253–267, Sep. 2021.

[59] A. B. Nassif, M. A. Talib, Q. Nasir, H. Albadani and F. M. Dakalbab, "Machine Learning for Cloud Security: A Systematic Review," IEEE Access, vol. 9, pp. 20717–20735, Jan. 2021.

"Network Intrusion Detection Systems Using Supervised Machine Learning Classification and Dimensionality Reduction Techniques: A Systematic Review", Z. Ashi, L. Aburashed, M. Al-Qudah and Abdallah Qusef.

ملخص البحث:

إنّ حماية السّرية والسلامة والتّوافُر للحيّز السّيبراني وأصول الشّبكات أصبحت مسألة تحظى بالاهتمام بشكلٍ متزايد. وقد خلقت الزّيادة السّريعة في استخدام الإنترنت ووجود الأنظمة الحديثة للحوسبة (مثل الحوسبة السّحابية) دوافع كبيرة للتَّطفُّل. لذا فإنّ على مهندسي الأمان أن يطوّروا تقنياتٍ مُبتكرة لمواجهة التهديدات التي تعترض أصول الشّبكات.

لقد ظهرت تقنيات جديدة متقدمة لإيجاد أنظمة أكثر فعالية لكشف التّطفّل باستخدام تقنيات تعلُّم الآلة وتقليل الحجم، وذلك لمساعدة مهندسي الأمان في التّوصُّل الى أنظمة عالية الفعالية في هذا المجال.

تقدم هذه الورقة مراجعةً نظاميّةً شاملةً لأنظمة كشف التّطفّل الأحدث التي استخدمت تقنياتٍ مراقبةً للتّصنيف في تعلّم الآلة وتقليل الحجم. وتوضّح هذه المراجعة كيف عملت مصنِّفات تعلّم الآلة وتقنيات تقليل الحجم على تطوير إنشاء أنظمة كشف التّطفّل؛ وذلك من خلال مقاييس التّقييم المعتمدة لمثل هذه الأنظمة.

وتجدر الإشارة الى أنّ النّقطة الأساسية في هذه الورقة تتمثّل في توفير أحدث المعرفة للباحثين المهتمين الجُدد في هذا المجال.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 04, December 2021.

A MOBILE AGENT-BASED METHOD TO COUNTER SINKHOLE ATTACKS IN WIRELESS SENSOR NETWORKS

Hadi Khosravi[1] and Mohammad GhasemiGol[2]

(Received: 29-Sep.-2021, Revised: 22-Nov.-2021, Accepted: 25-Nov.-2021)

ABSTRACT

Wireless sensor networks (WSNs) are an applied technology widely used in various areas. According to the WSN limitations, they usually face many types of attacks. The sinkhole attack is the most popular and dangerous attack in the routing of WSNs. There are many approaches to counter sinkhole attacks in the literature. The mobile agent methods generate better results in facing sinkhole attacks and overcoming the WSN limitations. In this paper, we present a new mobile agent-based method that applies the trust value of each sensor to detect and prevent sinkhole attacks. We compute the trust values to inform the sensor nodes about their neighbors' reputations. As shown in the experiments, the proposed method generates better results in packet loss ratio. It also fixes the security flaws of previous works and reduces the agents' overhead in the network compared to previous methods.

KEYWORDS

Security, Trust management, Mobile agent, Sinkhole attacks, Wireless sensor networks.

1. INTRODUCTION

Wireless sensor networks (WSNs) consist of many multifunctional low-cost and small-size sensors. These nodes are deployed in an unattended environment with the capability of sensing, wireless communication and computing (i.e., the collection and dissemination of environmental data). WSN is a combination of sensing and embedded techniques, distributed information processing and communication mechanisms [7]. It has many applications in various areas and is usually deployed in a risky environment [3], [19]. Thus, security is a vital issue in WSNs. Due to the resource constraints of the nodes, applying security mechanisms in these types of networks is a very conservative task [13]. Sensor nodes are physically placed in an open environment and are unprotected. WSNs have many more constraints than traditional networks. Because of computing, resource constraints and the broadcast nature of the transmission medium, WSNs have different security challenges compared to traditional networks. Moreover, these networks are easily exposed to a variety of security attacks. In most security attacks in WSNs, the compromised sensor nodes insert fake information into the network [23]. Therefore, WSN protocols and algorithms must be self-organized and security mechanisms should be considered to protect the sensor nodes against various attacks [9].

A routing protocol is a software placed on the network layer and is responsible for decisions about the output route of packets that should be transmitted. In other words, it is an algorithm to find a way to transfer data. Typically, in WSNs, the destination node is called a base station. The distance between the source and the destination nodes may be far or even outside of the transmission range. Therefore, data may be transmitted to reach the sink node through multiple hops [25]. Different attacks have been designed and implemented based on essential tasks of the network layer, which endanger data packets' security.

In this paper, amongst various attacks in the network layer, the sinkhole attack has been chosen, which is one of the most widespread and destructive attacks. The sinkhole attack is a dangerous attack in WSNs that prevents reaching complete and correct information to the base station node. In this attack, a malicious node misleads the surrounding nodes to attract traffic from a specific path [15], [12]. As shown in Figure 1, the malicious node claims have the shortest path to the sink. The primary metric for data routing is the best path to the base station in WSNs [6]. Hence, the malicious node can absorb a portion of network packets illegally. In fact, in a sinkhole attack, a malicious node sends false information about the routing to the neighbors to encourage them to choose it as their parent to get the network traffic of

[1] H. Khosravi is with the CERT Coordination Center, Univ. of Birjand, Birjand, Iran. Email: h.khosravi@cert.birjand.ac.ir

[2] M. GhasemiGol (Corresponding Author) is with Department of Electrical and Computer Engineering, University of Birjand, Birjand, Iran. Email: ghasemigol@birjand.ac.ir

that area [6]. Now, it can launch other attacks, such as selective forwarding, packet drops and packet modifications [15]. Thus, the sinkhole attack decreases the network's lifetime and increases the network overhead [14].

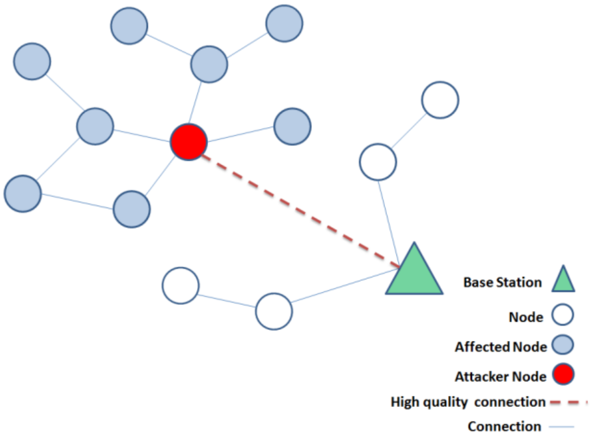


Figure 1. The sinkhole attack in a WSN [9].

There are many methods in the literature to address sinkhole attacks [1]-[2]. Here, we focus on the agent-based methods that apply mobile agents as a self-controlling program to transfer code and data between the sensor nodes [11], [16], [4]. With the aid of agents, we can reduce the communication costs by moving the processing code to the sensors instead of transferring data to a central processor node.

In this paper, we propose a new mobile agent-based method that applies the trust value of each sensor to detect and prevent sinkhole attacks. We use the trust value of each sensor to inform the sensor nodes about their neighbors' reputations. Hence, the main contributions of this paper are as follows:

- We compute the trust value of each node to inform the sensor nodes about their neighbor's reputations.
- We apply mobile agents to reduce the communication costs in WSNs by eliminating unnecessary data transfer.

The rest of this paper is organized as follows. We investigate the related work in Section 2. The proposed method is described in Section 3. In Section 4, we compare our proposed method with two related works regarding packet loss, energy consumption, throughput and agent overhead. Finally, the conclusion and the future direction are mentioned in Section 5.

2. Related Work

So far, many methods have been presented to detect and prevent sinkhole attacks. In this section, we examine some of them and the related work in this area. Sheela et al. detected the sinkhole attacks in WSNs by a mobile agent-based method. The mobile agents collect information from all sensor nodes to make each node aware of the network in terms of the malicious nodes. Normal nodes do not accept the fake information of the compromised nodes. This method has agent navigation and data routing algorithms which the first algorithm describes how to visit all nodes and to give the network information to nodes by the mobile agent. Also, the second algorithm describes how to use a node of this global network information for routing data packets. An essential feature of this method is that it detects the sinkhole attack without any encryption or decryption mechanism. However, if the number of nodes increases, the overhead of this method will be very high [22].

Sharmila and Umamaheswari have presented a solution to detect sinkhole attacks using message digest. In this method, a control scheme can be built for each packet in the network by using hash functions. The digest message is calculated by the source node using a combination of MD5 and SHA1 hash algorithms and is sent through a trustable path to the base station. Then, the message is sent through a node that claims to have the shortest route to the sink. If an adversary modifies the message, it can be detected by checking the modified message and the message digest. Due to the use of SHA1 and MD5 algorithms, overhead caused by the encryption and decryption operations is high and transfer the message is transferred in the two paths causing loss of energy and traffic overhead that are disadvantages of this method. Also, having a trustable path in this type of network is often not possible due to their nature [21].

Bahekmat et al. have presented an efficient algorithm to detect sinkhole attacks in WSNs. In their proposed algorithm, it is assumed that all nodes in the network are similar, randomly distributed and aware of their locations in the network. Each node sends a control packet to the base station directly before sending the data packet through hop-by-hop routing. If any change is made on the control fields, it indicates that there are malicious nodes in the path. The advantage of this method is the reduction of packet loss rate and energy consumption. The main disadvantage of this algorithm is that each sensor needs to use localization algorithms or have a Global Position System (GPS) in order to know its geographical location, which requires additional costs [5].

Hamedheidari and Rafeh have presented a defensive mechanism by mobile agents to counter sinkhole attacks. Each node is aware of its trusted neighbors using mobile agents with a three-step negotiation. The main purpose of the three-step trusting procedure between the node and the agent is an authentication mechanism that uses unique codes and hash algorithms. In this method, it has been assumed that all nodes are physically protected. This assumption is not very logical due to the nature of this type of network that is placed in remote areas. Now, by omitting this assumption, the attacker can gain access to the node physically. As a result, this method suffers from tampering attacks [9].

Naderi et *al.* detected the sinkhole area according to the energy consumption model in the network and the energy deviation of each node from other nodes. Also, nodes' energy information is collected and analyzed by the sink. Then, a trust evaluation mechanism is used, so that each node calculates the trust value of its neighbors. The trust mechanism starts after observing a contradiction in energy consumption in a limited area of the network and then a trust value is assigned to each node based on security requirements by the sensed event. The advantage of this method is to achieve considerable performance in factors that have higher risk. For example, a network that has more nodes and compromised nodes has a short distance to the sink and delivers more packets to the sink under challenging conditions. The major disadvantage of this method is that it acts only based on energy criteria. In other words, if a node has a higher energy consumption due to more telecommunication capabilities, it is incorrectly detected as a malicious node, which is referred to as a false positive [17].

Jahandoust and Ghassemi proposed an adaptive framework with a combination of subjective logic and an extension of timed automata to counter sinkhole attacks in WSNs. For this reason, they utilized a stochastic extension of the AODV routing algorithm. A subjective logic model is applied to detect the sinkhole nodes and find the most reliable path. Also, a probabilistic model monitors the network behavior to adaptively adjust the algorithm parameters [10].

In recent research, Nwankwo and Abdulhamid applied the ant colony method to detect sinkhole areas [18]. Although they claim that their method can improve the detection rate and false alerts, it applies ant colony as a time-consuming method which is not proper for WSN applications. Wang presented a three-layer detection scheme to monitor the heterogeneous Industrial WSN (IWSN). Unlike the previous method, their scheme does not utilize information and location information from the neighbors. At the first layer, the normal and Sybil nodes are found by a quadratic difference based on the received signal strength indicator (RSSI). The second layer continues the search for nodes detected in the first layer using a method based on residual energy. Finally, the base station detects the first and second high-energy nodes [24]. Jatti and Sonti presented an agent-based algorithm to detect and prevent sinkhole attacks in WSNs [11]. Their work is very similar to Hamedheidari and Rafeh’s method [9]. They just apply their presented method to a different routing algorithm and evaluate it through network simulator NS 2.35.

In this paper, we present a new mobile agent-based method to counter sinkhole attacks. Therefore, we focus on the agent-based literature and exceed it to fix the security flaws of previous works and reduce the overhead caused by the presence of the agents in the network. We compare our proposed method with two agent-based related works to show its performance in facing sinkhole attacks.

3. Proposed Method

Here, we explain our proposed method to apply mobile agents and trust management. In our method, each node computes the trust value of its neighbors. Furthermore, it uses mobile agents to make each node aware of the reputation values of its neighbors. Consequently, each node identifies its compromised neighbors and does not interact with them.

3.1 Agent Designing

Such as other methods, the mobile agent is an executable script that migrates from one node to another in the format of an agent packet. In the proposed method, we use a simplified type of agent code to detect the sinkhole nodes. As a result, it produces less computational overhead and decreases energy consumption in the nodes as well. Furthermore, agents do not communicate with each other and only interact with the nodes placed on them. So, no traffic overhead will exist due to the agents' communication with each other. It reduces the energy consumption in the nodes significantly [9].

3.2 Agent Migration

The migration allows an agent to move from an agent node (the node that contains the agent) to a neighbor node and back to the original node. Therefore, the agent only moves to a one-hop neighbor and does not need to maintain the agent migration path to return to the original node. As a result, the source and destination storage will suffice. We define here another action that is the agent cycling. Agent cycling is done when a mobile agent migrates to all one-hop nodes around an agent node.

3.3 Algorithm

First, nodes are randomly distributed with a uniform distribution in the network. After that, the base station selects several sensor nodes based on the expected number of agents in the network and sends agent packets to them. After receiving agents, each node sends a HELLO packet to neighbor nodes and creates a neighboring matrix. Figure 2 shows a WSN with nine nodes. In this network, node A consists of an agent and H is the malicious node. Table 1 shows the neighboring matrix of node B after sending HELLO packets. As can be seen, in this step, detected are only neighbors of a node which may be malicious.

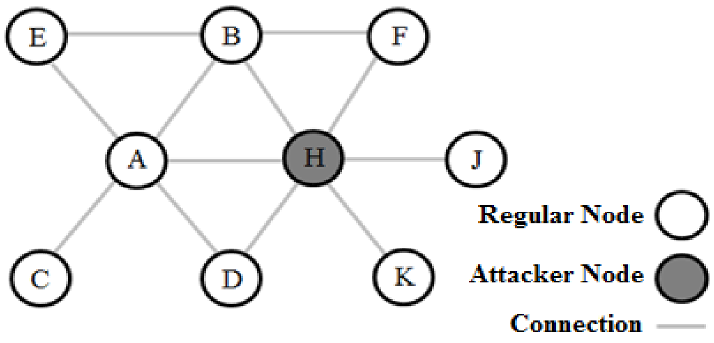


Figure 2. A WSN with nine nodes.

Table 1. Neighboring matrix of node B after sending HELLO packets.

The ID of the Neighbor Node	Node A	Node E	Node F	Node H
Sent packets	0	0	0	0
Correct packets	0	0	0	0
Sent packets 2	0	0	0	0
Trust value	50	50	50	50
Update state	0	0	0	0

Each node increases the number of sent packets variable by one after transmitting a packet to its one-hop neighbor. Next, each sender node (*x*) monitors its neighbor (*y*) for a limited time to investigate its forwarding behaviors. If node *x* detects the correct retransmission, it will increase the number of correct packets variable by one [8]. Before forwarding a new packet, each node checks to know whether the number of sent packets variable has reached a predefined threshold of forwarding, 100 for example, or not. Then, the direct trust from node x to node y can be calculated by the formula and variable value 'number of sent packets' proposed in [20] as in Equation (1) 'Number of correct packets' is set to zero.

$$T_{x,y} = \frac{S_{x,y}}{S_{x,y} + U_{x,y}} \times 100 \tag{1}$$

where $T_{x,y}$ is the trust from node x to node y, $S_{x,y}$ is the number of correct packets forwarded by node y and $U_{x,y}$ is the number of packets dropped by node y. Also, to reduce the forwarded data and computational overhead, trust values are stored as unsigned integers in the range of 0 to 100 instead of decimal values that represent the lowest and the highest level of trust, respectively. To update the trust for node y at node x (i.e., trust value variable), we apply formula (1) to calculate the trust average. The trust value in node x is calculated and stored in node x as well. Moreover, to get a more accurate trust value, the reputation value is calculated by using mobile agents. For this reason, the trust values of all one-hop neighbors of an agent node are collected by the agent. The reputation values are calculated at the agent node and then new reputation values are submitted to the neighbor nodes with the help of mobile agents.

The update state variable is binary. 0 means that the last update of variable 'trust value' has been done by direct calculation of trust value by the node and one means that it has been updated by the average of provided reputation value by the agent and stored trust value in the trust value variable. At first, the agent checks the value of the update state variable. If it is 0, it submits the reputation value of the node's neighbors to it and receives the values of variables 'trust value' and 'number of sent packets 2' of the node's neighbors and delivers them to the agent node during its migration. But, if the update state variable is set to 1, the mobile agent returns to the agent node without doing any extra operation. The purpose of using the update state variable is that if a node has not been done, the adequate number of interactions (i.e., 100), it can solely update the trust value variable. So, it is better to ignore these trust values, which can prevent the computational overhead imposed by these processes.

Furthermore, this simple binary variable prevents the impact of a fake agent on a node. According to our mechanism of updating the trust value variable, if a node updates the trust value based on the reputation, the new trust value cannot update until it is already updated based on the direct trust value which is calculated by the node. As a result, if an agent hands over incorrect reputation information to a node, the impact of this fake information can be reduced by calculating the trust value based on the direct trust calculated by the node. Initialization and updating of the variable 'update state' are done only by the node. At the first step of updating the trust value variable, the direct trust value is calculated as described earlier by the node. Then, the average of this value and the one stored in the trust value will be held as the new trust value and also the value of the two variables 'number of sent packets' and 'number of correct packets' will be changed to zero. This procedure could be done if the number of interactions reaches 100. On the other hand, if the value of the update state variable is zero, it means that the agent has not read the trust value yet. Hence, the number of interactions is stored in another variable named 'number of sent packets 2' in the node in which values will be multiples of 100 (except the default value). We use this variable to weigh the collected trust values of the nodes for calculations of reputation that are done within the agent node. Three tables are stored in the agent node; table 1 (which is also stored in all other nodes), table 2 and table 3. The data used to calculate the reputation value which has been transmitted to an agent node by the mobile agent is stored in Table 2. For each neighboring node, one dedicated table 2 is stored in an agent node. The reputation value of all nodes is stored in Table 3 as well.

Table 2. Transmitted information of the mobile agent to agent node after collecting data from node B.

The ID of the Neighbor Node	Node $_A$	Node $_E$	Node $_F$	Node $_H$
Trust value	95	90	80	10
Number of sent packets 2	200	300	200	200

Table 3. Reputation table.

The ID of the Neighbor node	Node $_A$	Node $_B$	Node $_C$	Node $_D$	Node $_E$	Node $_H$
Reputation	95	80	65	80	70	15

After agent cycling, the reputation of each node is calculated by using the information gathered by the mobile agent as follows:

$$R1_b = \frac{S_{a,b} \times T_{a,b} + S_{e,b} \times T_{e,b} + S_{f,b} \times T_{f,b} + S_{h,b} \times T_{h,b}}{S_{a,b} + S_{e,b} + S_{f,b} + S_{h,b}} \tag{2}$$

$$R2_b = \frac{R_a \times T_{a,b} + R_e \times T_{e,b} + R_f \times T_{f,b} + R_h \times T_{h,b}}{R_a + R_e + R_f + R_h} \tag{3}$$

$$R_b = \frac{R1_b + \ R2_b}{2} \tag{4}$$

where $S_{a,b}$ is the number of sent packets from node *a* to node *b*, $T_{a,b}$ is the trust value of node *a* to node *b* or the trust value variable and R_a is the reputation value of node *a*.

Since a malicious node may try to show the number of its interactions (i.e., the '*number of sent packets 2*' variable) extraordinary high to increase the impact of its comment in a weighted mean formula, we apply the following condition with a reasonable threshold of 400, for instance, to prevent this possible disorder.

```
IF number of sent packets 2 >= threshold, THEN
    number of sent packets 2 = threshold
```

4. Experimental Results

In this section, we express the security benefits of our proposed method compared with the Hamedheidari and Rafeh [9] and Jatti and Sonti [11] methods. We describe the security weaknesses of the previous methods and their incapability of detecting and resisting some sorts of attack. Moreover, we explain the advantages of our proposed method and the way in which we resolve these weaknesses. Then, we compare the proposed method with the mentioned two methods in terms of various standard evaluation parameters in normal conditions to see whether the proposed method, which brings superior safety, imposes more overhead than the previous methods or not.

In Hamedheidari and Rafeh's method, they assume that all nodes are physically protected. This assumption is not logical due to the nature of WSNs that are placed in risky areas. By removing this assumption, an attacker (i.e., human attacker) can take the node physically. In fact, it will gain access to *code 1*, *code 2 and* the NodeHashFunc(); so by having this information, it can create a fake node and place it in the network. In Hamedheidari and Rafeh's method, a fake node is treated as a normal node; and all actions of it are allowed, even hostile acts. By knowing this information, accessing *code 3* and creating a fake agent is not so hard; so the detection way of the base paper fails in this situation. Furthermore, the agent node multicasts the trust packet and its neighbors only check the sender ID of the packet to ensure transmitting by the agent node. Hence, an attacker can easily create a fake trust packet and pretend that the packet is sent by it *via* changing the ID of the trust packet to an agent node. Therefore, this can easily disrupt the network's ordinary workflow.

As described in detail in the proposed method section, unlike Hamedheidari and Rafeh's method, our proposed method has a reasonable performance in all the above conditions. Also, another positive point of the proposed method is that the value of the threshold to detect an attacker can be set according to the sensitivity of WSN's type. The more security is essential, the higher the threshold should be and *vice versa*.

4.1 Simulation

In this sub-section, we evaluate the performance of the proposed method within a simulation environment. For this purpose, we developed an agent-based simulator and then compared the results of our proposed method with two related works [9], [11].

4.1.1 Simulation Environment

Simulation environment has been considered to be 200 × 200 meters in simulations and we assume N sensor nodes with a uniform distribution that are randomly distributed in the environment and are mobile as well. Simulations have been done for N sensor nodes from 100 to 400 with the step of 100. Simulation time is 20 minutes and the results are recorded one time every 30 seconds. Also, each experiment has been repeated for any number of sensor nodes five times and corresponding diagrams are the average of 5-time runs. In each experiment, between 10-20% of nodes are malicious. Simulations were done for each number of nodes with various percents of agents (10%, 15%, 20% and 25%) in each experiment. In table 4, the simulation conditions are shown, so that E_{elect} is consumed energy to activation electronic circuits of transmitter and E_{fs} is the activation energy amplifier of the transmitter.

Table 4. Simulation environment.

Variable	**Value**
Network scale	$200_{m} \times 200_{m}$
Duration	20 $_{\text{minutes}}$
Routing protocol	AODV
Range of transmission	50 $_{m}$
Speed	10 $_{m/s}$
Initial energy	1 $_{\text{joule}}$
E_{elect}	50 $_{\text{nj/bit}}$
E_{fs}	10 $_{\text{pj/bit/signal}}$

4.1.2 Network Model

Base station: It is static and located in coordinates (100,100). In simulations, we found that this place has more efficiency. The base station is entirely safe and has infinite energy.

Sensor nodes: All nodes in the simulation are homogeneous and are not better than another. Nodes are distributed with a uniform distribution in the environment. They are mobile and move with a speed of 10 m/s by a random waypoint algorithm in all experiments.

Mobile agent: Only one type of agent is used in this method. The agents are randomly placed on nodes done by the base station at the beginning of the network creation. The agents perform agent cycling every 5 to 10 seconds.

Malicious node: Malicious nodes are the regular nodes in the network that generate sinkhole attacks. In each experiment, 10% to 20% of total nodes are malicious and distributed randomly around the network environment.

4.1.3 Experimental Result

Here, we compare the simulation results of our method with Hamedheidari and Rafeh's method [9] and Jatti and Sonti's method [11] in terms of packet loss, energy consumption, throughput and agent overhead. As shown in the experiments, our proposed method generates better results in terms of packet loss ratio and the agents' overhead. It also leads to acceptable energy consumption and throughput.

4.1.3.1 Energy Consumption

Since energy is the most vital resource for sensor nodes, the methods and approaches proposed for sensor nodes need to be economical in terms of energy consumption. Figures 3-6 show the energy consumption of our proposed method in comparison with the previous methods. As shown in these figures, increasing the number of agents increases the consumed energy. However, the amount of increase is less with 100 nodes than with 400 nodes in the network. It is because of more scattering between nodes in the large-scale networks with a few nodes. As a result, their neighbors are less and agents visit fewer nodes in every cycling. The energy consumption of the entire network is still low. But in dense networks; i.e., networks with a large number of nodes (because of having more neighbors), agent cycling is performed more, so more energy is consumed. The Hamedheidari and Jatti methods consume less energy than the proposed method because regular nodes know their trusted neighbors through trust packets that are sent by the agent node. Still, in the proposed method, a regular node calculates the trust value of its neighbors. So, in the proposed method, each node consumes more energy.

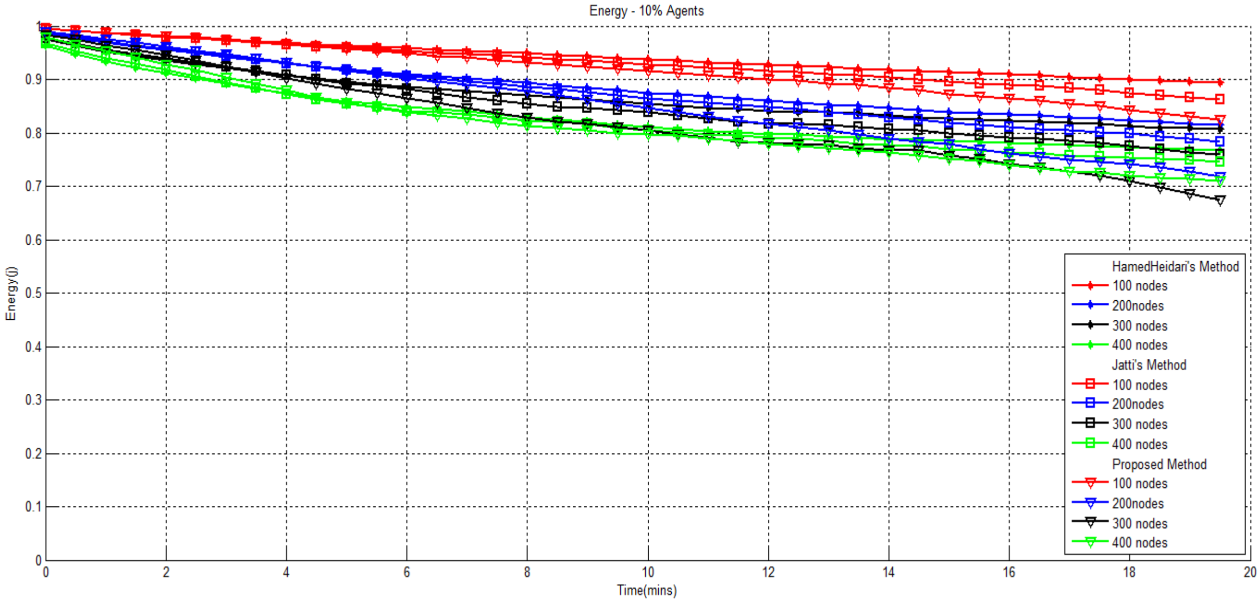


Figure 3. The energy consumption in the compared methods with 10 percent agent.

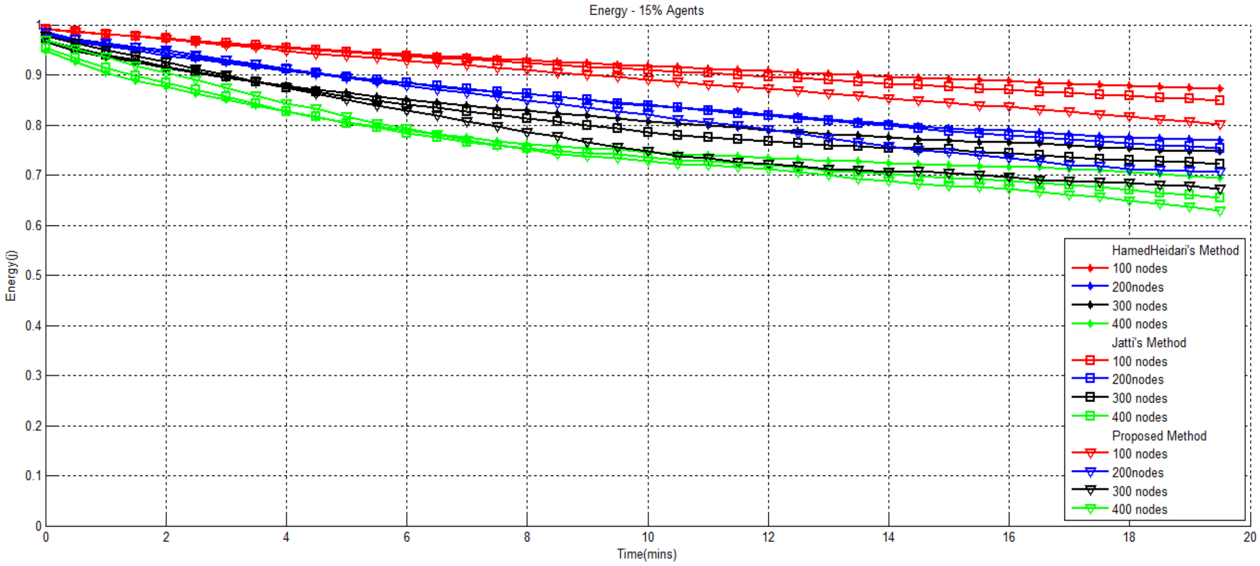


Figure 4. The energy consumption in the compared methods with 15 percent agent.

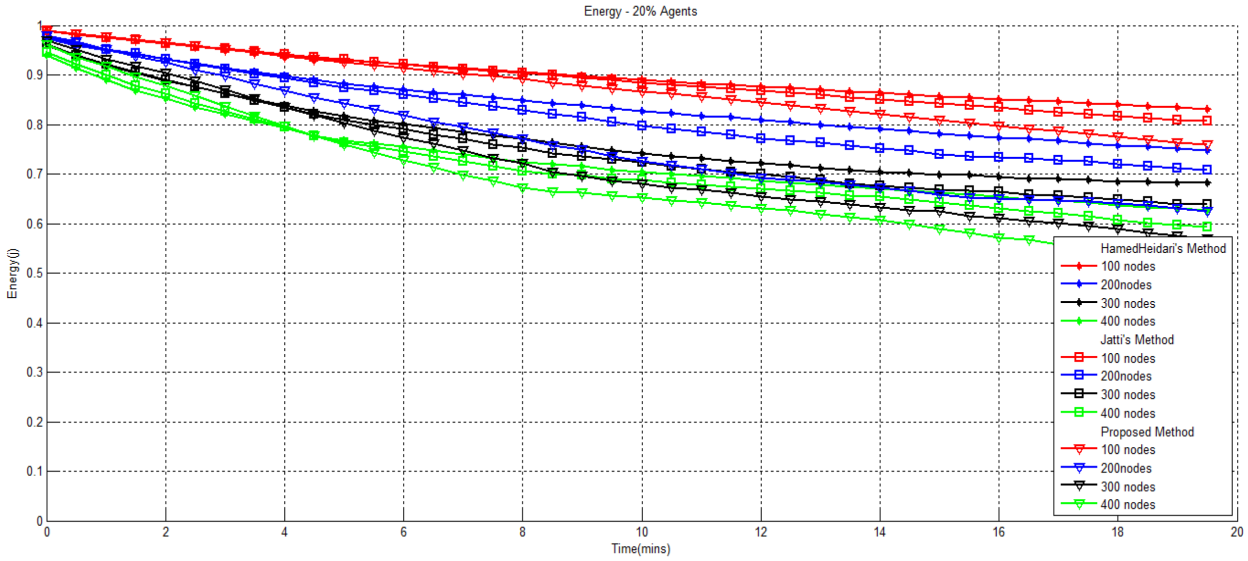


Figure 5. The energy consumption in the compared methods with 20 percent agent.

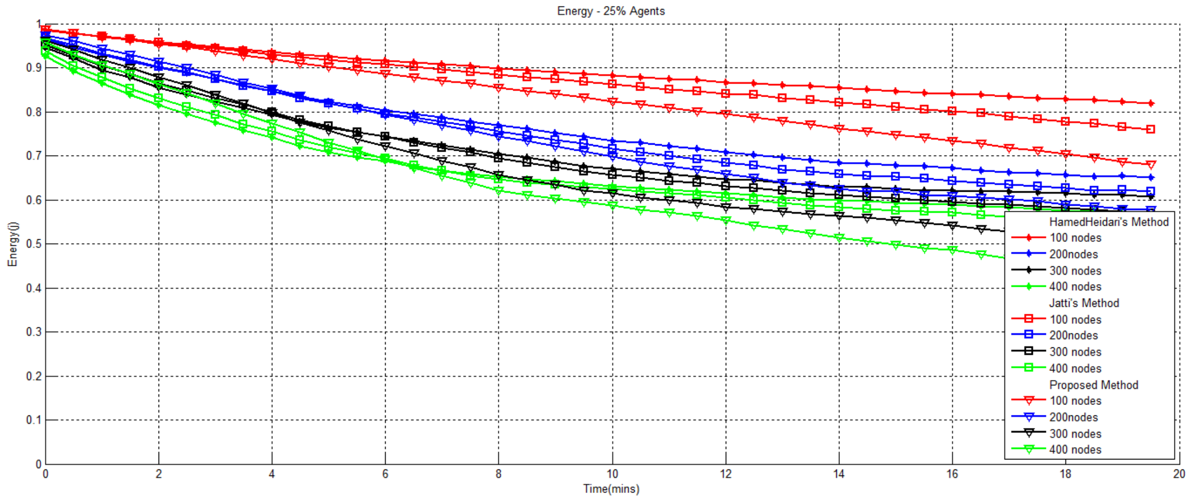


Figure 6. The energy consumption in the compared methods with 25 percent agent.

4.1.3.2 Packet Loss Rate

The most possible related problem in sinkhole attacks is packet loss. The attacker, after receiving the packets, does not transmit them. Packet loss is a vital problem in many applications. We compare the packet loss rate of our proposed method with the previous methods in Figures 7-10. The packet loss in the Hamedheidari and Jatti methods is caused by the presence of uncovered nodes for the agent. The uncovered nodes assume that all their neighbors are attackers and do not interact with them.

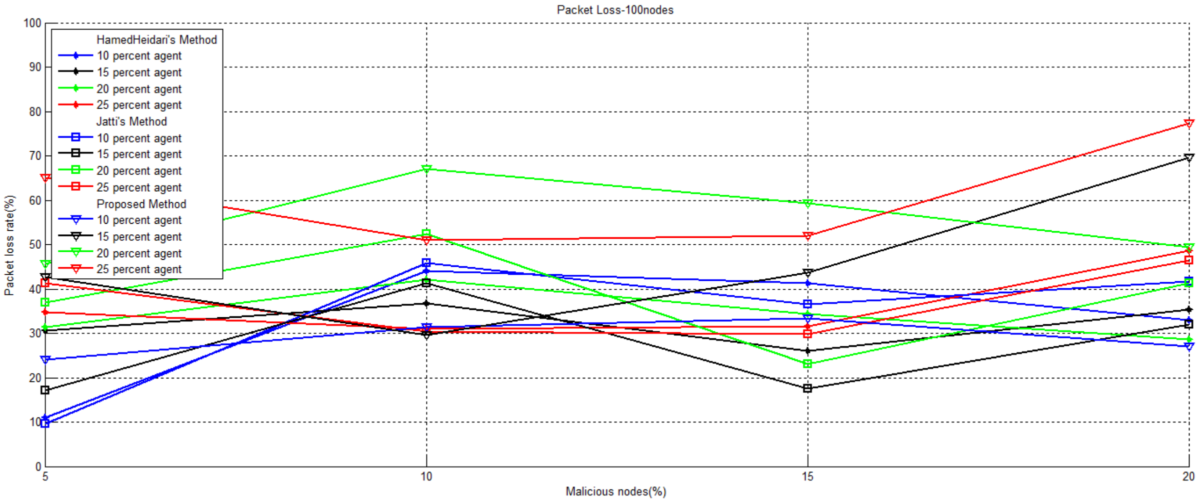


Figure 7. Comparison of the packet loss rate in the compared methods with 100 nodes in the network.

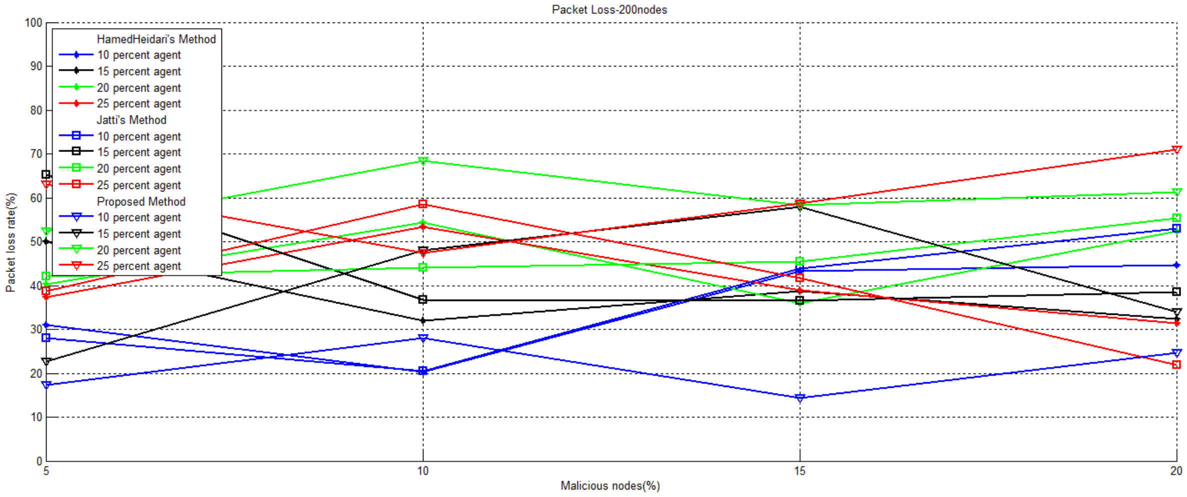


Figure 8. Comparison of the packet loss rate in the compared methods with 200 nodes in the network.

Therefore, by increasing the number of agents, the network is covered better and the packet loss rate gets reduced in the Hamedheidari and Jatti methods. Since the uncovered nodes can recognize their malicious neighbors, so in a network with a large number of nodes and a low percentage of agent nodes, the number of lost packets in the proposed method is less than in the compared methods, as shown in Figure 10. Another reason for packet loss is the end of energy of intermediate nodes in a data path. The energy consumption in the proposed method is more than the Hamedheidari and Jatti methods, so from this point of view, the number of packet losses in the compared methods is less than in the proposed method.

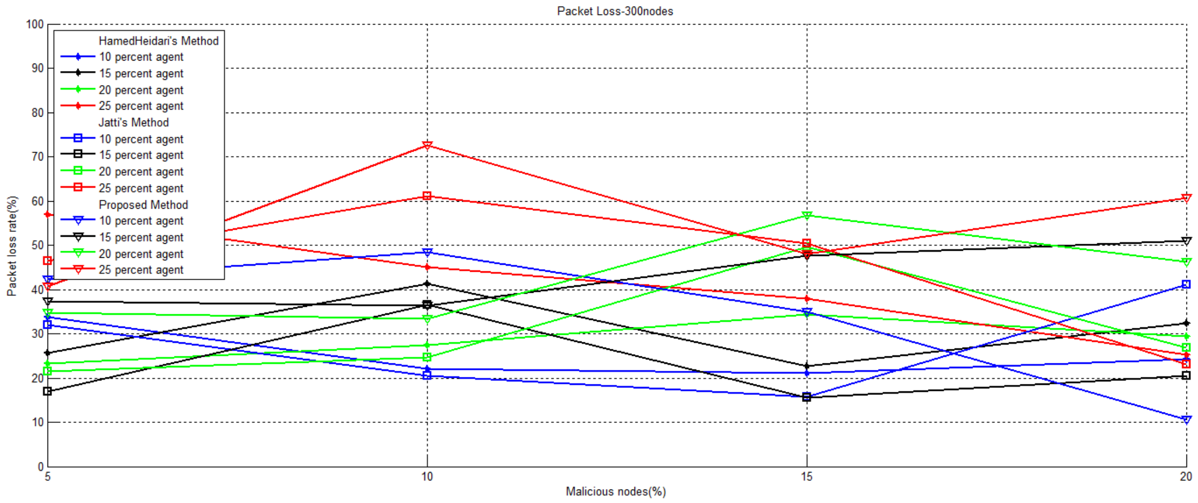


Figure 9. Comparison of the packet loss rate in the compared methods with 300 nodes in the network.

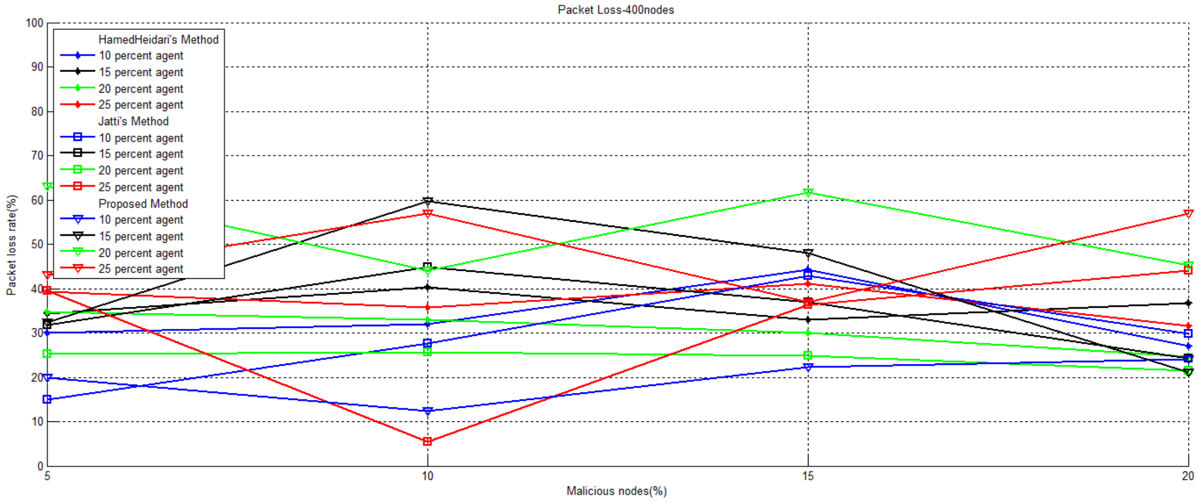


Figure 10. Comparison of the packet loss rate in the compared methods with 400 nodes in the network.

4.1.3.3 Throughput

Throughput is the average of successful message delivery in a communication channel. Since the sinkhole attack forwards the packets in the wrong paths or does not transmit them, throughput is reduced. We can measure the throughput by data packets per second or data packets per interval. Comparison of throughput with the two methods is depicted in Figures 11-14. In the Hamedheidari and Jatti methods, only the agents are responsible for detecting malicious nodes, whereas in our method, this process is done by agents and nodes. As shown, by increasing the number of agents, the compared methods have better performance than our method.

4.1.3.4 Mobile Agents' Overhead

The next criterion that we review in simulations is the average mobile agents' overhead in the network. A comparison of the average of mobile agents' overhead between the proposed method and the compared methods is shown in Figures 15-18. This criterion has been calculated by the rate of the number of

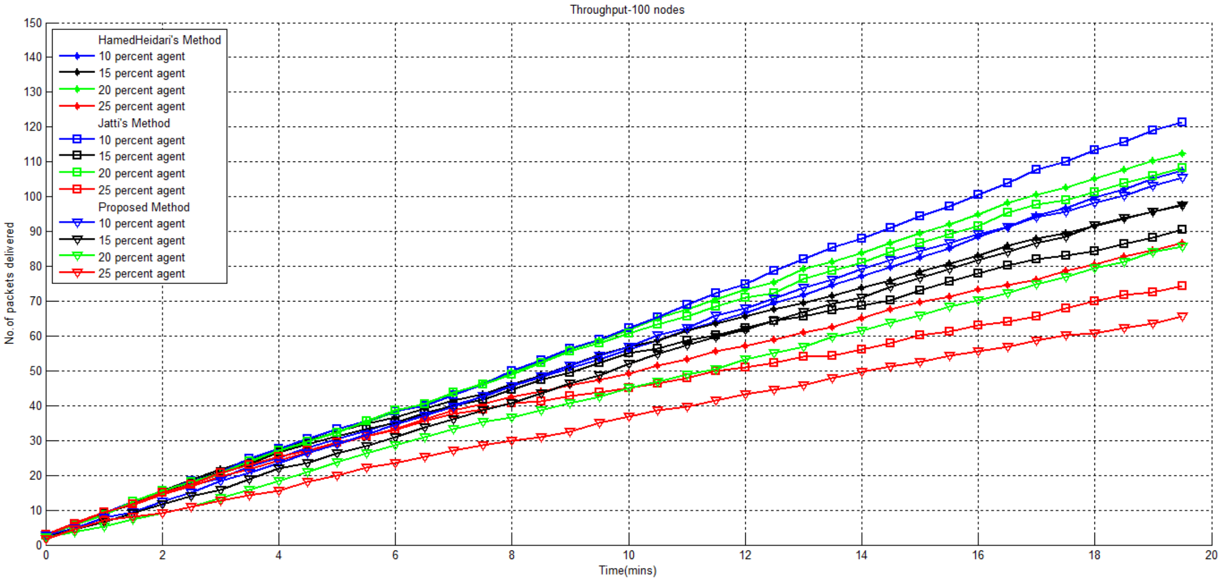


Figure 11. Comparison of throughput in the compared methods with 100 nodes in the network.

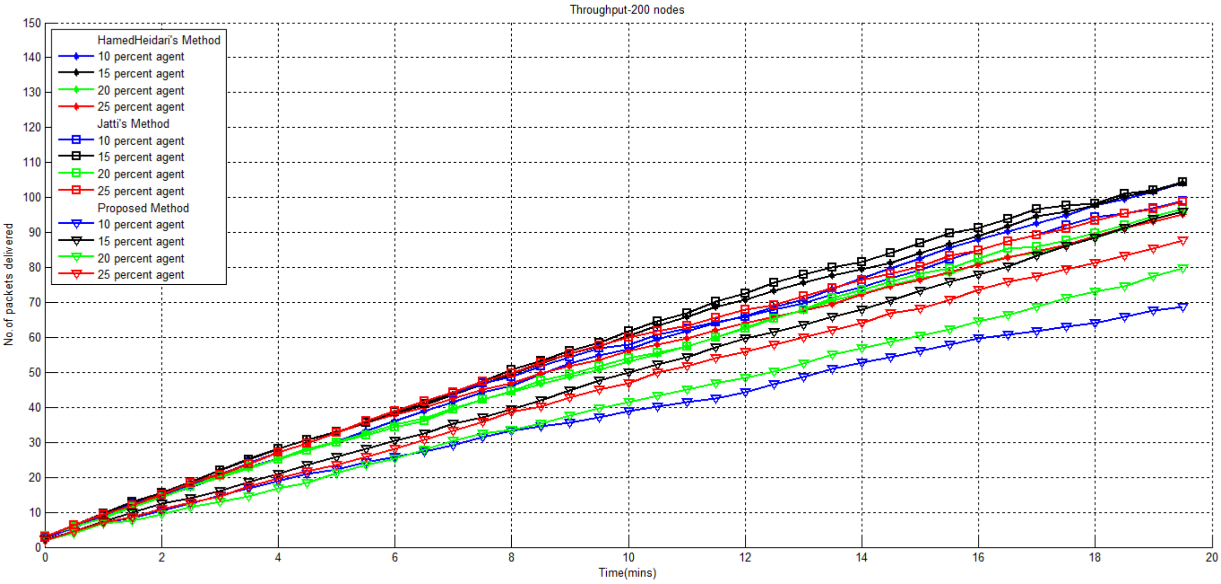


Figure 12. Comparison of throughput in the compared methods with 200 nodes in the network.

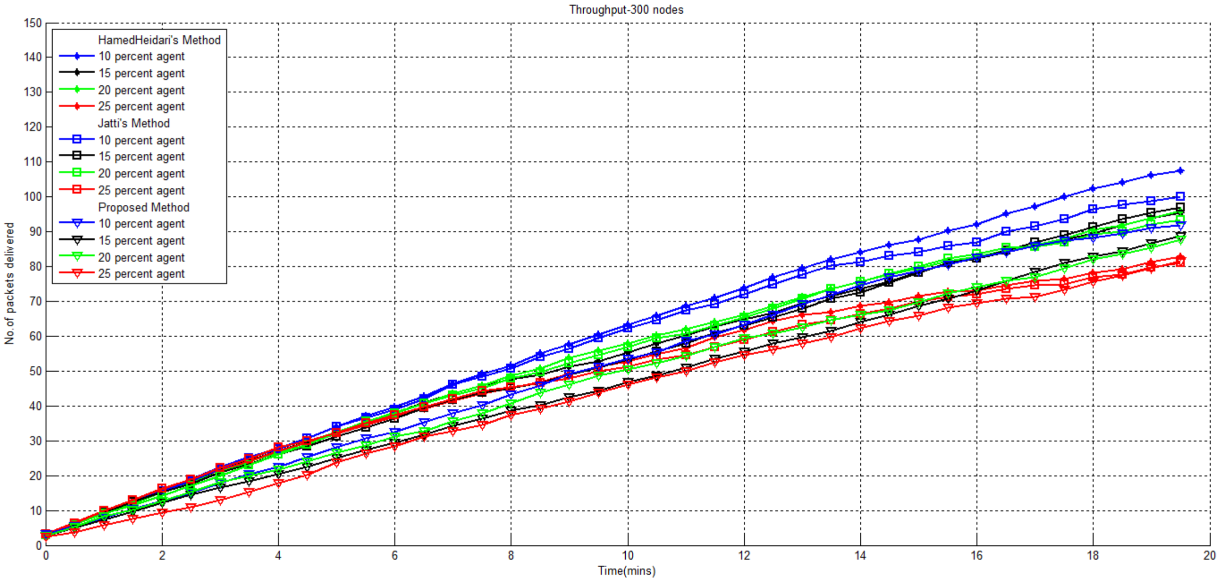


Figure 13. Comparison of throughput in the compared methods with 300 nodes in the network.

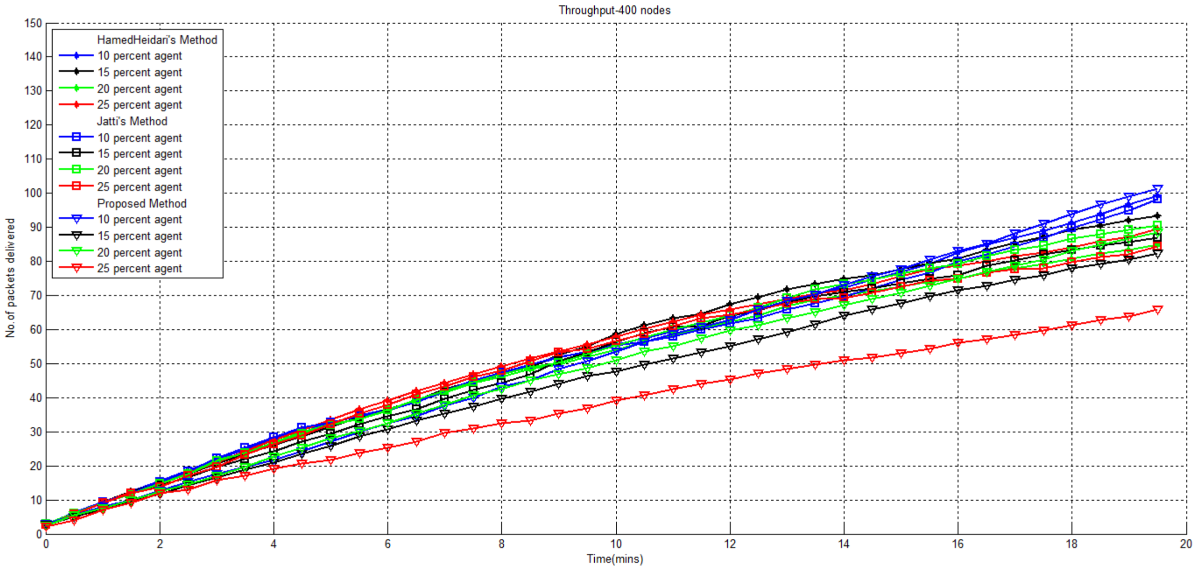


Figure 14. Comparison of throughput in the compared methods with 400 nodes in the network.

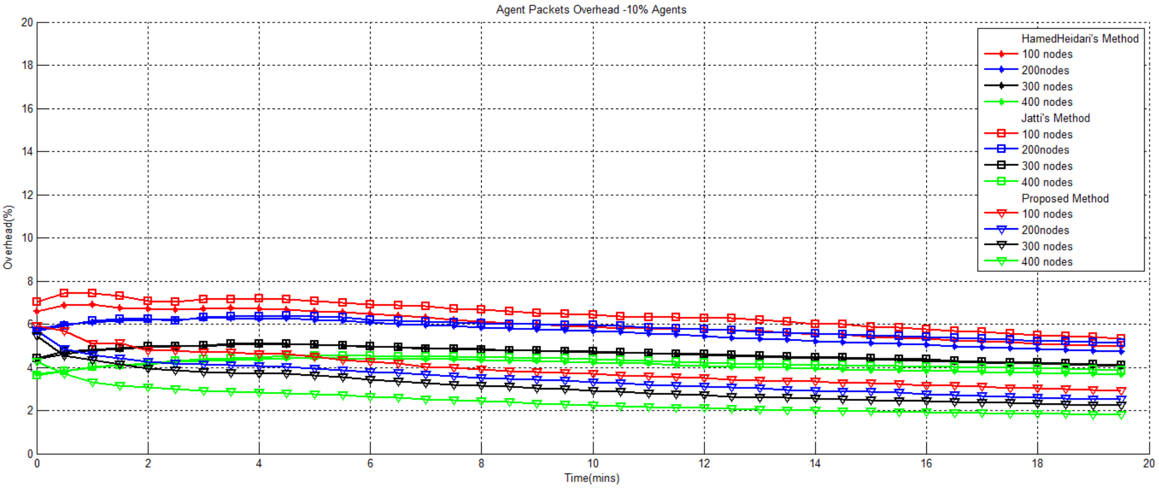


Figure 15. The mobile agents' overhead in the compared methods with 10 percent agent.

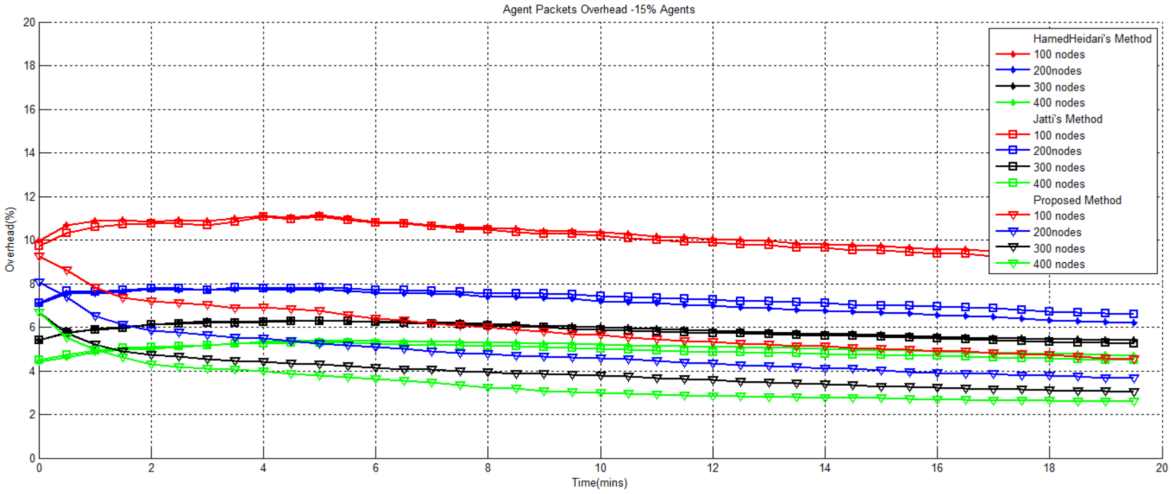


Figure 16. The mobile agents' overhead in the compared methods with 15 percent agent.

control packets). As shown, the agent overhead increases while the number of agent migrations becomes more. In other words, agents' overhead is reduced by increasing the number of sensor nodes. Moreover, there is no trust packet in our method; so, the agent's overhead is reduced more than the Hamedheidari and Jatti methods.

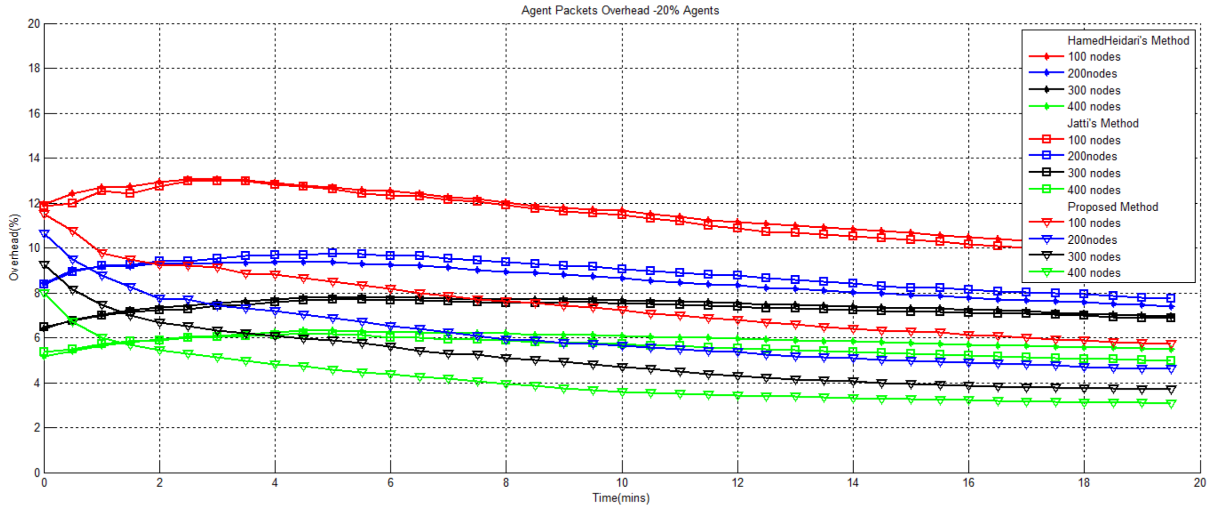


Figure 17. The mobile agents' overhead in the compared methods with 20 percent agent

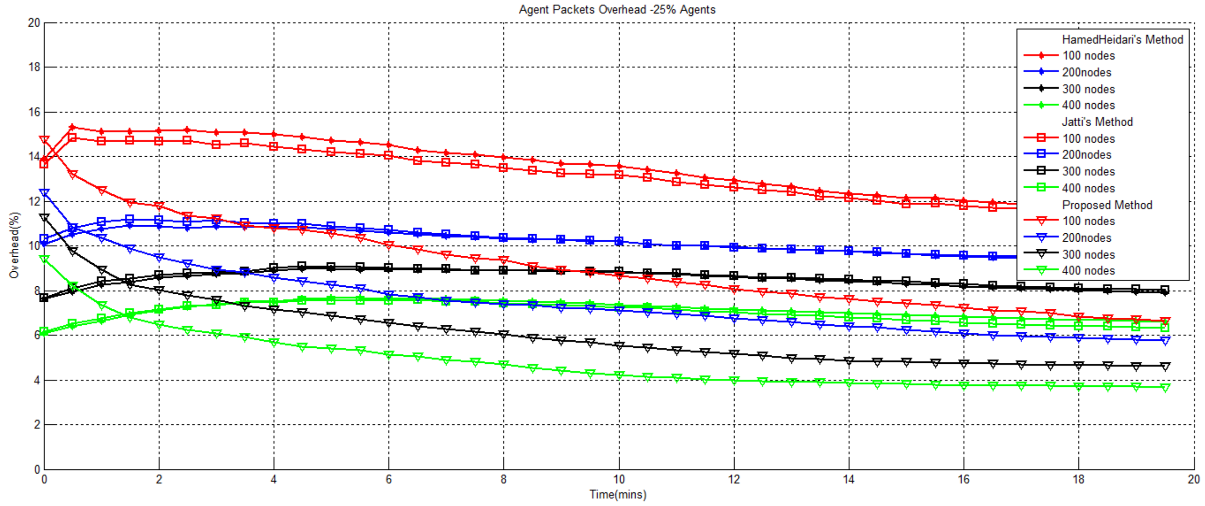


Figure 18. The mobile agents' overhead in the compared methods with 25 percent agent.

5. Conclusion and Future Work

In this research, we proposed a novel mobile agent-based technique to counter sinkhole attacks in WSNs. We carefully examined the most relevant method to the subject of this paper and issues which could challenge its security. We then presented our solution, which covered the security flaws of previous methods. The simulation results showed that our method could improve the overhead caused by the agents in the network and the packet loss ratio in comparison with the previous methods. At the same time, other criteria, such as energy consumption and throughput remained almost the same. Furthermore, our method resolved the issue of uncovered nodes in the previous methods by equipping each node to have the ability to detect adversaries on its own. In the future, we plan to apply fuzzy logic to improve the detection algorithm of the malicious nodes. Moreover, we want to extend our method to support the other routing protocols.

References

[1] M. Ali, M. Nadeem, A. Siddique, S. Ahmad and A. Ijaz, "Addressing Sinkhole Attacks in Wireless Sensor Networks: A Review," Int. Journal of Scientific & Technology Research, vol. 9, no. 8, pp. 406-411, 2020.

[2] R. Almesaeed, A. Al-Nasser and H. Al-Junaid, "A Comprehensive Survey on Routing and Security in Mobile Wireless Sensor Networks," International Journal of Electronics and Telecommunications, vol. 67, no. 3, pp. 483-496, 2021.

[3] I. Almomani and K. Sundus, "The Impact of Mobility Models on the Performance of Authentication Services in Wireless Sensor Networks," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 1, pp. 75-93, 2020.

[4] S. Aryai and G. S. Binu, "Cross Layer Approach for Detection and Prevention of Sinkhole Attack Using a Mobile Agent," Proc. of the 2nd IEEE International Conference on Communication and Electronics Systems (ICCES), pp. 359-365, DOI: 10.1109/CESYS.2017.8321299, Coimbatore, India, Oct. 2017.

[5] M. Bahekmat, M. H. Yaghmaee, A. S. H. Yazdi and S. Sadeghi, "A Novel Algorithm for Detecting Sinkhole Attacks in WSNs," International Journal of Computer Theory and Engineering, vol. 4, no. 3, pp. 418-421, 2012.

[6] J. A. Chaudhry, U. Tariq, M. A. Amin and R. G. Rittenhouse, "Dealing with Sinkhole Attacks in Wireless Sensor Networks," Advanced Science and Technology Letters, vol. 29 (SecTech 2013), pp. 7-12, 2013.

[7] H. M. A. Fahmy, Wireless Sensor Networks: Concepts, Applications, Experimentation and Analysis, 1st Ed., ISBN-13: 978-9811004117, Cairo, Springer, 2016.

[8] G. P. Gupta, M. Misra and K. Garg, "Energy and Trust Aware Mobile Agent Migration Protocol for Data Aggregation in Wireless Sensor Networks," Journal of Network and Computer Applications, vol. 41, pp. 300-311, DOI: 10.1016/j.jnca.2014.01.003, 2014.

[9] S. Hamedheidari and R. Rafeh, "A Novel Agent-based Approach to Detect Sinkhole Attacks in Wireless Sensor Networks," Computers & Security, vol. 37, pp. 1-14, DOI: 10.1016/j.cose.2013.04.002, 2013.

[10] G. Jahandoust and F. Ghassemi, "An Adaptive Sinkhole Aware Algorithm in Wireless Sensor Networks," *Ad Hoc* Networks, vol. 59, no. C, pp. 24-34, DOI: 10.1016/j.adhoc.2017.01.002, 2017.

[11] A. V. Jatti and V. K. Sonti, "Sinkhole Attack Detection and Prevention Using Agent Based Algorithm," Journal of University of Shanghai for Science and Technology, vol. 23, no. 5, pp. 526-544, 2021.

[12] G. Kalnoor, J. Agarkhed and S. R. Patil, "Agent-based QoS Routing for Intrusion Detection of Sinkhole Attack in Clustered Wireless Sensor Networks," Proc. of the 1st International Conference on Computational Intelligence and Informatics, pp. 571-583, Springer-Singapore, 2017.

[13] S. Kaur and N. Goyal, "A Survey on Security Attacks in Wireless Sensor Networks," International Journal of Advanced Research in Computer Science, vol. 7, no. 6, pp. 94-96, 2016.

[14] G. Kim, Y. Han and S. Kim, "A Cooperative-sinkhole Detection Method for Mobile *Ad Hoc* Networks," AEU-International Journal of Electronics and Communications, vol. 64, no. 5, pp. 390-397, 2010.

[15] I. Krontiris, T. Giannetsos and T. Dimitriou, "Launching a Sinkhole Attack in Wireless Sensor Networks; the Intruder Side," Proc. of the International Conference on Wireless and Mobile Computing, pp. 526-531, DOI: 10.1109/WiMob.2008.83, Avignon, France, 2008.

[16] L. Mechtri, F. T. Djemili and S. Ghanemi, "Agent-based Intrusion Detection in Wireless Networks," Implementing Computational Intelligence Techniques for Security Systems Design, pp. 97-130, DOI: 10.4018/978-1-7998-2418-3.ch005, IGI Global, 2020.

[17] O. Naderi, M. Shahedi and S. M. Mazinani, "A Trust Based Routing Protocol for Mitigation of Sinkhole Attacks in Wireless Sensor Networks," International Journal of Information and Education Technology, vol. 5, no. 7, pp. 520-526, 2015.

[18] K. E. Nwankwo and S. M. Abdulhamid "Sinkhole Attack Detection in A Wireless Sensor Networks Using Enhanced Ant Colony Optimization to Improve Detection Rate," Proc. of the 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf), pp. 1-6, Zaria, Nigeria, Oct. 2019.

[19] H. Salameh, M. Dhainat and E. Benkhelifa, "A Survey on Wireless Sensor Network-based IoT Designs for Gas Leakage Detection and Fire-fighting Applications," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 2, pp. 60-72, 2019.

[20] R. A. Shaikh, H. Jameel, B. J. d'Auriol, H. Lee, S. Lee and Y. J. Song, "Group-based Trust Management Scheme for Clustered Wireless Sensor Networks," IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 11, pp. 1698-1712, 2009.

[21] S. Sharmila and G. Umamaheswari, "Detection of Sinkhole Attack in Wireless Sensor Networks Using Message Digest Algorithms," Proc. of the IEEE International Conference on Process Automation, Control and Computing (PACC), pp. 1-6, Coimbatore, India, DOI: 10.1109/PACC.2011.5978973, 2011.

[22] D. Sheela, K. C. Naveen and G. Mahadevan, "A Non Cryptographic Method of Sink Hole Attack Detection in Wireless Sensor Networks," Proc. of the IEEE Int. Conf. on Recent Trends in Information Technology (ICRTIT), pp. 527-532, DOI: 10.1109/ICRTIT.2011.5972397, Chennai, India, 2011.

[23] G. Thirumalaimuthu, E. E. Lawrence and S. Meenakshi, "Security in Wireless Sensor Networks: Issues and Challenges," International Journal of Computer Application, vol. 6, no. 2, pp. 145-151, 2016.

[24] H. Wang, "A Three-tier Scheme for Sybil Attack Detection in Heterogeneous IWSN," Proc. of the Int.

Conf. on Computer Science Communication and Network Security (CSCNS2019), MATEC Web of Conferences, vol. 309, p. 02005, pp. 1-8, EDP Sciences, 2020.

[25] S.-H. Yang, Wireless Sensor Networks - Principles, Design and Applications, ISBN-13: 978-1447169321, London, Springer, 2014.

ملخص البحث:

شبكات المجسّات اللاسلكية تشكّل تقْنيةً مطبّقةً على نطاقٍ واسع في مجالاتٍ متعدّدة. وفيما يتعلّق بمحدِّدات شبكات المجسّات اللاسلكية، فهي تواجه العديد من الهجمات. وتعدّ هجمات "البالوعة" أكثر هذه الهجمات شيوعاً وأشدّها خطراً في تسيير شبكات المجسّات اللاسلكية. وهناك الكثير من الطّرق لمواجهة هذا النّوع من الهجمات في الأدبيّات المتعلّقة بالموضوع. وتعطي الطّرق المعتمدة على العوامل الديناميّة نتائج أفضل في مواجهة هذا النّوع من الهجمات والتغلّب على محدِّدات شبكات المجسّات اللاسلكية.

في هذه الورقة، نقدّم طريقةً تستند على العوامل الديناميّة وتستخدم قيمة الثّقة لكلّ مجسّ للكشْف عن هجمات "البالوعة" ومنْعها. نقوم بحساب قيم الثِّقة لإعلام عُقَد المجسّات عن مكاناتِ جاراتها.

وكما تبيّن التجارب، فإنّ الطّريقة المقترحة في هذه الورقة أسفرت عن نتائج أفضَل من حيث معدّل فَقْد الحُزَم. كذلك فهي تُصْلِح نواقص الأعمال السّابقة بخصوص أمْن الشّبكة، وتقلّل تكلفة العوامل في الشّبكة مقارنةً بالطّرق السّابقة.

JJCIT Annual List of Reviewers (2021)

Name, *Affiliation*, Country

Ali Lalbakhsh, *Macquarie Univ.*, Australia

Amirhossein Ghaderi, *York Univ.*, Canada

Gwan Hui Lee, *Kyungpook National Univ.*, S. Korea

Muhammad Ali, *Georgia Institute of Tech.*, USA

Alireza Ghasempour, *IUMS*, Iran

Fadi Muheidat, *California State Univ.*, USA

Lilia Tightiz, *Sejong Univ.*, S. Korea

Vinay Chamola, *PITS Pilani*, India

Temitayo O. Olowu, *Florida International Univ.*, USA

Asadullah Khalid, *Florida International Univ.*, USA

Yosra Fraiji, *ENSI*, Tunisia

Thomas E. Carroll, *PNNL*, USA

Wassim Trojet, *ESIGELEC*, France

Xiaofei Xie, *Nanyang Technological Univ.*, Singapore

Shigang Liu, *Swinburne Univ. of Tech.*, Australia

Glen Dario Rodrguez, *National Univ. of Engineering*, Peru

Tim Sonnekalb, *German Aerospace Center*, Germany

Guanjun Lin, *Swinburne Univ. of Technology*, Australia

Ali H. Majeed, *Univ. of Kufa*, Iraq

Jun-Cheol Jeon, *Kumoh NIT*, S. Korea

Mrinal Goswami, *Univ. of Petroleum and Energy Studies*, India

Jadav Chandra Das, *MAKAUT*, India

Ali Newaz Bahar, *MBSTU*, Bangladesh

Esam Alkaldy, *Univ. of Kufa*, Iraq

Mohammad Hossein Moaiyeri, *Shahid Beheshti Univ.*, Iran

Nima Jafari Navimipour, *IAU of Tabriz*, Iran

Abdalhossein Rezai, *Univ. of Science and Culture*, Iran

Soheil Sarmadi, *Univ. of South Florida*, USA

Saeid Emadi, *Olympia College*, Malaysia

Roberto Olmos Pimentel, *Benemrita Univ. de Puebla*, Mexico

Shervin Minaee, *Snap Inc.*, USA

Jonas M. Targino, *Univ. of São Paulo*, Brazil

Mohammed Beladgham, *Univ. Tahri Mohammed Bchar*, Algeria

Somayeh Sadeghi, *Univ. of Malaya*, Malaysia

Leonel Soriano-Equigua, *Univ. of Colima*, Mexico

Naveed Ali Kaim, *FUUAST*, Pakistan

Omar Longoria-Gandara, *ITESO*, Mexico

Charlotte Langlais, *Telecom Bretagne*, France

Cristiane Aparecida Lana, *Univ. of São Paulo*, Brazil

Aws A. Magableh, *Yarmouk Univ.*, Jordan

Abdelhakim Hannousse, *Univ. of 08 Mai 1945*, Algeria

Fernando Asteasuain, *UNDAV*, Argentina

Fernando Pinciroli, *Champagnat Univ.*, Argentina

Gary Allen, *Univ. of Huddersfield*, UK

Lixia Xiao, *HUST*, China

Behzad Mozaffari Tazehkand, *Univ. of Tabriz*, Iran

John G. Proakis, *UCSD*, USA

Zakariya Yahya Algamal, *Univ. of Mosul*, Iraq

Mahdieh Zabihimayvan, *CCSU*, USA

Eder S. Gualberto, *Univ. of Brasilia*, Brazil

Omar Saber Qasim, *Univ. of Mosul*, Iraq

Raffaele Della Corte, *UniNa*, Italy

Ashutosh Sharma, *Southern Federal Univ.*, Russia

Weidong Shi, *Univ. of Houston*, USA

Zhimin Gao, *AUM*, USA

Guido Governatori, *Data61 CSIRO*, Australia

Reza Fotohi, *Shahid Beheshti Univ.*, Iran

Qian Huang, *Southern Illinois Univ.*, USA

Hamid Jadad, *Dhofar Univ.*, Oman

S M Azharul Karim, *Elutions Inc.*, USA

Jorge Bernardino, *Polytehnic of Coimbra*, Portugal

Ali Shahidinejad, *Qom-IAU*, Iran

Xiaomin Jin, *XUPT*, China

Gonçalo Carvalho, *Univ. of Coimbra*, Portugal

Md Zia Ullah, *CNRS*, France

Christos Troussas, *Univ. of Piraeus*, Greece

Claude Moulin, *Sorbonne Univ.*, France

Danilo Pástor, *ESPOCH*, Ecuador

Neeraj Dhanraj Bokde, *Aarhus Univ.*, Denmark

Miao Qi, *Northeast Normal Univ.*, China

Mingliang Zhou, *Chonqqing Univ.*, China

Changgeng Yu, *Hezhou Univ. Guangxi*, China

Ruidong Chen, *GUET*, China

Ezzaki Ayoub, *Mohamed V Univ.*, Morocco

A. Uhl, *Univ. of Salzburg*, Austria

Mohd Khanapi Abd, *UTEM*, Malaysia

Ammar Awad Mutlag, *Ministry of Education*, Iraq

Mohammad Shojafar, *Univ. of Surrey*, UK

Mazin Abed Mohammed, *Univ. of Anbar*, Iraq

Humberto Jorge de Moura Costa, *UNISINOS*, Brazil

Ali Ahmadian, *UniRC*, Italy

V. J. Arulkarthick, *SNSEngg*, India

Mrinal Goswami, *NITD*, India

Jalal Rostami Monfared, *Islamic Azad Univ.*, Iran

Danil Sokolov, *Newcastle Univ.*, UK

Kawther A.Al-Dhlan, *Univ. of Hail*, KSA

Kais Haddar, *Univ. of Sfax*, Tunisia

Laurent Romary, *INRIA*, France

Ikmal Hafiz Jamal, *UiTM*, Malaysia

Mohammad Al-Ramahi, *Texas A&M Univ.*, USA

Ammar Alsalka, *Univ. of Leeds*, UK

Dirk Thorleuchter, *Fraunhofer INT*, Germany

Latifa Ben Arfa, *Tunis Univ.*, Tunisia

Christopher J. Garcia, *Univ. of Mary Washington*, USA

Vikram Bali, *JSSATE*, India

S. Khaddaj, *Kingston Univ.*, UK

Vanya Angelova Ivanova, *Univ. of PLovdiv*, Bulgaria

Boyan Zlatanov, *Univ. of PLovdiv*, Bulgaria

Aftab Ali Haider, *MAJU*, Pakistan

Wasi Haider But, *NUST*, Pakistan

Sarmad Maqsood, *KTU*, Lithuania

Guanqiu Qi, *Arizona State Univ.*, USA

MirceaFlorin Vaida, *UTC-N*, Romania

Sebelan Danishvar, *Brunel Univ.*, UK

Qian Jiang, *Yunnan Univ.*, China

Muhammad Uzair khan, *QUEST Lab*, Pakistan

Fabio Nogueira de, *Federal Univ. of Gois*, Brazil

Amro Al-Said Ahmad, *Philadelphia Univ.*, Jordan

Erik Cambria, *NTU*, Singapore

Amira Barhoumi, *Le Mans Univ.*, France

Felipe Ortega, *Univ. Rey Juan Carlos*, Spain

Hameed Hussain, *Univ. of Buner*, Pakistan

Tomasz RAK, *Rzeszow Univ. of Tech.*, Poland

Sebastião Emidio Alves, *UERN*, Brazil

Gerasimos Vonitsanos, *Univ. of Patras*, Greece

Christos Makris, *Univ. of Patras*, Greece

Beom-Su Kim, *CNU*, S. Korea

Rashid Amin, *UET*, Pakistan

Ki-Il Kim, *CNU*, S. Korea

Seba Susan, *DTU*, India

Mingxiang, *SDUST*, China

Victor H. Barella, *Univ. of São Paulo*, Brazil

Lamjed TOUIL, *Univ. of Monastir*, Tunisia

Shahram Babaie, *Islamic Azad Univ.*, Iran

Mutaz AlTarawneh, *Mutah University*, Jordan

Nuriddin Safoev, *TUIT*, Uzbekistan

Samira Sayedsalehi, *Islamic Azad Univ.*, Iran

Hongyang Ma, *QUT*, China

Siddhartha Bhattacharyya, *CHRIST*, India

Mario Mastriani, *Qubit Reset LLC*, USA

Longzhi Yang, *Northumbria Univ.*, UK

Peipei Li, *Hefei Univ. of Tech.*, China

Panida Songram, *Mahasarakham Univ.*, Thailand

Manju Khari, *GGSIP Univ.*, India

OmPrakash Vyas, *IIITs*, India

Dale G. Dzielski, *West Virginia Univ.*, USA

Seyed Hassan Mirian Hosseinabadi, *SUT*, Iran

Thomas Devine, *Fairmont State Univ.*, USA

Yasser Ali Alshehri, *RCJY*, KSA

Raed Shatnawi, *JUST*, Jordan

Tao Huang, *CAS*, China

JJCIT Annual List of Reviewers (2021)

Name, *Affiliation*, Country

Shuaiqi Liu, *Hebei Univ.*, China
Caleb Vununu, *PKNU*, S. Korea
Thanh Tran, *Mid Sweden Univ.*, Sweden
Mohammad Azam Khan, *Korea Univ.*, S. Korea
Jie Xue, *Shandong Normal Univ.*, China
Zecheng He, *Princeton Univ.*, USA
Andraž Krašovec, *Univ. of Ljubljana*, Slovenia
Domenico Ciuonzo, *UniNa*, Italy
Antonio Montieri, *UniNa*, Italy
Wahiba Ben Abdessalem, *ISG*, Tunisia
Aziz Nanthaamornphong, *Prince of Songkla Univ.*, Thailand
Jose Ruiz-Pinales, *Univ. of Guanajuato*, Mexico
Taraggy M. Ghanim, *Misr Int. Univ.*, Egypt
Rolla Almodfer, *WUT*, China
Asif Ali Laghari, *SMIU*, China
Rahul Yadav, *Harbin Institute of Technology*, China
N. Z. Jhanjhi, *Taylor's Education Group*, Malaysia
Rutuparna Panda, *VSSUT*, India
Sanjay Agrawal, *VSSUT*, India
Zhiping Tan, *Beijing Normal Univ.*, China
Yi Wang, *Rolls-Royce*, UK
Bo Lei, *XUPT*, China
Manoj Kumar Naik, *SOA*, India
Dulani Meedeniya, *Univ. of Moratuwa*, Sri Lanka
Salah Alghyaline, *WISE*, Jordan
Yanming Chen, *AHU*, China
Hendry, *UKSW*, Poland
Petr Hurtik, *Univ. of Ostrava*, Czechia
Abu Jar Md., *IUT*, Bangladesh
Ting Lan, *MUST*, China
Mohamed Ben Halima, *Univ. of Sfax*, Tunisia
Ahmad Al-Ahmad, *AUM*, Kuwait
Yehia I. Alzoubi, *AUM*, Kuwait
Chanapha Butpheng, *NDHU*, Taiwan
Ashraf Jaradat, *AUM*, Kuwait
Zahra Mogharrabi-Rad, *Islamic Azad Univ.*, Iran
Asma Benmessaoud Gabis, *ESI*, Algeria
Sarzamin Khan, *COMSATS*, Pakistan
Carmen Elena CÎRNU, *ICI*, Romania
Anwar Kalghoum, *ENSI-UMA*, Tunisia
Jaydip Sen, *PRAXIS*, India
Omid Sharifi-Tehrani, *Imam Hossein Univ.*, Iran
Yi Ou, *IME*, China
M. Irfan Khattak, *UET Peshawar*, Pakistan
Dubari Borah, *UCCS*, USA
Socheatra Soeung, *UTP*, Malaysia
Francesco Cauteruccio, *Univ. of Calabria*, Italy
Himanshu Sharma, *KIET*, India
Anastasios Doulamis, *NTUA*, Greece
Iraklis Moutidis, *Univ. of Exeter*, UK
Shahid Mumtaz, *Inst. de Telecom.*, Portugal
Jie Ding, *HUST*, China
Mahyar Nemati, *Deakin Univ.*, Australia
Axel Sikora, *Hahn-Schickard*, Germany
Mukesh Kumar Maheshwari, *BAHRIA*, Pakistan
Ali Razavi, *DeepMind*, UK
Joon Huang Chuah, *UM*, Malaysia
Erdi Callı, *Radboudumc*, Netherlands
Maksym Kholiavchenko, *RPI*, USA
Mohammad Khaleel Sallam, *NEU*, Turkey
Zuraini Othman, *UTEM*, Malaysia
Mario Versaci, *UNIRC*, Italy
Gregory Randall, *FING*, Uruguay
Xu Qin, *UESTC*, China
Mohammed Alweshah, *BAU*, Jordan
Tarik A. Rashid, *UKH*, Iraq
Alireza Goli, *Univ. of Isfahan*, Iran
Abderrahim Zannou, *USMBA*, Morocco
Xu Yu, *QUST*, China
Yongquan Zhou, *GXUN*, China
Seyedali Mirjalili, *Griffith Univ.*, Australia
Md Sipon Miah, *NUI Galway*, Ireland
Hongwei Li, *TUM*, Germany
Zahid Ullah, *PAF IAST*, Pakistan
Ying Cui, *QUST*, China
Qian Wang, *Xi'an Univ.*, China
Keivan Borna, *Kharazmi Univ.*, Iran
Minghui Wang, *QUST*, China
Syafeeza Ahmad Radzi, *UTEM*, Malaysia
Mohd. Abdul Muqeet, *MJCET*, India
Behrouz Pourghebleh, *Islamic Azad Univ.*, Iran
Saeid Seyedi, *FTRC-YUNTECH*, Taiwan
Majid Haghparast, *SRBIAU*, Iran
Peiman Ghasemi, *AZAD*, Iran
Fariba Goodarzian, *MIR Labs*, USA
Behruz Mohammadi, *Shahed Univ.*, Iran
M. Hamed Mozaffari, *UOTTAWA*, Canada
Jose-Agustin Almaraz-Damian, *Instituto Politecnico Nacional*, Mexico
Volodymyr Ponomaryov, *Instituto Politecnico Nacional*, Mexico
Andres Iglesias, *Univ. of Cantabria*, Spain
Akemi Gálvez, *Univ. of Cantabria*, Spain
Amjad Iqbal, *INRS*, Canada
Taimoor Khan, *NIT Silchar*, India
Heng Luo, *CSU*, China
Rezaul Azim, *Univ. of Chittagong*, Bangladesh
Yadgar I. Abdulkarim, *Charmo Univ.*, Iraq
Amir BENZAOUI, *Univ. of Skikda*, Algeria
Hammam Alshazly, *INB- Univ. of Lübeck*, Germany
Mechab Boubaker, *Uni-SBA*, Algeria
Chih-Peng Fan, *NCHU*, Taiwan
Yacine Khaldi, *Univ-BOUIRA*, Algeria
Abdeldjalil Ouahabi, *Univ. of Tours*, France
Silvia Parusheva, *UE-Varna*, Bulgaria
Kamarul Faizal Hashim, *UUM*, Malaysia
Khalid Mohamed Nahar, *Yarmouk Univ.*, Jordan
Yazid Bounab, *Univ. of Oulu*, Finland
Md. Alamgir Alamgir Hossain, *HSTU*, Bangladesh
Mohamed Amine Ferrag, *Univ-GUELMA*, Algeria
Farhan Sadique, *Univ. of Nevada*, USA
Olakunle Ibitoye, *Carleton Univ.*, Canada
Guanjun Liu, *Tongji Univ.*, China
Vaishali Yatish Ganganwar, *Army Institute of Tech.*, India
Joong-Hwan Baek, *Korea Aerospace Univ.*, S. Korea
Stephanie Cairns, *McGill University*, Canada
Jofrey L. Leevy, *Florida Atlantic Univ.*, USA
Sangwoo Mo, *KAIST*, S. Korea
Ravneet Kaur, *Hitachi America Ltd.*, USA
Durai Raj Vincent, *VIT*, India
David Samuel Bhatti, *SEECS*, Pakistan
Salah Eddine Benatia, *Univ-MASCARA*, Algeria
Chiyu Zhang, *Univ. of British Columbia*, Canada
Meshrif Alruily, *Jouf Univ.*, KSA
Daniela Moctezuma, *CENTROGEO*, Mexico
Dhafar Hamed Abd, *Al-Maaref Univ. College*, Iraq
Khadija Bousselmi, *Univ-SMB*, France
Georgios L. Stavrinides, *AUTH*, Greece
Mohamed Mohsen Gammoudi, *ISAMM*, Tunisia
Zaki Brahmi, *Taibah Univ.*, KSA
Mohamed Hammad, *Menofia Uni.*, Egypt
Mikhail I. Gofman, *California State Univ.*, USA
Messaoud Ramdani, *Univ-ANNABA*, Algeria
Kamal Amroun, *Univ-BEJAIA*, Algeria
M.R. Bogdanov, *USATU*, Russia
Feriel Cherifi, *Univ-BEJAIA*, Algeria
Mariusz Pelc, *Univ. of Greenwich*, UK
Alberto Antonietti, *Politecnico*, Italy
Adam Osseiran, *Edith Cowan Univ.*, Australia
Natalia Browarska, *Politechnika OPOLSKA*, Poland
Fatima Zahra GUERSS, *Ibntofail Univ.*, Morocco
Ahmad Heryanto, *Sriwijaya Univ.*, Indonesia
Rodrigo Rocha Silva, *Univ. of Coimbra*, Portugal
Lei Zhang, *Hong Kong Polytechnic Univ.*, Hong Kong
Junaid Akhtar, *NAMAL Institute*, Pakistan
Zenun Kastrati, *Linnaeus Univ.*, Sweden
Charles Nicholas, *UMBC*, USA
Liliya Akhtyamova, *MIPT/Phystech*, Russia
Oana Geman, *Stefan cel Mare Univ.*, Romania
Shapla Khanam, *Universiti Malaya*, Malaysia

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات (JJCIT) مجلة علمية عالمية متخصصة محكمة تنشر الأوراق البحثية الأصيلة عالية المستوى في جميع الجوانب والتقنيات المتعلقة بمجالات تكنولوجيا وهندسة الحاسوب والاتصالات وتكنولوجيا المعلومات.

تحتضن وتنشر جامعة الأميرة سمية للتكنولوجيا (PSUT) المجلة الأردنية للحاسوب وتكنولوجيا المعلومات، وهي تصدر بدعم من صندوق دعم البحث العلمي في الأردن. وللباحثين الحق في قراءة كامل نصوص الأوراق البحثية المنشورة في المجلة وطباعتها وتوزيعها والبحث عنها وتنزيلها وتصويرها والوصول إليها. وتسمح المجلة بالنسخ من الأوراق المنشورة، لكن مع الإشارة إلى المصدر.

الأهداف والمجال

تهدف المجلة الأردنية للحاسوب وتكنولوجيا المعلومات (JJCIT) إلى نشر آخر التطورات في شكل أوراق بحثية أصيلة وبحوث مراجعة في جميع المجالات المتعلقة بالاتصالات وهندسة الحاسوب وتكنولوجيا المعلومات وجعلها متاحة للباحثين في شتى أرجاء العالم. وتركز المجلة على موضوعات تشمل على سبيل المثال لا الحصر: هندسة الحاسوب وشبكات الاتصالات وعلوم الحاسوب ونظم المعلومات وتكنولوجيا المعلومات وتطبيقاتها.

الفهرسة

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات مفهرسة في كل من:

ELSEVIER Scopus

DOAJ DIRECTORY OF OPEN ACCESS JOURNALS | crossref Text and Data Mining | OCLC WorldCat

BASE | ACADEMIA | Google scholar

فريق دعم هيئة التحرير

عنوان المجلة

الموقع الإلكتروني: www.jjcit.org
البريد الإلكتروني: jjcit@psut.edu.jo
العنوان: جامعة الاميرة سمية للتكنولوجيا، شارع خليل الساكت، الجبيهة، عمان، الأردن.
صندوق بريد: 1438 عمان 11941 الأردن
هاتف: +962-6-5359949
فاكس: +962-6-7295534

Princess Sumaya University for Technology
جامعة الأميرة سميّة للتكنولوجيا

صندوق دعم البحث العلمي
Scientific Research Support Fund

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات

كانون الأول ٢٠٢١ المجلد ٧ العدد ٤
ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

JJCIT

jjcit@psut.edu.jo
www.jjcit.org

مجلة علمية عالمية متخصصة محكمة
تصدر بدعم من صندوق دعم البحث العلمي