Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 2, April 2026, pp. 1231∼1238
ISSN: 2302-9285, DOI: 10.11591/eei.v15i2.9649

Challenges in applying DeepInsight for cyber threat detection

Malik AL-Essa[1], Mohammad Qatawneh[1,2], Nidal Turab[2], Yazeed Alsarhan[3]
[1]Department of Computer Science, King Abdullah II School of Information Technology, The University of Jordan, Amman, Jordan
[2]Department of Networks and Cybersecurity, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan
[3]Department of Computer Science, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan

Article Info

Article history:

Received Dec 5, 2024
Revised Aug 11, 2025
Accepted Feb 22, 2026

Keywords:

Cyber threat
DeepInsight
Deep learning
Intrusion detection
Malware detection

ABSTRACT

As the world suffers from intrusions and malware extensively nowadays, intrusion detection systems (IDS) play a critical role in protecting cyberspace from attacks. However, attacks become more complex every day, leading to the necessity of developing new techniques that can protect our digital infrastructure from cyber-attacks. Deep learning (DL) is one of the techniques that are investigated to fight against cyber-attacks. However, due to the nature of traffic data, most of the techniques focus on the deep neural network (DNN) as the performance of the DNN depends on the training data. In this paper, we investigate the effectiveness of using convolutional neural networks (CNN) to detect malware apps and network intrusions. The cybersecurity datasets are converted from tabular data into images using the DeepInsight technique. Experiments are conducted using two datasets, NSL-KDD and CICMaldroid20 datasets. The proposed method demonstrates that converting cybersecurity datasets from tabular data into images may decrease the model's accuracy. Furthermore, this approach introduces additional challenges in the detection of network intrusions and malware. Moreover, the added architectural complexity may cause a dilution or distortion of feature representations, making it harder for the model to preserve the original semantic meaning of critical features.

Corresponding Author:

Malik AL-Essa
Department of Computer Science, King Abdullah II School of Information Technology
The University of Jordan
Amman, Jordan
Email: m.alessa@ju.edu.jo

1. INTRODUCTION

Over the last decade, deep learning (DL) has been attracted a lot of attention as a powerful tool of the artificial intelligence (AI) paradigm to be used cybersecurity problems such as intrusion detection and malware detection where DL achieves advanced classification capabilities [1]-[3]. Intrusion detection systems (IDS) can be classified into two main classes: signature-based intrusion detection systems (SIDS) [4] and anomaly-based intrusion detection system (AIDS) [5]. SIDSs work by matching the network traffic with previous intrusions to detect malicious activity. In other words, SIDSs are based on the previous intrusion signatures in the database to detect intrusions, so, when there is a match with a previous intrusion signature, an alarm is triggered. On the other side, AIDS uses machine learning (ML) or DL to identify anomalies in network traffic. ML and DL models are trained in order to be able to differentiate between normal and abnormal behaviors in network traffic,

therefore, any significant deviation between the observed behavior and the model prediction is considered an anomaly [5]. Various ML and DL techniques have been proposed in the literature to detect cyber-threats, however, as the cybersecurity datasets are tabular data, CNNs can not be used with cybersecurity data to detect malware apps and intrusion in cybersecurity datasets. One of the solutions is to convert the tabular data into images using various algorithms proposed in the literature. DeepInsight [6] is considered one of the well-known techniques to convert tabular data into images, it will be used in this work to convert cybersecurity datasets into images to explore the performance of CNNs in the detection of malware apps and intrusions.

Despite the promising results reported for image-based encodings of tabular data, prior DeepInsight-based studies in cybersecurity often remain limited to simplified settings and report mainly aggregate accuracy, leaving the behavior under multi-class and severely imbalanced datasets insufficiently examined. Accordingly, this paper is framed as a challenge-oriented critical analysis that identifies the key limitations and failure modes of applying DeepInsight in realistic IDS/malware scenarios, rather than presenting a performance-only evaluation.

The paper is organized as follows: section 2 reviews related works. The proposed methodology is discussed in section 3. The results of the evaluation are discussed in section 4. Finally, section 5 refocuses on the purpose of the research, draws conclusions, and illustrates future developments.

2. RELATED WORK

IDS is one of the techniques used to protect our data and network infrastructure from cyber threats. Intrusion is any unauthorized activity that can cause damage to the information system by threatening the confidentiality, integrity, and availability of the information [5]. IDSs can be categorized into SIDSs, AIDSs, and hybrid-based IDSs [5]. SIDSs achieve good results in detection accuracy for previously known intrusions [7]. However, they suffer from limitations in detecting intrusions, as they are not able to detect unknown attacks because there are no previous signatures for those attacks in the database to be matched. Different ML techniques are proposed by [8] to detect attacks against personal medical devices. Four ML techniques are used in [8]: KNN, SVN, RF, and DT. Due to the rapid development of DL techniques, they have been used extensively currently in IDS. An IDS that uses CNNs is proposed in [9]. The idea in [9] is to convert traffic data into images and then feed the images to a CNN model to detect the intrusions. The residual neural network (ResNet) [10] is also used for classifying network traffic [11], [12]. An unsupervised DL-based approach for intrusion detection is proposed in [13], where the autoencoder architecture is used. A DL model that is based on the Restricted Boltzmann Machine (RBM) in cyber security intrusion detection in large-scale smart grids is used in [14].

The work proposed in [15] integrates AI, honeypots, and intrusion prevention system (IPS) to detect these attacks, alert the attacker, and limit their session to detect cyber-threats. An IDS proposed in [16] that is based on feature selection and traditional machine learning algorithms. The work in [17] instigates the strengths of six distinct DL algorithms: deep neural networks (DNN), CNN, RNN, long short-term memory (LSTM), GRU, and a hybrid CNN-LSTM architecture in detecting cyber-threats, where both binary and mulit-class classification is investigated in this work. As the internet of things (IoT) continues to grow in popularity, it has become an increasingly common target for cyber-attacks. The study in [18] proposed a hybrid intrusion detection model that combines ML and DL techniques to enhance IoT cybersecurity, where different cyber-security benchmark datasets were used to evaluate the proposed method. Another IoT-related study presented in [19] addresses the challenge of adapting to emerging threats and the lack of consideration for IoT-specific complexities.

Qaddos *et al.* [19] proposed an innovative approach by hybridizing convolutional neural networks (CNN) and gated recurrent units (GRU), specifically designed for IoT intrusion detection. Wireless sensor networks (WSNs), a fundamental component of modern wireless technology, offer cost-effective solutions for various monitoring applications; however, they are vulnerable to numerous security threats, including unauthorized access, attacks, and suspicious activities. The work in [20] evaluates the use of different techniques, i.e., CNN, LSTM, and DNN in protecting WSNs from cyber threats. A comprehensive review of research studies focused on IDS was conducted in [21], offering valuable insights into the methodologies, datasets, and evaluation metrics commonly used in the field. In addition, [21] highlights current trends, challenges, and potential directions for future research in IDS development and deployment.

DeepInsight is proposed in [22] to convert network traffic into images. We worked with multi-class

detection, while the work in [22] dealt with binary classifications.

3. METHOD

The dataset in this work is represented as $\mathcal{D} = \{(\mathbf{x}_j, y_j)\}_{i=1}^{K}$ of K training samples. $\mathbf{x} \in \mathbf{D} \subseteq \mathbb{R}^d$ is a d-dimensional vector of input features that describe cyber-data samples (e.g., malware apps or network traffic flow traces), whereas $y \in Y$ is the value of the class variable Y that may assume N different classes where: *benign* samples and different types of *attacks*, depending on those historically collected and classified. DeepInsight is used to convert the cybersecurity dataset samples into images. The methodology of this work is defined in four-stepped for cyber-threat detection:

S1: to convert malware samples and network traffic flow traces into 2D images using DeepInsight called D'.
S2: to train a DL model $M_\delta : \mathbb{R}^d \mapsto Y$ with parameter δ estimated from $\mathcal{D}$.
S3: to train a DL model $M'_\delta : \mathbb{R}^d \mapsto Y$ with parameter δ' estimated from $\mathcal{D}'$.
S4: to compare the results of step S2 and S3.

In step S1, a new dataset is generated by transforming the original tabular data into 2D images using the DeepInsight method, which enables the application of computer vision techniques to non-image data. Steps S2 and S3 involve training DL models separately on the original tabular dataset and the image-converted dataset, respectively. Finally, in step S4, a comprehensive comparative analysis is performed to evaluate the performance and effectiveness of both approaches, highlighting the strengths and limitations of applying image-based learning to tabular data.

3.1. DeepInsight

DeepInsight [6] is a technique that allows using CNN on data that is not originally in image format. It achieves this by converting the data into a well-organized image representation. Furthermore, DeepInsight allows CNNs to automatically learn important features directly from non-image data, achieving good performance in various tasks. Deepinsight utilized t-distributed stochastic neighbor embedding (t-SNE) [23] (along with other techniques) to convert tabular data into images. t-SNE is primarily used within DeepInshight for dimensionality reduction. Malware samples often contain hundreds of interrelated features, as well as traffic network data. DeepInsight excels at converting these samples into images, effectively reducing the data dimensionality while preserving the crucial relationships among features.

4. EMPIRICAL EVALUATION AND RESULTS

4.1. Datasets

Two datasets are used in the evaluation of this work. The first one is a network traffic security dataset, i.e., NSL-KDD. The second dataset is an Android malware dataset called CICMaldroid20 dataset. Table 1 shows the descriptions for each one of them.

Table 1. Dataset description

Dataset	#Training set	#Testing set	#Classes	#Features
NSL-KDD	25192	22544	5	41
CICMalDroid20	8118	3480	5	41

The NSL-KDD dataset [24] is an old dataset that has been used for many years in the cybersecurity field research, however, it is still used in the literature. The NSL-KDD data set is a multi-class dataset that consists of benign network flow traces and four attack classes: denial of service (DoS), user to root (U2R), remote to local (R2L), and probing attacks. The dataset is split up into training and testing datasets. 21 different attack sub-categories are included in the training dataset, and 37 different sub-categories are included in the testing dataset, which means that there are 16 different novel attacks in the testing dataset. The NSL-KDD dataset has 41 different features (3 categorical features, 37 numeric features, and 1 class feature). Detailed descriptions of these features are provided in [24]. The dataset is considered an imbalanced dataset with two rare classes, i.e., U2R and R2L. For the experiment with this dataset, the NSL-KDD used in this work is the dataset that includes KDDTrain+20% as the training set and KDDTest+ as the testing set [24].

The CICMalDroid20 dataset [25] consists of samples of recent Android apps collected from various sources. It includes five categories: Adware, Banking malware, SMS malware, Riskware, and Benign. As

detailed in [25], each app is represented by 470 features capturing both static and dynamic characteristics extracted using CopperDroid. The CICMalDroid20 dataset is a balanced dataset. For this study, we employed a stratified split of the dataset, using (70%) for the training set and (30%) for the testing set.

4.2. Implementation details

Python 3.9 using Keras 2.7 library are used to implement the methods considered in this evaluation. Two DL architectures are used in this work. In the first one, for each dataset we optimized the hyper-parameter of the neural networks as reported in Table 2 using the tree-structured Parzen estimator algorithm, as implemented in the Hyperopt library, by utilizing 20% of the training set as a validation set. Validation loss is used to choose the best configuration of the selected parameters.

Table 2. The search space of the hyper-parameter

Hyper-parameter	Values
Learning rate	[0.0001, 0.001]
Mini-batch size	$\{2^5, 2^6, 2^7, 2^8, 2^9\}$
Drop-out	[0,1]
# of neurons per hidden layer	$\{2^5, 2^6, 2^7, 2^8, 2^9, 2^{10}\}$

Table 3 shows the layers selected for each neural network architecture. The architecture of the model trained using tabular data consists of three dense layers, furthermore one drop-out layer, and one batch-normalization layer are used to prevent overfitting. The second architecture, i.e., CNN model, consists of 2D CNNs layers, Two MAXPooling2D layers, one Flatten layer, and One drop-out layer to prevent overfitting. In the last layer for both of the architectures, the softmax activation function is used to output the probabilities for each target class, and the ReLU activation function was used in all the other hidden layers. We utilized the Adam update rule for gradient-based optimization and initialized the weights using the Xavier scheme. Training was limited to a maximum of 150 epochs, with an early stopping strategy to select the best models by minimizing validation loss across both neural network architectures.

Table 3. Neural network architectures used in this work

Architecture	Layer
DNN	Three fully-connected layers
	One drop-out layer
	One batch-normalization layer
CNN	2D CNN layers
	One drop-out layer
	Two MaxPooling2D layers
	One flatten layer

4.3. Empirical evaluation and discussion

We evaluated the proposed method in this work using data cybersecurity datasets, i.e., CICMaldroid20 and NSL-KDD. The experiments in this work aims to investigate the effectiveness of converting malware apps and network traffic into images in order to use CNNs as a classifier to detect malicious apps and traffic.

4.4. Performance metrics

We measured standard multi-class classification metrics on the classifications produced on the testing sets. WeightedF1, MacroF1, and OA that measure the overall accuracy are used in this work for evaluating the conducted experiments. To focus on assessing the number of true positives, OA is used in this work, while to evaluate false negatives and false positives, in addition to the true positives, F1-based scores are used. Moreover, F1 can be measured per class. So, we consider WeightedF1 and MacroF1 to aggregate F1 values measured on all the classes in a single value, by computing the weighted mean and the simple mean, respectively. In balanced domains, WeightedF1 and MacroF1 are expected to measure close values, while on the other side, i.e., imbalanced domains, WeightedF1 may return a misleading evaluation of performance of rare classes due to the prominence of the majority classes in the metric. As our study includes both balanced (CICMaldroid20 dataset) and imbalanced datasets (NSL-KDD dataset), we analyzed the accuracy performance of the proposed evaluation along all these metrics in the experimentation.

4.5. Results and discussions

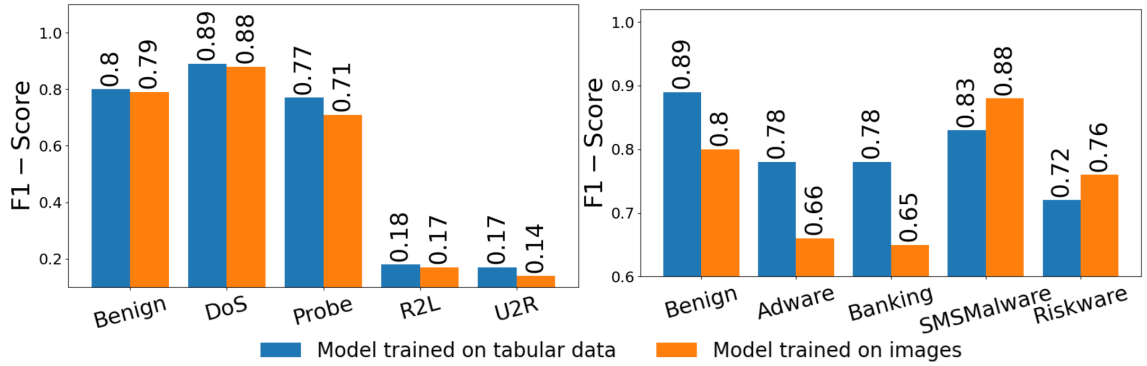
The empirical evaluation was done to explore to what extent converting malware apps and network traffic traces can influence the accuracy of the classification model. We measured the accuracy of the performance of M_δ model by learning the model using the tabular data set D and M'_δ by learning the model using the image dataset, i.e., the dataset that has been converted from tabular data into images using DeepInsight as illustrated in section 3. Across all experiments, we evaluated standard multi-class accuracy metrics i.e., WeightedF1, MacroF1, and overall accuracy OA - using the testing set of each dataset.

Table 4 shows the results of each model. We note that in NSL-KDD, the OA gained using tabular data to train the model is higher than when using data converted into images to train the model. In all classes, there was a decrease across all the metrics where WeightedF1 decreased from 0.74 to 0.73, MacroF1 decreased from 0.56 to 0.54, and OA decreased from 0.78 to 0.76. CICMaldroid20 has similar results to NSL-KDD, where the WeightedF1 decreased from 0.79 to 0.78, MacroF1 decreased from 0.80 to 0.75, and OA decreased from 0.80 to 0.77.

Table 4. M_δ and M'_δ metrics

Method	NSL-KDD			CICMaldroid20		
	WeightedF1	MacroF1	OA	WeightedF1	MacroF1	OA
M_δ	0.74	0.56	0.78	0.79	0.80	0.80
M'_δ	0.73	0.54	0.76	0.78	0.75	0.77

Figure 1(a) shows the per-class F1-scores for NSL-KDD, where the F1-score decreases across all classes when using image-converted data. Figure 1(b) shows the per-class F1-scores for CICMaldroid20, where two classes (SMSMalware and Riskware) improve while the remaining classes decline or remain close. We believe this difference is due to dataset characteristics: NSL-KDD is highly imbalanced, whereas CICMaldroid20 is relatively balanced. Overall, these results suggest that DL performance can be partially preserved in balanced settings, while imbalanced datasets still lead to weaker malicious-behavior detection due to rare attack classes.



(a) (b)

Figure 1. Per-class F1-scores on the test set for the model trained on tabular data (M_δ) and the model trained on image-converted data (M'_δ), with; (a) NSL-KDD and (b) CICMaldroid20

The use of DeepInsight to convert cybersecurity datasets into images to use the powerful capabilities of CNNs will not help improve the performance of the model when dealing with multi-class classification. On the other hand, in cybersecurity, understanding which features (e.g., packet size, protocol type, and IP addresses) indicate malicious behavior is critical for analysts to validate alerts and build trust in detection systems. DeepInsight works by grouping similar components together where it uses a dimensionality reduction algorithm, i.e., t-SNE to reduce the dimensionality of the tabular data. The problem is that when converting tabular data into images, it will not be possible to identify and interpret the features of the malware app or cybersecurity data-trace. As DeepInsight is considered a technique to transform tabular data into images to use the powerful capabilities of CNN, it still has challenges to be used. One of the main challenges is that transforming tabular data, i.e., cyber-security data, into images could result in losing information related to the

type of attacks. Attacks in cyber-data depend on some features, so converting them into images will make those features vague. Another challenge is the computation complexity due to the use of DeepInsight. Converting tabular data will require additional computation complexity.

5. CONCLUSION

This study investigated whether converting cybersecurity tabular data into images via DeepInsight improves multi-class threat detection, which directly addresses the gap identified in the introduction regarding the limited evidence for realistic multi-class and imbalanced settings. Empirically, the image-converted pipeline did not improve performance on either dataset: on NSL-KDD, WeightedF1 decreased from 0.74 to 0.73, MacroF1 decreased from 0.56 to 0.54, and OA decreased from 0.78 to 0.76; on CICMaldroid20, WeightedF1 decreased from 0.79 to 0.78, MacroF1 decreased from 0.80 to 0.75, and OA decreased from 0.80 to 0.77.

The main contributions of this work are threefold. First, we provide a challenge-oriented evaluation of DeepInsight for cybersecurity classification across both imbalanced (NSL-KDD) and balanced (CICMaldroid20) datasets using consistent multi-class metrics. Second, we show that image conversion can weaken class-level discrimination, especially for imbalanced attack distributions, even when CNN-based models are applied. Third, we highlight practical limitations beyond accuracy, including reduced feature interpretability and additional computational overhead. Overall, these findings clarify the trade-offs of applying DeepInsight in IDS and malware analysis and provide a more coherent evidence base for future work on robust and explainable cybersecurity detection models.

FUNDING INFORMATION

Not applicable.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Malik AL-Essa	✓	✓	✓	✓	✓		✓		✓	✓				
Mohammad Qatawneh	✓	✓			✓					✓	✓			
Nidal Turab						✓			✓		✓	✓		
Yazeed Alsarhan	✓								✓			✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data is openly available in a public repository.

REFERENCES

[1] A. H. Salem, S. M. Azzam, O. E. Emam, and A. A. Abohany, "Advancing cybersecurity: a comprehensive review of AI-driven detection techniques," *Journal of Big Data*, vol. 11, 2024, doi: 10.1186/s40537-024-00957-y.

[2] J. Prümmer, T. van Steen, and B. van den Berg, "A systematic review of current cybersecurity training methods," *Computers & Security*, vol. 136, pp. 1-20, 2024, doi: 10.1016/j.cose.2023.103585.

[3] M. Al-Essa and A. Appice, "Dealing with imbalanced data in multi-class network intrusion detection systems using xgboost," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2021, pp. 5-21, doi: 10.1007/978-3-030-93733-1_1

[4] S. Einy, C. Oz, and Y. D. Navaei, "The anomaly-and signature-based IDS for network security using hybrid inference systems," *Mathematical Problems in Engineering*, vol. 2021, no. 1, pp. 1-10, 2021, doi: 10.1155/2021/6639714.

[5] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1-22, 2019, doi: 10.1186/s42400-019-0038-7.

[6] A. Sharma, E. Vans, D. Shigemizu, K. A. Boroevich, and T. Tsunoda, "DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture," *Scientific Reports*, vol. 9, no. 1, pp. 1-7, 2019, doi: 10.1038/s41598-019-47765-6.

[7] C. Kreibich and J. Crowcroft, "Honeycomb: creating intrusion detection signatures using honeypots," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 1, pp. 51–56, 2004, doi: 10.1145/972374.972384.

[8] AKM I. Newaz, A. K. Sikder, L. Babun, and A. S. Uluagac, "Heka: A novel intrusion detection system for attacks to personal medical devices," in *2020 IEEE Conference on Communications and Network Security (CNS)*, Avignon, France, 2020, pp. 1-9, doi: 10.1109/CNS48642.2020.9162311.

[9] S. Z. Lin, Y. Shi, and Z. Xue, "Character-level intrusion detection based on convolutional neural networks," in *2018 International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro, Brazil, 2018, pp. 1-8, doi: 10.1109/IJCNN.2018.8488987.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[11] H.-K. Lim, J.-B. Kim, J.-S. Heo, K. Kim, Y.-G. Hong, and Y.-H. Han, "Packet-based network traffic classification using deep learning," in *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Okinawa, Japan, 2019, pp. 046-051, doi: 10.1109/ICAIIC.2019.8669045.

[12] J. Xue, Y. Chen, O. Li, and F. Li, "Classification and identification of unknown network protocols based on CNN and T-SNE," in *2nd International Conference on Electronic Engineering and Informatics*, 2020, vol. 1617, doi: 10.1088/1742-6596/1617/1/012071.

[13] F. Farahnakian and J. Heikkonen, "A deep auto-encoder based approach for intrusion detection system," in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, 2018, pp. 178-183, doi: 10.23919/ICACT.2018.8323688.

[14] H. Karimipour, A. Dehghantanha, R. M. Parizi, K.-K. R. Choo, and H. Leung, "A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids," *IEEE Access*, vol. 7, pp. 80778–80788, 2019, doi: 10.1109/ACCESS.2019.2920326.

[15] M. M. Abualhaj, S. Al-Khatib, A. Al-Allawee, A. Munther, and M. Anbar, "Enhancing Network Intrusion Detection Systems Through Dimensionality Reduction," in *Proceedings of the Sixth International Conference on Soft Computing and Data Mining (SCDM) 2024*, pp. 244–253, 2024, doi: 10.1007/978-3-031-66965-1_24.

[16] M. Abualhija, N. Al-Shaf'i, N. M. Turab, and A. Hussein, "Encountering social engineering activities with a novel honeypot mechanism," *International Journal of Electrical & Computer Engineering*, vol. 13, no. 6, pp. 7056-7064, 2023, doi: 10.11591/ijece.v13i6.pp7056-7064.

[17] S. Elsayed, K. Mohamed, and M. A. Madkour, "A comparative study of using deep learning algorithms in network intrusion detection," *IEEE Access*, vol. 12, pp. 58851–58870, 2024, doi: 10.1109/ACCESS.2024.3389096.

[18] M. Sajid *et al.*, "Enhancing intrusion detection: a hybrid machine and deep learning approach," *Journal of Cloud Computing*, vol. 13, pp. 1-24, 2024, doi: 10.1186/s13677-024-00685-x.

[19] A. Qaddos, M. U. Yaseen, A. S. Al-Shamayleh, M. Imran, A. Akhunzada, and S. Z. Alharthi, "A novel intrusion detection framework for optimizing IoT security," *Scientific Reports*, vol. 14, pp. 1-22, 2024, doi: 10.1038/s41598-024-72049-z.

[20] H. Sadia *et al.*, "Intrusion detection system for wireless sensor networks: A machine learning based approach," *IEEE Access*, vol. 12, pp. 52565–52582, 2024, doi: 10.1109/ACCESS.2024.3380014.

[21] A. Arshad, M. Shaan, H. F. Ahmed, and M. Iqbal, "Exploring Secure Processing Architecture: A Comprehensive Review," *Dialogue Social Science Review (DSSR)*, vol. 3, no. 1, pp. 476–491, 2025.

[22] T. P. Tran, V. C. Nguyen, L. Vu, and Q. U. Nguyen, "DeepInsight-convolutional neural network for intrusion detection systems," in *2021 8th NAFOSTED Conference on Information and Computer Science (NICS)*, Hanoi, Vietnam, 2021, pp. 120-125, doi: 10.1109/NICS54270.2021.9701572.

[23] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579-2605, 2008.

[24] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada, 2009, pp. 1-6, doi: 10.1109/CISDA.2009.5356528.

[25] S. Mahdavifar, D. Alhadidi, and A. A. Ghorbani, "Effective and efficient hybrid android malware classification using pseudo-label stacked auto-encoder," *Journal of Network and Systems Management*, vol. 30, 2022, doi: 10.1007/s10922-021-09634-4.

BIOGRAPHIES OF AUTHORS

Malik AL-Essa has a Ph.D. in Computer Science. Currently works as an Assistant Professor in the Department of Computer Science, University of Jordan, Jordan. His research interests include cybersecurity, IDS, and XAI. He can be contacted at email: m.alessa@ju.edu.jo.

Mohammad Qatawneh is a faculty member in the Department of Computer Science at the University of Jordan. Currently, he is on sabbatical leave, serving at Al-Ahliyya Amman University in the Department of College of IT, Networks, and Cybersecurity. He earned his Ph.D. in Computer Science from Kiev University in 1996 and his M.Sc. in Computer Engineering from the University of Donetsk, USSR, in 1988. His research focuses on blockchain technology, cybersecurity, digital forensics, and the IoT. He can be contacted at email: M.qatawneh@ammanu.edu.jo.

Nidal Turab Ph.D. in Computer Science Professor at the Department of Networks and Cyber Security, Al-Ahliyya Amman University, Jordan. His research interests include WLAN security, computer networks security and cloud computing security, eLearning, and internet of things. He can be contacted at email: N.turab@ammanu.edu.jo.

Yazeed Alsarhan Ph.D. in Internet of Things, Assistant Professor at the Department of Computer Science, Al-Ahliyya Amman University, Jordan. Research interests focused on wireless sensor networks, internet of things, and wireless communication. He can be contacted at email: y.alsarhan@ammanu.edu.jo.