A Hybrid Deep Learning Framework for Real-Time Network Intrusion Detection in IoT Environments

OBAIDA M. AL-HAZAIMEH, ASHRAF A. ABU-EIN
Department of Computer Networks and Cybersecurity,
Faculty of Information Technology,
Jadara University,
Irbid, 21110,
JORDAN

Abstract: - In this study, a hybrid deep learning technique is proposed as the foundation for a real-time network intrusion detection system (i.e., NIDS) for Internet of Things (i.e., IoT) networks. (i.e., NIDS) stands for network intrusion detection system. Learning spatial information is accomplished using convolutional neural networks (i.e., CNNs), while modeling temporal information in network traffic patterns with long short-term memory (i.e., LSTMs) is also used. It is possible to create the architecture in such a way that it is both lightweight and efficient, taking into consideration the limitations of Internet of Things devices, such as the restricted processing power and communication protocols that they must support. Using the NSL-KDD benchmark dataset, the proposed model achieves a detection accuracy of 98.7%, while using the UNSW-NB15 benchmark dataset, it achieves 97.3%, and using the CIC-IDS-2017 benchmark dataset, it achieves 98.1%. Furthermore, the inference time for each packet is less than 2.5 milliseconds. It has been demonstrated through the findings that the CNN-LSTM hybrid model possesses superior detection performance in comparison to conventional machine learning classifiers and single-architecture deep learning models when it comes to identifying both known and novel attacks for various Internet of Things network topologies.

Key-Words: - Cybersecurity, deep learning, intrusion detection system, Internet of Things, network security, NSL-KDD dataset, UNSW-NB15 dataset.

Received: March 3, 2026. Revised: May 5, 2026. Accepted: June 7, 2026. Published: August 7, 2026.

1 Introduction

Myriad IoT gadgets have transformed computer and communications networks. By 2026, 75 billion Internet of Things devices will be required for applications such as industrial automation, smart homes, critical infrastructure, and healthcare. Existing security solutions cannot address IoT devices' heterogeneous and resource-limited characteristics, which make them a larger cyberattack surface, [1], [2], [3], [4]. Organizations are warned of cyberattacks using NIDS, which analyzes network traffic. In the IoT, neither signature-based nor anomaly-based NIDS has worked. Anomaly-based systems have high false alarm rates and are difficult to adapt to traffic changes in heterogeneous IoT deployments. Signature-based systems are ineffective against zero-day attacks and variants not yet seen in their signatures. Deep learning research can automatically discover complicated patterns and hierarchical structures from raw network information, addressing these issues. From packet payloads, flow statistics, and network traffic sequences, LSTM and CNNs may extract geographical and temporal features, [5], [6], [7], [8], [9], [10], [11], [12]. We present a CNN-LSTM hybrid network intrusion detection system for real-time IoT networks in this research. A novel hybrid model that combines CNN for spatial feature extraction and LSTM for temporal modeling of network traffic to classify traffic; an efficient preprocessing strategy that reduces computation resource requirements and facilitates deployment on IoT edge devices; and extensive evaluations on three well-known benchmark datasets to demonstrate superior intrusion detection.

1.1 Conventional Intrusion Detection Methods

Pattern matching was the primary method that was utilized in the initial attempts at intrusion detection. This method involved comparing sets of signatures

that corresponded to known assaults with network traffic at the time. However, these technologies are not able to detect new assaults, despite the fact that they are quite good in identifying known incidents. Anomaly detection systems that are based on statistics, such as principal component analysis (i.e., PCA) and Gaussian mixture models (i.e., GMM), are more advanced than signature matching. These systems learn the normal operation of the network and spot suspicious behavior. The methodologies stated above, on the other hand, have limited performance when it comes to the high dimensionality of Internet of Things network traffic and the wide diversity of lawful communication patterns that Internet of Things devices can use, [11], [13], [14], [15].

1.2 Machine Learning-Based Detection

Random forests, support vector machines (i.e., SVM), and gradient boosting are among the popular machine learning models used for network intrusion detection. They rely on feature engineering and domain knowledge to select features from network traffic data [4]. Recent research has shown that ensemble methods can attain detection rates of over 95% on benchmark datasets, but their effectiveness drops dramatically in the face of advanced evasion strategies and unseen attacks that do not follow the same distribution as the training data, [16].

1.3 Deep Learning for Network Security

For network intrusion detection, deep learning models are extensively employed since they automatically learn features. For unsupervised anomaly detection, autoencoders are utilized, whereas DNNs and CNNs classify traffic. LSTM and its extensions have been used to capture network traffic temporal correlations. Current model evaluation methods employ a single dataset and fail to account for real-time IoT implementation computing problems, [17], [18]. Hybrid deep learning architectures are being investigated. CNN-RNN hybrid models for network traffic classification have been studied in enterprise networks with enough computational resources. For IoT edge devices, this study addresses the construction of efficient hybrid models with high detection accuracy, low latency, and minimal resource needs, [19], [20].

2 Proposed Methodology

2.1 System Architecture

The hybrid architecture is comprised of three main components: data preprocessing, feature extraction and classification, and real-time alert generation. The data pre-processing module processes raw packet capture data, aggregates network flows, and normalizes the data. The classification engine uses the hybrid CNN-LSTM model, and the alert system produces threat alerts categorised according to the model's predictions.

2.2 Data Preprocessing

Using the network key, which is a five-tuple consisting of the source IP, the destination IP, the source port, the destination port, and the protocol, the first step is to divide the network traffic into flow pairs and two-way streams. Each flow is subjected to the computation of 78 statistical features, which include distributions of packet sizes, inter-packet arrival durations, the existence of flags, and other information that is specific to the protocol. Normalization of the features is implemented using min-max normalization to the interval [0, 1], and the feature vectors are reshaped into matrices for feeding into the CNN. For temporal modeling, sequences of flows are created using W = 10 flows and a window of size.

2.3 CNN-LSTM Hybrid Architecture

The hybrid CNN-LSTM architecture processes features in two sequential stages: spatial extraction via CNN layers and temporal modelling via LSTM layers. CNNs detect local statistical patterns within a single flow record, while LSTMs capture dependencies across consecutive flows that may indicate multi-step attack campaigns. The architecture of the proposed CNN-LSTM model is illustrated in Figure 1, [18], [19].

Since, in theory, neither the CNN nor the LSTM can model inter-window temporal dependencies on its own, nor can the LSTM capture local spatial correlations when trained on raw data, the CNN-LSTM combination should be preferred. Collectively, they reduce computational cost and enhance the gradient flow by lowering the effective sequence length the LSTM must process.

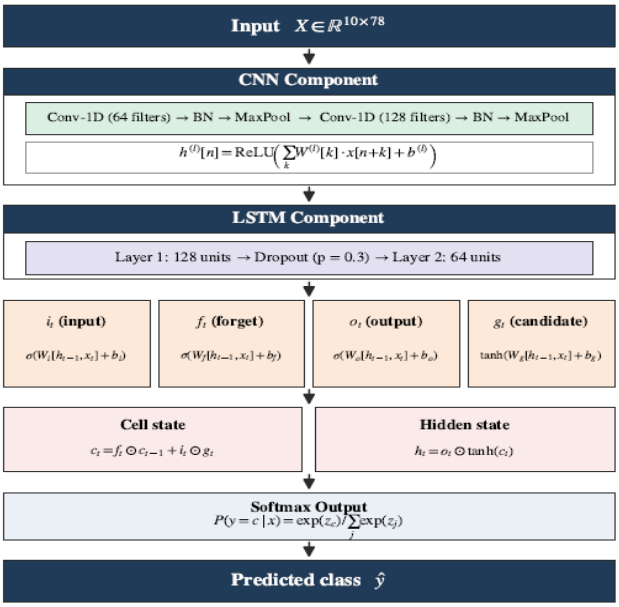

Fig. 1: Architecture of the proposed CNN–LSTM model
Source: created by the authors

2.4 Model Optimization

The model is trained using the Adam optimizer. Figure 2 shows the updating rules for the gradient *g_t*.

A. Adam Optimiser

1st moment	2nd moment	Parameter update
$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$	$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$	$\theta_t = \theta_{t-1} - \alpha \hat{m}_t / (\sqrt{\hat{v}_t} + \varepsilon)$
$\alpha = 0.001$	$\beta_1 = 0.9$	$\beta_2 = 0.999$
$\varepsilon = 10^{-4}$	LR decay = ×0.95 every 5 epochs	

B. Loss Function

Categorical Cross-Entropy (with class weighting)

$$L = -\frac{1}{N}\sum_i \sum_c w_{i,c} \cdot \log P(y = c \mid x_i)$$

Class weight: $w_c = N/(C \cdot N_c)$

C. Data Handling

Feature normalisation	Class imbalance
Min–max scaling → [0, 1]	SMOTE oversampling (minority classes)

D. Computational Complexity

Forward-pass per sample: $O(W \cdot F \cdot K \cdot N_f + W \cdot N_h^2)$

Edge deployment: 8-bit quantisation → tractable inference

Fig. 2: Training procedure and optimisation
Source: created by the authors

3 Experimental Evaluation

3.1 Datasets

We use three different datasets as benchmarks. 125,973 training cases and 22,544 test instances are included in the NSL-KDD dataset. Additionally, there are 41 features and 5 traffic classifications. In the UNSW-NB15 dataset, there are 49 characteristics and 10 different types of attacks, and there are 175,341 training records and 82,332 test records. More than 2.8 million flow records make up the CIC-IDS2017. These flow records include 78 features for benign traffic and 14 different assaults. Eighty percent of the training data is used for testing, and twenty percent is used for training. Ten percent of the training data is kept for validation purposes.

3.2 Evaluation Metrics

The evaluation is carried out on the basis of different metrics from the confusion matrix as shown in Figure 3.

A. Confusion-Matrix Metrics

Accuracy	Precision	Recall (DR)
(TP+TN) / (TP+TN+FP+FN)	TP / (TP+FP)	TP / (TP+FN)
F1-Score	**False Alarm Rate (FAR)**	**Averaging**
2·TP / (2·TP+FP+FN)	FP / (FP+TN)	Macro-average across all C classes

B. Experimental Protocol

Runs	Reporting	Inference
5× independent (different seeds)	Mean values	Per-flow latency measured
GPU	**CPU**	**Mode**
NVIDIA RTX 3080	Intel Core i9-12900K	Real-time assessment

Fig. 3: Evaluation metrics and experimental setup
Source: created by the authors

Each experiment is repeated five times with a different set of random seeds on a workstation with an NVIDIA RTX 3080 GPU and an Intel Core i9-12900 K processor, and the mean values are displayed.

3.3 Results and Analysis

Across the NSL-KDD, UNSW-NB15, and CIC-IDS2017 datasets, the CNN-LSTM approach that was proposed has the maximum detection accuracy, as demonstrated in Figure 4 and Figure 5.

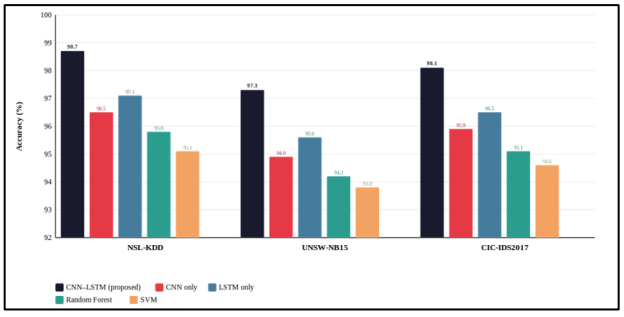

Fig. 4: Accuracy– Performance comparison
Source: created by the authors

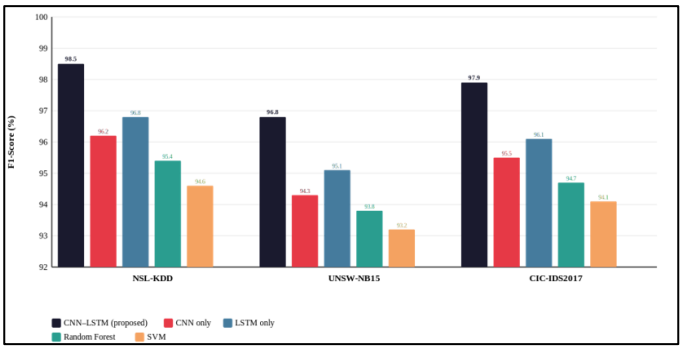

Fig. 5: F1 Score– Performance comparison
Source: created by the authors

For the best performance of the proposed model and baselines on the three datasets in Table 1, the accuracy, F1-score, recall, and precision are presented. As shown in the image, the CNN-LSTM hybrid model outperforms all other approaches in all metrics and datasets. UNSW-NB15 has the most attack categories and is hardest to classify; hence, it has the greatest improvement margin.

Table 1. Performance Comparison on Benchmark Datasets (%)

Method / Dataset	Acc.	Prec.	Rec.	F1
Random Forest (NSL-KDD)	94.5	93.8	94.1	93.9
SVM (NSL-KDD)	95.1	94.6	94.9	94.7
CNN (NSL-KDD)	96.3	95.9	96.1	96.0
LSTM (NSL-KDD)	97.1	96.8	97.0	96.9
CNN-LSTM (NSL-KDD)	98.7	98.4	98.6	98.5
Random Forest (UNSW-NB15)	93.2	92.5	92.8	92.6
SVM (UNSW-NB15)	94.0	93.4	93.7	93.5
CNN (UNSW-NB15)	95.1	94.6	94.9	94.7
LSTM (UNSW-NB15)	95.8	95.3	95.6	95.4
CNN-LSTM (UNSW-NB15)	97.3	96.9	96.7	96.8
Random Forest (CIC-IDS2017)	95.0	94.3	94.7	94.5
SVM (CIC-IDS2017)	95.8	95.2	95.5	95.3
CNN (CIC-IDS2017)	96.1	95.7	95.9	95.8
LSTM (CIC-IDS2017)	96.9	96.5	96.7	96.6
CNN-LSTM (CIC-IDS2017)	98.1	97.8	98.0	97.9

Source: created by the authors

Ablation tests on NSL-KDD showed that removing the CNN stage decreases accuracy by 1.6 points (97.1%), replacing the LSTM layer with global average pooling decreases accuracy by 2.4 points (96.3%) and increases training instability, and removing batch normalization decreases accuracy by 0.9 points and increases training instability. These data support temporal modeling using LSTM with the greatest gain per component and independent component contributions.

With a 96.2% accuracy rate on previously discovered attack patterns in the CIC-IDS2017 dataset, the hybrid model performs exceptionally well when it comes to the identification of zero-day intrusions. The false alarm rate is kept under 1.3% across all datasets, which is a substantial improvement compared to traditional anomaly detection systems, which typically have a false alarm rate greater than 5%. The model's average inference time is 2.3 milliseconds per flow, which is significantly less than the real-time processing requirement for deployment in an IoT gateway.

3.4 Per-Category Attack Detection

Examination of individual attack category detection performance shows the CNN-LSTM architecture attains the best accuracy for volumetric attacks like denial-of-service (i.e., DoS) and distributed denial-of-service (i.e., DDoS) attacks with over 99.2% detection accuracy. The model achieves 97.8% accuracy for reconnaissance attacks, such as port scanning and network mapping. The model's accuracy for exploitation attacks (i.e., modification of payloads) is 95.4%, which is the lowest among the attack categories, but still outperforms the baseline methods by 3.1-4.6 percentage points. Analysis of the confusion matrix shows the most common source of misclassification is between similar attack subcategories, such as generic Denial of Service (i.e., DoS) and application layer flooding attacks. This is due to the similarity in statistical patterns of the flow-level features. But from a security standpoint, these misclassifications are mostly irrelevant because the model is able to accurately detect malicious traffic even if the subcategory of the attack is misclassified.

4 Discussion

4.1 Feature Extraction Analysis

Intrusion detection using varying feature extraction strengths is shown via the CNN-LSTM hybrid model. Local statistical features like atypical packet size statistics and rare packet flag combinations are

extracted by the CNN, while the LSTM extracts temporal attack patterns like suspicious scanning patterns and ramp-up sequences in DDoS attacks from multiple consecutive flows. Flow duration, total forward packets, and packet length variance have the largest activation weights in CNN layers, indicating the relevance of individual variables. Since these traits indicate attacks, expert knowledge supports this. With the highest activation to the series of time intervals between packet arrivals and flow state transitions, the LSTM layers capture temporal features of attacks that single-flow analysis misses.

4.2 Computational Efficiency

High-efficiency data preprocessing supports low-end IoT edge gateway deployment. Low-precision model quantization (i.e., 8-bit integer) and structured pruning reduce the model from 12.4 MB to 4.8 MB (61.3% smaller with less than 0.5% accuracy loss). The smaller model fits into the memory footprint of off-the-shelf IoT gateways like the Raspberry Pi 4 and NVIDIA Jetson Nano, allowing deployment on edge devices with limited memory and processing capacity. The model predicts a Jetson Nano throughput of 435 flows per second, suitable for small to medium-sized IoT networks with up to 25,000 flows per minute. For enterprise IoT applications, the model can be distributed on numerous edge devices or a central server with GPU processing and more than 12,000 flows per second per NVIDIA RTX 3080 GPU.

4.3 Comparison with State-of-the-Art

The proposed model compares favorably against recent literature. On NSL-KDD, [9] reported 97.5% accuracy with a five-layer DNN 1.2 points below our 98.7%. [15] achieved 97.1% with a standalone LSTM but without spatial feature extraction. [8] reported 96.8% on encrypted traffic with a CNN-only model, while our hybrid delivers 98.1% on the more challenging CIC-IDS2017. Ensemble methods (random forest, gradient boosting) range from 93–95% across the three datasets, confirming a consistent 2–4% gap versus deep learning approaches. The key differentiators of the proposed architecture are: (i) end-to-end spatial and temporal feature extraction without separate handcrafted preprocessing; (ii) evaluation across all three major benchmarks for direct cross-dataset comparison; (iii) edge-deployment optimization yielding a 4.8 MB quantized model (61.3% reduction, $\leq$0.5% accuracy loss); and (iv) real-time throughput of 435 flows/s on the NVIDIA Jetson Nano. Regarding novelty relative to the author's prior publications: reference [20] addresses ML-based penetration testing using classical classifiers for vulnerability assessment a different task from real-time intrusion detection. References [13], [14], [18], and [19] address network security and encryption topics, and none proposes or evaluates a CNN-LSTM hybrid IDS architecture. The present work therefore constitutes a distinct original contribution.

4.4 Limitations and Future Work

Supervised training requires labeled data, which is a downside. Real-world IoT applications make it difficult and expensive to collect large labeled samples of all attacks. Research will examine semi-supervised and federated learning methods to reduce labeled data and ensure privacy in collaborative IoT contexts. Attention methods and transformer-based architectures can let the model preferentially scan traffic sequences for the most distinct temporal patterns, improving feature extraction. Since IoT protocols are moving to encrypted channels, this architecture will be extended to analyze encrypted IoT traffic while ensuring end-to-end security. Finally, adversarial training methodologies boost model resilience against evasion assaults, when advanced adversaries produce traffic to sneak past a deep learning-based detector.

5 Conclusion

CNN-LSTM deep learning (i.e., DL) systems for real-time intrusion detection in IoT networks are proposed in this work. The model takes advantage of convolutional layers to extract spatial characteristics from network traffic and LSTM layers to simulate temporal patterns to characterize network traffic behavior. Extended testing on three typical benchmark datasets demonstrates that the proposed model achieves state-of-the-art accuracy of over 97% with an average inference latency of 2.3 ms per flow, validating its use in real-time IoT. Ablation investigations demonstrate the architectural components' individual contributions. All metrics and datasets show that the hybrid model outperforms deep learning models and machine learning classifiers. The model is precise (96.2%) on zero-day attack scenarios, demonstrating its practicality in dynamic IoT environments with new threats. Future research includes federated learning for privacy-preserving distributed model training across multiple IoT

deployments, real-time analysis of encrypted IoT traffic for end-to-end security, and adversarial training to strengthen the model against evasion attacks.

Acknowledgement:
We like to convey our appreciation to Jadara University for their assistance in the completion of this project. Their resources and support facilitated this research.

References:

[1] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016, https://doi.org/10.1109/COMST.2015.2494502.

[2] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," *in Proc. Military Communications and Information Systems Conf. (MilCIS)*, Canberra, Australia, 2015, pp. 1–6, https://doi.org/10.1109/MilCIS.2015.7348942.

[3] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. *4th Int. Conf. Information Systems Security and Privacy*, Funchal, Madeira, Portugal, 2018, pp. 108–116, https://doi.org/10.5220/0006639801080116.

[4] O. M. Al-Hazaimeh, A. A. Abu-Ein, And T. M. Younes, "Cybersecurity Foundations: Understanding Computer, Network, And Internet Security," *Wseas Engineering World*, Vol. 7, Pp. 112-119, 2025, https://doi.org/10.37394/232025.2025.7.12.

[5] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, pp. 1–19, 2020, https://doi.org/10.1016/j.jisa.2019.102419.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015, https://doi.org/10.1038/nature14539.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, https://doi.org/10.1162/neco.1997.9.8.1735.

[8] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in Proc. *IEEE Int. Conf. Intelligence and Security Informatics*, Beijing, China, 2017, pp. 43–48, https://doi.org/10.1109/ISI.2017.8004872.

[9] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, https://doi.org/10.1109/ACCESS.2019.2895334

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

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002, https://doi.org/10.1613/jair.953.

[12] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," Neurocomputing, vol. 415, pp. 295–316, 2020, https://doi.org/10.1016/j.neucom.2020.07.061.

[13] E. Hodo, X. Bellekens, A. Hamilton, P. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of IoT networks using artificial neural network intrusion detection system," in Proc. *Int. Symp. Networks, Computers and Communications*, Yasmine Hammamet, Tunisia, 2016, pp. 1–6, https://doi.org/10.1109/ISNCC.2016.7746067.

[14] O. M. Al-Hazaimeh, A. Ma'moun, T. Abuain, and A. A. Abu-Ein, "End-to-end cybersecurity encryption-video algorithm," *WSEAS Transactions on Computer Research*, vol. 13, pp. 116-123, 2025, https://doi.org/10.37394/232018.2025.13.12.

[15] J. Kim, J. Kim, H. Thu, and H. Kim, "Long short-term memory recurrent neural network classifier for intrusion detection," in Proc. *Int. Conf. Platform Technology and Service*, Jeju, South Korea, 2016, pp. 1–5.

[16] S. Potluri and C. Diedrich, "Accelerated deep neural networks for enhanced intrusion detection system," in Proc. *IEEE Int. Conf. Emerging Technologies and Factory Automation*, Limassol, Cyprus, 2016, pp. 1–8, https://doi.org/10.1109/ETFA.2016.7733515.

[17] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in Proc. *IEEE National Aerospace and Electronics Conf.*, Dayton, OH, USA, 2015, pp. 339–344, https://doi.org/10.1109/NAECON.2015.7443094.

[18] Al-Hazaimeh, Obaida M., and A. Ma'moun. "Vehicle to vehicle and vehicle to ground communication-speech encryption algorithm." In *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, Tenerife, Canary Islands, Spain, pp. 1-4. IEEE, 2023, https://doi.org/10.1109/ICECCME57830.2023.10252814.

[19] Al-Qasrawi, Isra Sitan, and Obaida Mohammed Al-Hazaimeh. "A pair-wise key establishment scheme for AD HOC networks." *International Journal of Computer Networks & Communications* 5, no. 2 (2013): 125-136, https://doi.org/10.5121/ijcnc.2013.5210.

[20] Bawaneh, Mohammed J., et al., "Enhanced Iot Cybersecurity Through Machine Learning-Based Penetration Testing." *Applied Computer Science* 21, No. 2 (2025): 96-110, https://doi.org/10.35784/acs_7397.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The current study was developed by the authors, who were equally involved in every step of the process, from the conceptualization of the problem to the final results and resolution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.