Cybersecurity Enhancement in IoT Wireless Sensor Networks using Machine Learning

ELIE EL AHMAR[1], ALI RACHINI[1], HANI ATTAR[2]

[1]Department of Computer Science and Information Technology,
Holy Spirit University of Kaslik (USEK),
P.O. Box 446 Jounieh, Mount Lebanon - LEBANON

[2]Faculty of Engineering,
Zarqa University, Zarqa, JORDAN

Abstract: In the context of the Internet of Things, this paper presents approaches in order to enhancing the security in Wireless Sensor Networks. It addresses the challenges arising from the lack of standardization in IoT. On the other hand, this paper proposes a machine learning and AI algorithms to detect the intrusion detection. WSNs, which are crucial for data collection across various applications, face several security threats like eavesdropping and Denial of Service (DoS) attacks. The proposed approach in this paper present accuracy rates of 0.98 for Random Forest, 0.90 for SVM, and 0.95 for KNN. It demonstrates the effectiveness of machine learning in identifying various types of attacks. This method not only improves authentication efficiency but also significantly enhances the detection and classification of diverse security threats, paving the way for substantial advancements in cybersecurity within IoT environments.

Key-Words: Wireless Sensor Networks, IoT, Security Enhancement, SVM, KNN, Random Forest, Attack Detection, DDOS attack.

Received: December 17, 2023. Revised: August 16, 2024. Accepted: September 21, 2024. Published: October 14, 2024.

1 Introduction

Wireless Sensor Networks (WSNs) are essential for gathering data from various environments. They use small autonomous devices known as sensors. Some critical application that use the WSN are: Environmental monitoring, surveillance, and industrial automation. However, the characteristics of WSNs, like limited resources, low computational power, and vulnerability to attacks, need robust security measures, [1].

There are three fundamental components to security in WSNs: confidentiality, integrity, and availability. Confidentiality is to protect sensitive information from unauthorized access. Integrity ensures that the data remains unchanged during transmission or storage stages. Availability guarantees uninterrupted network access for all the authorized users. Achieving these security objectives requires the implementation of encryption, authentication, access control, and intrusion detection mechanisms, [2].

The main challenges in securing WSNs depend on resources, which include energy, processing power and memory. Those challenges require lightweight, energy-efficient security solutions that still provide strong protection. Moreover, WSNs are vulnerable to various types of attacks, like eavesdropping, node compromise and the injection of malicious nodes, which can disturb the network operations and compromise data integrity. Other threats include node malfunction and failure, which can affect network performance, as well as message corruption and traffic analysis, which can let drop network topology and routing information. Specific attacks like routing loops, selective forwarding, sinkhole attacks, and Sybil attacks exploit network vulnerabilities to take in data transmission.

AI has become a powerful tool for detecting attacks in WSNs. Rule-based attack signature-based detection methods frequently fall behind the ever-changing nature of threats, as they require manual updates and maintenance. However, machine learning algorithms offers several advantages, they can learn from various amounts of data, enabling the identification of complex patterns and anomalies associated with various attacks. This adaptability allows AI models to detect unknown attacks. On the other hand, those algorithms more robust and capable to deal with new and emerging threats. Furthermore, AI techniques can provide real-time detection and response, facilitating proactive measures to mitigate potential damages. By constantly checking the network and analyzing incoming data, AI models can identify suspicious activities on the network. The usage of AI in WSN attack detection offers more robust, adaptable and efficient approach compared

to traditional methods. It enhances the ability of WSNs to detect known and unknown attacks, thereby ensuring the security, integrity, and availability of the collected data.

The main purpose of this paper is to propose ways to improve WSN security. The primary focus is lying on the authentication challenges in IoT, like the lack of standardization, the limited processing power of devices, and the management of authentication for a large number of interconnected devices, [3]. After examining and implementing standardized authentication protocols, lightweight authentication methods, and leveraging AI and machine learning techniques, the goal is to establish robust security measures that ensure security of IoT networks and devices. In this paper, we demonstrated that machine learning algorithms like Random Forest, SVM, and KNN can effectively classify multiple types of attacks that can be detected in IoT frameworks. We also emphasized the importance of advanced algorithms in enhancing IoT network security, which is critical for applications such as environmental monitoring and industrial automation. In conclusion, we highlighted the significance of standard authentication protocols for managing a large number of interconnected IoT devices.

The structure of this paper is organized as follows: Section 2 reviews relevant literature, providing valuable insights into existing research. Section 3 outlines the methodology used and details the datasets utilized for the study. Section 4 presents the results of our research efforts, offering a comprehensive analysis of the findings. The study concludes with Section 5, which summarizes the key findings, discusses their implications, and suggests directions for future research.

2 Related Work

Intrusion detection in IoT environments relies on supervised machine learning (ML) algorithms, [4]. Deep learning (DL) techniques such as Autoencoders (AEs), [5], Feedforward Neural Networks (FNNs), [6], Deep Belief Networks (DBNs), [7], and dense random neural networks, [8], have been widely adopted to address intrusion detection challenges. The authors in [9], proposed a bidirectional LSTM within a deep blockchain framework for secure data exchange in multicloud IoT services. The authors in [10], introduced a memory module in AE models to store and locate space feature representations. They enhancing the detection of unknown attacks. Simialrly, the authors in [11], combined federated learning and fog/edge computing for distributed denial-of-service (DDoS) traffic detection on IoT devices. Additionally, a federated learning scheme on a decentralized platform was presented in [12]. The study in [13], aims to empirically assess the efficacy of various Machine Learning algorithms in enhancing the performance of Intrusion Detection Systems (IDS).

The research work in [14], provides a survey about IoT and Machine Learning on multimodal information-based learning for safety and security. The work in [15], proposes using Deep Learning to aid in the development of Effective Multimedia Data Models (DLA-EMDM), where in [16], offers a comprehensive analysis of security threats against WSN and IoT, along with the strategies for preventing, detecting and mitigating those threats.

DL models have been integrated into intrusion detection systems (IDSs) to enhance their performance. For instance, a framework combining gated recurrent units (GRUs), a multihead self-attention mechanism (MHSA), and feedforward layers was proposed, [17], enabling effective extraction and parallel execution of traffic representations. Hyperparameter optimization using evolutionary techniques has shown promise in improving IDS performance, such as particle swarm optimization for CNNs, [18].

However, addressing class imbalance in IoT data remains a challenge. The authors in [19], employed oversampling techniques to mitigate class imbalance and applied a two-layer model combining LSTM and random forest classifiers. The authors in [20], introduced a cost-sensitive learning strategy in sparse autoencoder models and enhanced cost adjustment using an evolutionary algorithm. Few-shot learning with variational feature representation was utilized to tackle the out-of-distribution problem in imbalanced data, [21].

D-Sign, [22], employs DL for intrusion detection and signature generation of unknown web attacks, combining misuse detection and anomaly detection engines for comprehensive threat analysis. These advancements underscore the importance of integrating DL models and addressing class imbalance to enhance the effectiveness of IDSs in IoT environments.

The authors in [23], develop the WSN-DS dataset to enhance intrusion detection in wireless sensor networks (WSNs), focusing on classifying four types of DoS attacks using data collected via NS-2 simulations of the LEACH protocol, and analyzed with an Artificial Neural Network (ANN) trained

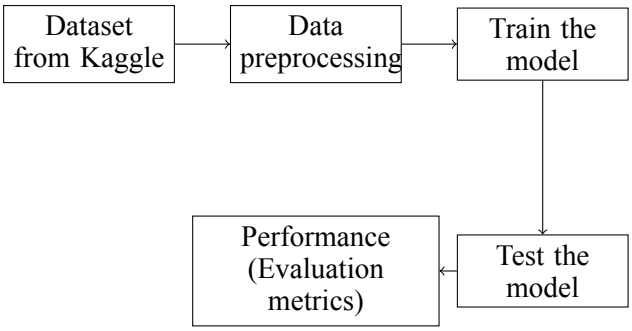

Figure 1: Proposed system model

using WEKA.

3 Methodology

In this section, we outline the dataset used, introduce the system model adopted, and discuss the data preprocessing steps undertaken in a systematic manner.

3.1 System Model

The block diagram shown in Figure 1 outlines the step-by-step process of constructing and evaluating a machine learning model. We use a dataset obtained from Kaggle, this dataset undergoes a series of preprocessing steps aimed at cleaning and preparing the data for training. Once the data is preprocessed, the model is trained on this data. During this phase, the system learns the relationships within the dataset. After this phase, the model is tested in order to evaluate its performance and accuracy of predicting of unseen data. After that, the model's accuracy is checked to see how well the trained model does at the given task.

3.2 Dataset

The WSN-DS dataset is available on Kaggle, [24]. It is used for intrusion detection systems (IDSs) in WSNs. It focuses on Denial of Service (DoS) attacks, which are a significant security threat to WSNs. This dataset is used to identify and classify four types of DoS attacks like: blackhole, grayhole, flooding, and scheduling. The data is derived from the LEACH protocol and NS-2. This enhances the effectiveness of IDS and advances WSN security research. The dataset contains 374,661 instances, offering a potential for analysis and robust algorithm development. Also, it includes 19 attributes for WSN security. These features encompass a three-digit Node ID, simulation Time, and a binary flag indicating if a node is a Cluster Head (CH). It also includes identifiers for current and maximum CH distance. Key metrics such as RSSI, node energy levels, broadcast messages, and join requests are tracked. The dataset monitors node ranking in TDMA schedules and the quantity of data packets exchanged between nodes and CHs, and the Base Station. It also measures the CH-to-Base Station distance. Attack classification is noted for each node, distinguishing normal operations from the four DoS attack types: blackhole, grayhole, flooding, and scheduling.

3.3 Data Pre-Processing

In the data pre-processing phase, we checked the dataset for null values and duplicates. While no null values appeared, we found 8,873 duplicated entries. To ensure data integrity, we removed these duplicates. This step was crucial to avoid biases and inaccuracies. By eliminating duplicates, each observation remains unique, maintaining high data quality for further analyses. On the other hand, in the label encoding step, attack types were converted to numerical values. This process mapped textual attack types to numbers. It helps computational models process and analyze data efficiently. The encoding table is essential for interpreting attack types within the dataset. Finally, we balanced the data. This involved equalizing the row count for each attack type. It ensures unbiased model training and a thorough analysis of attack characteristics. A balanced dataset prevents any attack type from dominating, promoting fairness and accuracy in classification evaluations.

3.4 Evaluation Metrics

In this section we provide the evaluation metrics, [25].

3.4.1 Confusion Matrix

In AI and machine learning, a confusion matrix is used to assess classification algorithms. The matrix divides the predictions into four categories: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). These categories are essential for calculating performance metrics like accuracy, precision, recall, and the F1 score. These metrics help us to evaluate the model's predictive abilities.

3.4.2 Precision

A metric that measures the extent of accurately anticipated positive occurrences is characterized as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

3.4.3 Recall

The classifier's recall, also known as the true positive rate, measures its ability to correctly identify positive instances and is calculated as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

3.4.4 F1-Score

The F1-Score, representing the mean of precision and recall, provides a fair evaluation metric and can be expressed using the formula:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

3.4.5 Classification Report

The Classification Report gives a summary of model for each class. It presents measurements like accuracy, review, and F1-Score.

4 System Implementation

In this section, we present and analyze the algorithms devised for our study, which are fundamental for detecting attacks in IoT networks. The algorithms outlined here aim to enhance the precision and effectiveness of attack detection, enabling a deeper understanding and interpretation of security threats in IoT environments.

4.1 Random Forest

Data scientists use various machine learning algorithms to uncover patterns in large datasets, providing critical insights for strategic decisions. Random Forest is a favored algorithm due to its ability to handle both classification and regression tasks. Introduced by Leo Breiman and Adele Cutler in the early 2000s, it enhances traditional decision trees by combining multiple random decision trees to improve prediction performance, manage overfitting, and handle complex datasets effectively, [26]. Decision trees, though intuitive, often overfit, limiting their generalizability. Random Forests counter this by merging multiple decision trees and using specific training strategies to boost accuracy and reduce overfitting. They rely on key hyperparameters such as tree size, the number of trees, and the number of sampled features, blending bagging with decision trees and resampling to create diverse predictive models. Each decision tree is built by selecting optimal splits based on criteria like the Gini index for classification or variance reduction for regression, considering only a subset of attributes at each node. The final prediction is an aggregate of all trees' outputs, averaged for regression tasks and majority-voted for classification. This ensemble method enhances Random Forests' predictive power and versatility, making them essential for data scientists in deriving insights and supporting organizational decisions. Random Forest aggregates the forecasts of M individual decision trees to derive the ultimate prediction, [27]:

$$\hat{Y}_{\text{RF}} = \frac{1}{M} \sum_{j=1}^{M} \hat{Y}_j \tag{4}$$

Here:

- $\hat{Y}_{\text{RF}}$ denotes the Random Forest forecast.
- $\hat{Y}_j$ signifies the prediction generated by the jth decision tree.

4.2 The Support Vector Machine (SVM)

The Support Vector Machine (SVM) operates by mapping data to a high-dimensional attribute space, facilitating classification even when linear separation is unattainable, [28]. It identifies a separator between categories, transforming the data to align with a hyperplane for classification, thereby enabling the utilization of new data features to predict group assignments. The primary objective is to provide the algorithm with flexibility in selecting the separation line, accommodating a margin of error known as the "soft margin". We will now elucidate the Soft Margin Classifiers algorithm, positioned between the Support Vector Machine and the Maximal Margin Classifier.

Soft Margin Classifiers revolve around the concept of margin, which denotes the distance between a separating line and the nearest observation. To enhance adaptability, a threshold is introduced to specify the allowable number of observations within the margin. The goal remains to maximize the margin while allowing for observations within this threshold. By prioritizing margin maximization over the precise classification of points within the margin, the algorithm demonstrates robustness to outliers and extreme values, fostering a more generalized classification model. The decision function for SVM is expressed as:

$$f(x) = \text{sign}\left(\sum_{i=1}^{n} \beta_i y_i K(x, x_i) + b\right) \tag{5}$$

Where:

- $f(x)$ denotes the predicted class label.
- β_i represents the Lagrange multipliers.
- y_i signifies the class label of the training sample.
- $K(x, x_i)$ stands for the kernel function.
- b denotes the bias term.

4.3 K-Nearest Neighbors (KNN)

An algorithm is given a dataset with labeled output values in supervised learning. This dataset serves as the foundation for training and building a predictive model. This prepared algorithm can

accordingly be utilized on new, unlabeled data to predict their corresponding output values. Among the various supervised learning approaches, the K-Nearest Neighbors (KNN) algorithm stands out for its intuitive methodology, [29].

Initially, the KNN algorithm involves the selection of a value for K, representing the number of nearest neighbors to consider in the classification process. Following this, the distance from the unlabeled feature to each of the other data points is calculated. The K data points closest to the unlabeled point are then determined based on these calculated distances. The algorithm moves on to determining the distribution of categories among these neighboring points after identifying the K closest neighbors. By counting the number of points belonging to each category, it determines the predominant class within the selected group.

Once the class distribution among the K nearest neighbors is established, the algorithm assigns the new, unlabeled feature to the category that is most prevalent within this group. This step completes the classification process, and the model is now ready for use in making predictions on new data instances. The predicted class label using KNN is determined by the majority class among the K nearest neighbors:

$$\hat{C} = \text{majority}\left(\{c_i\}_{i \in \mathcal{N}_K(x)}\right) \tag{6}$$

Where:

- $\hat{C}$ denotes the predicted class label.
- c_i represents the class labels of the K nearest neighbors of data point x.

5 Results and Discussion

Table 1 shows the classification report for: Random Forest, SVM and KNN algorithms. It evaluates their performance across three classes (0, 1, and 2) using precision, recall, F1-score, and accuracy metrics.

A comparison of classification reports for Random Forest, SVM, and KNN algorithms is shown in the table above. Each algorithm's performance is evaluated across five classes, named from 0 to 4. For Random Forest, it achieves high precision, recall, and F1-score across all classes, demonstrating its effectiveness in classification. The accuracy for Random Forest is also very high, reaching 98%. Conversely, SVM shows slightly lower performance, especially in class 2, where precision and recall are relatively lower compared to other classes. SVM has an overall accuracy of 90%. KNN performs reasonably well, with precision, recall, and F1-score metrics above 0.9 for most classes. However, it shows a slightly lower accuracy of 95% compared to Random Forest. Overall, Random Forest outperforms all other models in every class, followed by KNN and SVM. The table provides significant insights into the strengths and weaknesses of each algorithm in classifying data from different categories.

Table 1. Comparison of Classification Reports

Algorithm	Class	Precision	Recall	F1-score	Accuracy
Random Forest	0	0.97	1.00	0.99	0.98
	1	1.00	1.00	1.00	
	2	0.99	0.98	0.98	
	3	0.94	0.99	0.97	
	4	1.00	0.94	0.97	
SVM	0	0.71	1.00	0.83	0.90
	1	1.00	1.00	1.00	
	2	0.96	0.62	0.75	
	3	0.94	0.97	0.96	
	4	0.99	0.93	0.96	
KNN	0	0.88	0.98	0.93	0.95
	1	1.00	1.00	1.00	
	2	0.95	0.87	0.91	
	3	0.94	0.97	0.96	
	4	0.98	0.92	0.95	

Table 2 presents a comparative analysis of the accuracy achieved by different algorithms. This table includes our models and some existing works in literature.

Table 2. Comparative Analysis between our models and existing work

Authors	Algorithm	Accuracy
Our Models	Random Forest	0.98
	SVM	0.90
	KNN	0.95
[30]	DNN	0.96

Our models, using Random Forest, SVM, and KNN algorithms, achieved accuracies of 0.98, 0.90, and 0.95, respectively. These results demonstrate the precision with which our proposed models classify IoT network attacks. The authors in [23], on the other hand, obtained an accuracy of 0.95 using a 10-Fold Cross Validation method on the same dataset. This shows that their approach was less precise than our Random Forest model. Furthermore, [30], used a DNN model in order to achieve an accuracy of 0.96. While their DNN approach yielded slightly higher accuracy compared to our SVM and KNN models. on the other hand, our Random Forest model still achieved the highest accuracy among all the methods analyzed in the table. Overall, our models outperform other approaches, with Random Forest become as the most accurate algorithm for IoT environment attack detection in this paper. This competitive

edge underscores the robustness and reliability of our approach compared to existing systems.

6 Conclusion

WSNs are networks of distributed sensors that monitor and record environmental conditions, transmitting the collected data to a central location. These networks are particularly vulnerable to various security attacks due to their distributed nature and limited resources. In this paper, we assess the performance of Random Forest, SVM, and KNN for IoT attack detection using the WSN network. The best features are chosen using their correlation in our paper. On other word, we place an emphasis on the significance of feature selection and data preprocessing. Our machine learning models are significantly more accurate as a result of this approach. Random Forest achieves the highest accuracy at 98%, outperforming SVM (90%) and KNN (95%). Compared to previous approaches, our models perform competitively, demonstrating the effectiveness of AI in protecting IoT networks from cyber threats. These findings highlight the importance of utilizing advanced algorithms to improve the security posture of IoT environments, contributing to the advancement of intrusion detection systems despite evolving cybersecurity challenges.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT to check spelling and grammar, and to extract certain data values from images. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References:

[1] M. J. Khan, S. U. Khan, and A. U. Khan, "A review of the applications of wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, pp. 1–17, 2012.

[2] W. Du and J. Deng, "Security in wireless sensor networks," *IEEE wireless communications*, vol. 12, no. 6, pp. 16–24, 2005.

[3] A. S. Rachini and R. Khatoun, "Distributed key management authentication algorithm in internet of things (iot)," in *2020 Sixth International Conference on Mobile And Secure Services (MobiSecServ)*, pp. 1–5, 2020.

[4] A. Mishra and N. Gupta, "Supervised machine learning algorithms based on classification for detection of distributed denial of service attacks in sdn-enabled cloud computing," in *Cyber Security, Privacy and Networking: Proceedings of ICSPN 2021*, pp. 165–174, Springer, 2022.

[5] K. Yang, Y. Shi, Z. Yu, Q. Yang, A. K. Sangaiah, and H. Zeng, "Stacked one-class broad learning system for intrusion detection in industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 251–260, 2022.

[6] M. Ge, N. F. Syed, X. Fu, Z. Baig, and A. Robles-Kelly, "Towards a deep learning-driven intrusion detection approach for internet of things," *Computer Networks*, vol. 186, p. 107784, 2021.

[7] S. Manimurugan, S. Al-Mutairi, M. M. Aborokbah, N. Chilamkurti, S. Ganesan, and R. Patan, "Effective attack detection in internet of medical things smart environment using a deep belief neural network," *IEEE Access*, vol. 8, pp. 77396–77404, 2020.

[8] S. Latif, Z. e. Huma, S. S. Jamal, F. Ahmed, J. Ahmad, A. Zahid, K. Dashtipour, M. U. Aftab, M. Ahmad, and Q. H. Abbasi, "Intrusion detection framework for the internet of things using a dense random neural network," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6435–6444, 2022.

[9] O. Alkadi, N. Moustafa, B. Turnbull, and K.-K. R. Choo, "A deep blockchain framework-enabled collaborative intrusion detection for protecting iot and cloud networks," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9463–9472, 2021.

[10] H. Lu, T. Wang, X. Xu, and T. Wang, "Cognitive memory-guided autoencoder for effective intrusion detection in internet of things," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3358–3366, 2022.

[11] J. Li, L. Lyu, X. Liu, X. Zhang, and X. Lyu, "Fleam: A federated learning empowered architecture to mitigate ddos in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 4059–4068, 2022.

[12] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of things intrusion detection: Centralized, on-device, or federated learning?," *IEEE Network*, vol. 34, no. 6, pp. 310–317, 2020.

[13] A. Rachini, C. Fares, M. A. Assaf, B. Jamal, and R. Khatoun, "Ai-powered network intrusion detection: A new frontier in cybersecurity," in *2023 24th International Arab Conference on Information Technology (ACIT)*, pp. 1–8, 2023.

[14] H. Attar, "Joint iot/ml platforms for smart societies and environments: a review on multimodal information-based learning for safety and security," *ACM Journal of Data and Information Quality*, vol. 15, no. 3, pp. 1–26, 2023.

[15] M. S. Mahdi Hussin, M. R. Al-Hameed, M. Al-Tahee, S. A. Zearah, H. A. Diame, and A. R. Al-Tameemi, "Deep learning with wireless sensor network platform for multimedia data modeling," in *2023 Annual International Conference on Emerging Research Areas: International Conference on Intelligent Systems (AICERA/ICIS)*, pp. 1–6, 2023.

[16] S. Lata, S. Mehfuz, and S. Urooj, "Secure and reliable wsn for internet of things: Challenges and enabling technologies," *IEEE Access*, vol. 9, pp. 161103–161128, 2021.

[17] M. Abdel-Basset, V. Chang, H. Hawash, R. K. Chakrabortty, and M. Ryan, "Deep-ifs: Intrusion detection approach for industrial internet of things traffic in fog environment," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7704–7715, 2021.

[18] X. Kan, Y. Fan, Z. Fang, L. Cao, N. N. Xiong, D. Yang, and X. Li, "A novel iot network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network," *Information Sciences*, vol. 568, pp. 147–162, 2021.

[19] N. Gupta, V. Jindal, and P. Bedi, "Lio-ids: Handling class imbalance using lstm and improved one-vs-one technique in intrusion detection system," *Computer Networks*, vol. 192, p. 108076, 2021.

[20] A. Telikani and A. H. Gandomi, "Cost-sensitive stacked auto-encoders for intrusion detection in the internet of things," *Internet of Things*, vol. 14, p. 100122, 2021.

[21] W. Liang, Y. Hu, X. Zhou, Y. Pan, and K. I.-K. Wang, "Variational few-shot learning for microservice-oriented intrusion detection in distributed industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5087–5095, 2022.

[22] S. Kaur and M. Singh, "Hybrid intrusion detection and signature generation using deep recurrent neural networks," *Neural Computing and Applications*, vol. 32, no. 12, pp. 7859–7877, 2020.

[23] I. Almomani, B. Al-Kasasbeh, and M. Al-Akhras, "Wsn-ds: A dataset for intrusion detection systems in wireless sensor networks," *Journal of Sensors*, vol. 2016, 2016.

[24] B. Kasasbeh, "WSN-DS: A dataset for intrusion detection systems in wireless sensor networks," 2021.

[25] D. Axman and R. Yacouby, "Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models," 2020.

[26] S. M. Dubey, B. Kanwer, G. Tiwari, and N. Sharma, "Classification for eeg signals using machine learning algorithm," in *International Conference on Artificial Intelligence of Things*, pp. 336–353, Springer, 2023.

[27] F. Hernandez Vivanco, R. Smith, E. Thrane, and P. D. Lasky, "A scalable random forest regressor for combining neutron-star equation of state measurements: a case study with gw170817 and gw190425," *Monthly Notices of the Royal Astronomical Society*, vol. 499, no. 4, pp. 5972–5977, 2020.

[28] A. Karami and S. T. A. Niaki, "An online support vector machine algorithm for dynamic social network monitoring," *Neural Networks*, vol. 171, pp. 497–511, 2024.

[29] M. Li, G. Huang, L. Wang, and W. Xie, "Comprehensive classification assessment of gnss observation data quality by fusing k-means and knn algorithms," *GPS Solutions*, vol. 28, no. 1, p. 21, 2024.

[30] G. Xu, A. J. P. Delima, I. K. D. Machica, J. C. T. Arroyo, Z. He, and W. Su, "Improvement of wireless sensor networks against service attacks based on machine learning,"

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

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

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

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

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

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

The research presented in this scientific article was funded by the Holy Spirit University of Kaslik (USEK)

Conflicts of Interest

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