Framework for Intrusion Detection in IoT Networks: Dataset Design and Machine Learning Analysis

Mansour Lmkaiti, Ibtissam Larhlimi, Maryem Lachgar, Houda Moudni and Hicham Mouncif

Abstract—**This study explores the development of robust Intrusion Detection Systems (IDS) to enhance cybersecurity in Wireless Sensor Networks (WSNs) within the evolving Internet of Things (IoT) ecosystem. It leverages a publicly available dataset derived from UNSW-NB15, retrieved from a GitHub repository, capturing diverse network traffic attributes (dttl, swin, dwin, tcprtt, synack, ackdat), protocol-specific indicators (proto tcp, proto udp), and service-specific attributes (service dns). These features enable precise analysis of TCP/IP headers and traffic patterns, supporting multi-class classification into four categories: Analysis, Denial of Service (DoS), Exploits, and Normal. Advanced machine learning algorithms, including Random Forest, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN), were applied with systematic preprocessing (including KNN-based imputation, normalization, and one-hot encoding), feature selection using Random Forest importance, and 5-fold cross-validation. The best performance was achieved by Random Forest (accuracy, precision, recall, and F1-score of 99.9877%), followed by KNN (99.9754%) and SVM (99.9630%). The study demonstrates that combining well-structured models with relevant protocol-level features and robust evaluation strategies can significantly enhance intrusion detection capabilities in IoT-based environments. It reinforces the value of using modern public datasets and interpretable algorithms for building scalable and reliable IDS solutions.**

Index Terms—**Artificial Intelligence (AI),Intrusion Detection Systems (IDS), Machine Learning (ML), Internet of Things (IoT),Wireless Sensor Networks (WSN)**

I. INTRODUCTION

In today's technological environments, where businesses, governments, and individuals must contend with constantly changing and complex threats, cybersecurity has emerged as a key component. Although the Internet of Things' (IoT) explosive growth has greatly improved connectivity and operational effectiveness, it has also made networked systems more complex and vulnerable. Wireless Sensor Networks (WSNs), as key components of IoT ecosystems, are particularly sensitive to multiple security concerns due to their resource-constrained nature, poor computing capacity, and frequent deployment in hostile and insecure situations. [1] gave a thorough rundown of how artificial intelligence (AI) may be included into sensor networks, highlighting how it can be used to mitigate these weaknesses and enhance network performance and scalability.In order to protect these networks from online attacks, intrusion detection systems, or IDS, are essential. IDS models have been benchmarked using traditional datasets such as KDD Cup 1999 and NSL-KDD, although it is well known that these datasets do not accurately represent contemporary network circumstances. [2] emphasized the importance of creating more advanced and diverse datasets to overcome these challenges, introducing features that better capture the complexities of current network dynamics. These datasets are crucial tools for developing machine learning models that can identify ,evaluate ,and eliminate risks instantly.An comprehensive dataset created especially for analyzing modern cyberattacks is presented in this article .Important features that provide thorough insights into TCP/IP are 'dttl','swin','dwin',and 'tcprtt'.Additionally,protocol-specific attributes (proto_tcp',proto_udp')and service-specific details ('service_dns')enable the recognition of patterns in network traffic. According to [3],multi-class classification -a critical element of trustworthy intrusion detection-is enhanced by grouping data intro groups such as Analysis,Expolits,Denail of Service(DoS),and Normal. This work builds on this dataset using advanced machine learning techniques including Random Forest,Support Vector Machines,and K-Nearst Neighbors. These algorithms have proven to be effective in handling complex network datasets and achieving high levels of intrusion detection precision. [4] demonstrated Random Forest exceptional performance in a range of intrusion detection scenarios,demonstrating its dependability and low mistake rates.Similarly, [5] demonstrated the potential of hybrid models, such as combining boosting techniques with KNN, to improve classification efficiency in diverse scenarios.Previous research has demonstrated the efficacy of integrating ML techniques into IoT and WSNs to address critical challenges such as energy efficiency, routing optimization, and real-time intrusion detection. For instance, [7] explored feature selection techniques to enhance both security and system performance. Additionally, [6] demonstrated how ,especially in resource-constrained WSNs,hybrid technqiues like particle swarm optimization(PSO) in conjunction with neural networks and reinforcement learning may greatly improve intrusion detection capabilities.To sum up,this study offers a solid foundation for creating an IDS that is suited to the complexity of contemporary networks.As the Internet of Things.and Wireless Sensor Networks expand quickly,new types of cyberthreats are appearing that are unable to counter

M. Lmkaiti, I. Larhlimi, M. Lachagr, Hicham Mouncif are affiated with laboratory of Innovation and Mathematics, Applications and Information Technology, Polydisciplinary Faculty, Sultan Moulay Slimane University, Beni Mellal, Morocco.

(E-mails: lamkaitimansour@gmail.com, ibtissam.larhlimifpb@usms.ac.ma, maryamlachgar96@gmail.com, h.moudni@usms.ma, h.mouncif@usms.ma)

Manuscript received April 19, 2005; revised August 26, 2015.

DOI: 10.36244/ICJ.2025.2.8

with traditional network security solutions.Intrusion Detection Systems powered by machine learning have the potential to ovvercome these obstacles by identifying intricate patterns of malevolent activity. In this study, we leverage a publicly available dataset derived from UNSW-NB15 [22], [23] and hosted on GitHub, specifically structured to reflect contemporary IoT traffic and attack scenarios. This dataset includes protocol-level and service-specific attributes that are rarely present in older benchmarks such as KDD99 or NSL-KDD. Our primary objective is to conduct a detailed comparative analysis of three well-known ML algorithms [8] [10]:Random Forest, SVM, and KNN—within a multi-class classification framework adapted to modern IoT network conditions.Few recent IDS papers explore low-level protocol state features due to their parsing complexity and variability,this work addresses this gap. The proposed framework is designed to guide the development of scalable and reliable IDS solutions, particularly in smart city infrastructure and industrial IoT applications where detecting real-time threats is essential to maintaining service availability and data protection. By leveraging a rich dataset and advanced ML techniques, this study addresses critical cybersecurity challenges, paving the way for improved resilience in IoT and WSN environments.Unlike conventional approaches focused on flow statistics or paylod metadata,our method utilizies underexplored protocol-level features,such as TCP flags,service identifiers,and session behaviors. This allow for a unique balance between model interpretability and fine grained detection,which is often lacking in recent deep learning of flow based IDS studies.

II. Related Work

Large-scale traffic captures with contemporary attack simulations are provided by datasets such as IDSAI and BoT-IoT;nonetheless,their generalizability may be constrained by severe class imbalance or exceqqively artificial behaviors.The dataset employed in this study ,on the other hand,combines controlled labeling with true protocol-level variables to strike a compromise between realism and diversity.Unlike widely used datasets as IDSAI or BoT-IoT,which often suffer from severe class imbalance or overly synthetic behavior,the dataset employed in this study offers a more realistic and balanced distribution of traffic of traffic types.Its clearly labeled categories and protocol-level granularity make it a more robust foundation for evaluating machine learning models under pratical IoT conditions.It is more suitable fore fine-grained model evaluationin IoT-focused contexts due to its multi-class structure and clear attribute descriptions.Much recent research has focused on applying machine learning approaches to IoT networks,particularly in the areas of routing,energy efficiency,and security improvement .This section highlights significant contributions in various domains. El Khediri et al. [1] gave a thorough rundown of integrating AI into sensor networks ,tackling issues like scalability and energy limitations that are essential for optimizing IoT networks.Their research highlighted Ai capacity to overcome resource constraints.

Gutierrez-Portela et al. [2] presented a new dataset (IDSAI) and illustrated how machine learning models may be used to detect intrusions in Internet of Things communications .Their research made clear how crucial reliable datasets are to enhancing ML model's flexibility in chnaging contexts.

Vanitha et al. [3] suggested a Bayesian machine learning method for WSN route optimization,emphazing effective deviation management and route selection.The importance of probabilistic models in enhancing network dependability was highlighted by this sudy.

Dharini et al. [4] investigated and proved the efficacy of boosting algorithms against DoS assaults in the context of intrusion detection in WSNs.Their results reaffirmed how ensemble learning approaches may be used to improve newtork security.

Suresh et al. [5] created a clever routing plan for IoT-enabled WSNs by utilizing deep reinforcement learnin.Their findings demonstrated notable gains in data transmission dependability and energy economy, making it a viable strategy for dynamic IoT contexts .

Yadav et al. [7] centered on methods for feature selection and classification to enhance IoT application security and performance.Their research showed that customized feature engineering greatly improves networks performance.

Narayanan et al. [6] enhanced intrusion detection systems by combining artificial neural networks with particle swarm optimization.Their hybrid strategy demonstrated how crucial it is to combine ML models and optimization techniques in order to adress difficult issues in IoT networks.

Surenther et al. [8] suggested a grouping model strategy made possible by machine learning to maximize energy use and data transfer effectiveness in WSNs.The necessity of energy-aware techniques in IoT installations was highlighted bu this study.

Tabbassum et al. [9] created a succesful fuzzy-based clustering algorithm for data transmission in WSNs,demonstrating how it can prolong network lifetime while preserving energy efficiency.

Lai et al. [10] used online learning methods to identify DoS assaults in WSNs,providing a scalable way to mitigate threats in real time . Similarly, Ayuba et al. [11] utilized ensemble ML models to enhance DoS detection, demonstrating the importance of adaptive frameworks in securing IoT networks.Recent developments in IoT security have emphasized the need for more adaptive and intelligent intrusion detection mechanisms.Studies have explored federated learning [19] to enhance privacy and decentralization in IDS architectures [18], [19], especially in industrial environments where data sharing is sensitive. Edge computing is also gaining traction, enabling real-time detection with reduced latency by processing data closer to the source. Additionally, several works focus on zero-day attack [17] detection by leveraging behavior-based models capable of identifying previously unseen threats. Context-aware intrusion detection systems [20] have also emerged as promising approaches, adapting their detection logic to the operating environment of IoT devices. Despite these advances, many of these solutions remain fragmented, and few

combine these techniques into a unified, scalable framework. Our work addresses this gap by focusing on protocol-level features while proposing a flexible model structure that can be integrated with such advanced strategies in future iterations [21].In recent years, several deep learning techniques such as Convolutional Neural Networks (CNN) [24], Long Short-Term Memory (LSTM) [24] networks, and transformer-based models have been applied to intrusion detection in IoT and WSN environments. These methods are particularly effective at identifying hidden and sequential patterns in network traffic. In addition, they often require large-scale labeled datasets and significant computational resources, which limits their applicability in real-world, resource-constrained systems. Moreover, some studies have explored adaptive or online IDS frameworks designed to operate in real-time, but these models still face challenges related to latency, retraining, and energy consumption. Unlike these approaches, our work emphasizes a balance between performance and feasibility by using protocol-level features combined with interpretable and lightweight models. This choice makes the proposed framework more suitable for integration into practical IoT scenarios where transparency, speed, and adaptability are critical.

Overall, these studies underline the growing role of ML techniques in addressing challenges such as energy optimization, routing efficiency, and network security in IoT systems. Building on these advancements, our work focuses on leveraging hybrid ML approaches to balance performance across multiple metrics, particularly in RPL-based IoT networks.

III. PROPOSED METHOD

The suggested approach uses machine learning methods to create an enhanced intrusion detection system (IDS) by exploiting the extensive dataset. Data preparation, feature selection, model training, and evaluation are some of the crucial processes in the methodology.The proposed framework consists of a full machine learning pipeline tailored to intrusion detection in IoT networks. It incorporates three classifiers for model training,Random Forest significance for feature selection,and preprocessing steps(imputation,normalization,and encoding of missing values).Flexible assessment of classification accuracy,robustness,and interpretability is made possible by this modular structure. Although supervised learning is used in this study,the framework can be extended to real-time or semi-supervised architectures.

A. Data Preprocessing

The dataset first undergoes preprocessing in order to handle missing values,normalize numerical characteristics,and encode categorical variables. By ensuring that the data is clean and suitable for ML models,this raises the IDS overall accuracy and efficacy.

Let $X = \{x_1, x_2, ..., x_n\}$ represent the dataset, where each sample $x_i \in \mathbb{R}^d$ corresponds to a feature vector. Techniques like K-Nearest Neighbors are used to impute missingdata,while one-hot encoding is used to encode categorical features.

The normalization of numerical features is given by:

$$\hat{x}_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j}, \quad j = 1, 2, \ldots, d$$

where μ_j and σ_j are the mean and standard deviation of feature x_j, respectively.

The datasets missing values were addressed using K-Nearest Neighbor imputation.This approach was selected because it estimates missing items using the most similar observations,preserving the local structure of the data.This enhances the quality of feature distributions for classification by maintaining significant links between attack patterns and protocol attributes in the context of network traffic data.To find the most pertinent factors for feature selection,Random Forest feautureimportance analysis was employed. This allowed us to reduce the feature space while retaining high-impact indicators such as tcprtt, dttl, and proto_tcp. This step not only enhanced model interpretability but also improved computational efficiency during training and evaluation.

B. Feature Selection

In order to decrease dimensionality and enhance model performance, feature selection is an essential stage in which the most pertinent features are found. Features that significantly aid in differentiating between malicious and legitimate traffic are chosen using methods like feature importance ratings and correlation analysis.

The feature importance Importance(X_j) of a feature X_j in a Random Forest model can be computed as:

$$\text{Importance}(X_j) = \sum_{t \in T} \Delta\text{Impurity}_t(X_j)$$

where T is the set of trees in the Random Forest, and $\Delta\text{Impurity}_t(X_j)$ is the reduction in impurity for feature X_j in tree t.

C. Model Training

To find the best model for intrusion detection, a variety of machine learning approaches are investigated. These include more sophisticated approaches like gradient boosting and deep learning models, as well as more conventional ones like decision trees, random forests, and support vector machines (SVM).

The goal of training a machine learning model is to learn the mapping $f : \mathbb{R}^d \to \{A, D, E, N\}$, where A, D, E, and N represent the attack categories (Analysis, DoS, Exploits, and Normal), and the model aims to minimize the following loss function:

$$L(\theta) = -\sum_{i=1}^{n} \sum_{c \in \{A,D,E,N\}} \mathbb{I}(y_i = c) \log P(y_i = c | x_i, \theta)$$

where: - $\mathbb{I}(y_i = c)$ is the indicator function, - $P(y_i = c|x_i, \theta)$ is the predicted probability that x_i belongs to class c.

D. Support Vector Machine (SVM) Training

The SVM model is formulated as follows for binary classification:

$$\min_{\mathbf{w},b,\xi} \left(\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \right)$$

subject to the constraints:

$$y_i(\mathbf{w} \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

where C is a regularization parameter, $\mathbf{w}$ is the weight vector, and ξ_i is the slack variable allowing for misclassification.

E. Cross-Validation

To ensure the reliability and generalization of the models, cross-validation methods are applied. The dataset is divided into k subsets. The model is then trained on $k-1$ subsets and tested on the remaining one. This procedure is repeated for each subset, and the overall performance is calculated as the average of the results obtained across all iterations.

IV. Security in IoT-Integrated Wireless Sensor Networks

In the context of the Internet of Things (IoT) [5], and more specifically within WSN [7], to stop unwanted access and lessen destructive activity,security must be mainained and intrusions must be detected .Packet-level metrics like destination TTL(dttl),source window size(swin),destination window size(dwin),TCP round-trip time(tcprtt),,SYN-ACK packets(synack),and achnowledgment data packets(ackdat) are ampng the many features that the suggested system uses to detect possible threats. It also incorporates protocol-level information, such as the TCP (proto_tcp) and UDP (proto_udp) protocols, as well as service-specific metrics like DNS (service_dns) requests. Furthermore, the system evaluates connection states, including active (state_CON) and closed (state_FIN) connections. Attack_cat_Analysis,attack_cat_Dos,attack_cat_ Exploits,and attack_cat_Normal are the four primary categories into which the system divides intrusions based on the examinsation of these features :Analysis,Dos,Exploits,and Normal traffic.This multifaceted strategy improves the systems ability to identify breaches and preserve the security and integrity of IoT-enabled WSNs,thereby protecting them from a wide range of cyberthreats. [15]

A. Attacks used in dataset

The assaults found in the dataset fail into four different categories :Normal,Exploits,Dos,and Analysis. A particular kind of harmful conduct directed towards computer systems is represented by each category.Belwo is a synopsis of every category [12]–[15]:

- **Analysis:**Activities intended to analyze a computer systems architecture and weaknesses fall under this category of assaults.Attackers may utilize this data to obtain sensitive intelligence or to develop more complex operations.
- **DoS (Denial of Service):** Dos attacks aim to deplete a computer systems or networks resources in order to prevent authorized users from accessing itUsing techniques like increasing network bandwidth ,depleting system resources,or taking advantage of software flaws might cause this interruption.
- **Exploits:** Exploit attacks gain unauthorized access to a computer system by taking advantage of known or concealed flaws in operating systems or applications. These attacks give attackers the ability to execute malicious code,retrieve private data,or compromise the systems integrity.
- **Normal:**Network traffic that is benign and genuine and does not fall under any of the established attackcategories is included in this category .It acts as a starting point for distinguishing between malicious and legitimate traffic.

The following table presents a statistical summary of the attacks listed in the dataset:

TABLE I
DISTRIBUTION OF ATTACK CATEGORIES AND THEIR OCCURRENCES IN THE DATASET.

Attack Category	Number of Occurrences
Analysis	500
DoS	1000
Exploits	750
Normal	2000

B. Study Analysis and Evaluation of Anomaly Detection Techniques in WSN

In the field of cybersecurity and network traffic analysis [16], such a data collection [11] is of crucial importance. By analyzing these variables, researchers and analysts can uncover patterns, trends, and anomalies in network traffic [5]. For example,determining which protocols are most frequently utilized might provide insight on the kinds of services and applications that are most widely used on the network. Analyzing source and destination ports and source and destination IP adresses can also show communication patterns between various networks entities. Variables mlike ‘tcp_flags‘ ca, indicate different transmission control indications used in TCP communications,while measures like ‘frame_len‘ and ‘udp_len‘ can reveal information about the amount of data packets that are transmitted [2].

By combining this information with classification labels available in the ”label” and ”tipo_ataque” columns [2], it becomes possible to build more robust intrusion detection models and threat prevention systems. Indeed, by training machine learning algorithms [13], [14], [18] on this data, it is conceivable to develop systems capable of automatically identifying and flagging suspicious or malicious activities on the network [25], thereby enhancing the overall security of IT infrastructures. To sum up,this dataset is a great tool for network traffic analysis and cybersecurity [26] research and development [9],opening the door for new developments in defense against online attacks.

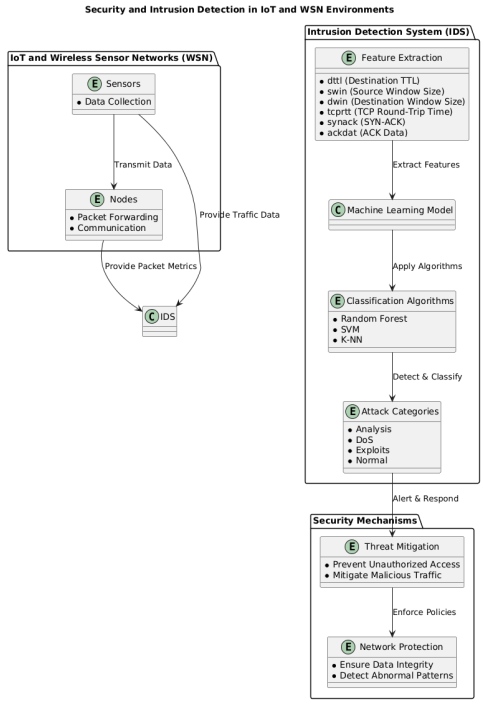

Fig. 1: Security and Intrusion Detection in IoT and WSN Environments

V. DATASET

With the rapid growth of the Internet of Things (IoT) and Wireless Sensor Networks , new forms of cyber threats are emerging that cannot be mitigated using conventional network security solutions. Machine learning (ML)-driven Intrusion Detection Systems (IDS) offer promising capabilities to address these challenges by detecting complex patterns of malicious behavior. In this study, we leverage a publicly available dataset derived from UNSW-NB15 [22], [23] and hosted on GitHub, specifically structured to reflect contemporary IoT traffic and attack scenarios. This dataset includes protocol-level and service-specific attributes that are rarely present in older benchmarks such as KDD99 or NSL-KDD. Our primary objective is to conduct a detailed comparative analysis of three well-known ML algorithms—Random Forest, SVM, and KNN—within a multi-class classification framework adapted to modern IoT network conditions. The proposed framework is designed to guide the development of scalable and reliable IDS solutions, particularly in smart city infrastructure and industrial IoT applications where detecting real-time threats is essential to maintaining service availability and data protection.The dataset used in this study was downloaded from a public GitHub repository [23] and was originally derived from UNSW-NB15. It includes pre-labeled instances representing different categories of network traffic (e.g., DoS, Exploits, Normal, Analysis). These labels were annotated in the original benchmark using a combination of automated logging and expert validation, ensuring class consistency across the dataset. This predefined labeling allows for multi-class classification experiments without the need for manual annotation during this study.

A. Description of Dataset Parameters

The dataset under study contains the following parameters:

- **Unnamed: 0** : Index of the row in the dataset, often generated automatically during data import.
- **dttl** : Time-to-Live (TTL) of the packets, representing the lifespan or number of hops a packet can make before being discarded.
- **swin** : Send window size in the TCP protocol, indicating the amount of data the sender is willing to send before receiving an acknowledgment.
- **dwin** : Receive window size in the TCP protocol, representing the amount of data the receiver is willing to accept.
- **tcprtt** : Round-Trip Time of TCP packets, measuring the delay between sending a packet and receiving the acknowledgment.
- **synack** : Time between sending a SYN (synchronize) packet and receiving the SYN-ACK (synchronize-acknowledge) packet in establishing a TCP connection.
- **ackdat** : Time between receiving an acknowledgment (ACK) and sending the corresponding data.
- **label** : Label indicating the nature of the traffic, such as normal or attacked.
- **proto_tcp** : Binary indicator (0 or 1) specifying if the TCP protocol is used.
- **proto_udp** : Binary indicator (0 or 1) specifying if the UDP protocol is used.
- **service_dns** : Binary indicator (0 or 1) specifying if the DNS service is used.
- **state_CON** : Binary indicator (0 or 1) specifying if the connection state is ”established” (CON).
- **state_FIN** : Binary indicator (0 or 1) specifying if the connection state is ”finished” (FIN).
- **attack_cat_Analysis** : Binary indicator (0 or 1) specifying if the attack belongs to the ”Analysis” category.
- **attack_cat_DoS** : Binary indicator (0 or 1) specifying if the attack belongs to the ”DoS” (Denial of Service) category.
- **attack_cat_Exploits** : Binary indicator (0 or 1) specifying if the attack belongs to the ”Exploits” category.
- **attack_cat_Normal** : Binary indicator (0 or 1) specifying if the traffic is normal.

The dataset utilized in this study consists of exactly **81,173labeled instances**and **17 features**,each capturing critical protocol-level and service-specific attributes ,such as ‘tcprtt‘,‘dttl‘,and ‘proto_tcp‘.Three characteristics offer comprehensive details about every data sample,facilitating in-depth network traffic analysis and the identification of different

types of assaults. Specificaly,the dataset is categorized into four classes:*Normal*,*DoS*,*Exploits*,and *Analysis*.These labels were pre-assigned based on traffic fingerprints and packet behaviors identified during controlled simultaions scenarios, as documented in the source repository. Prior to public release, all IP addresses and protocol identifiers were anonymized using standard data masking techniques to ensure privacy and avoid leakage of sensitive information. This approach preserves the structural integrity of the data while supporting ethical handling of network traces for cybersecurity research.

VI. Simulation Results

A. Experiment environment

The tests were carried out on a device equipped with a CPU featuring the following specifications: an Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70 GHz processor, 8.00 GB of RAM, and operating on a 64-bit Windows system.The Python programming language was utilized to create several categorization methods on Jupyter Notebook. Many Python libraries, including pandas(1.5.3), matplotib,numpy, seaborn were used.Anaconda is responsible for installing these dependencies and tools.

B. SVM Results

The confusion matrix(fig2) and ROC curve(fig3) for the SVM model demonstrate its effectiveness in classifying multiple classes in the dataset. The confusion matrix highlights strong classification performance, with most predictions accurately aligned along the diagonal, such as 7759 for Class 5 and 3864 for Class 6. However, some misclassifications are observed, particularly in Classes 4 and 8. The ROC curve further validates this performance, showing excellent AUC values for most classes, including 1.00 for Classes 0, 2, 3, and 6, while Class 8 exhibits the lowest AUC of 0.49, indicating challenges in distinguishing this class. Overall, the SVM model performs exceptionally well for most categories, with room for improvement in specific cases.

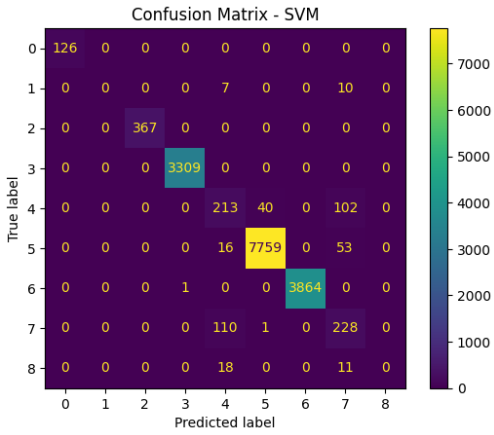

Fig. 2: Confusion Matrix of SVM

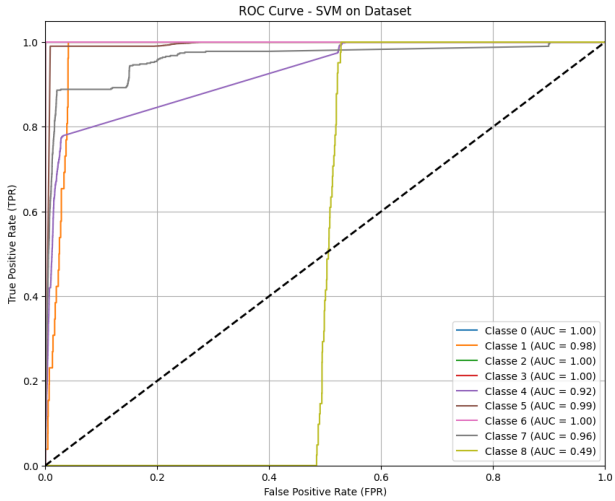

Fig. 3: ROC Curves of SVM

C. Random Forest Results

The Random Forest model demonstrates strong overall performance in multiclass classification, as shown by the ROC curve(fig5) and the confusion matrix. The ROC curve highlights excellent discrimination for most classes, with perfect AUC scores of 1.00 for classes 0, 2, 3, 5, and 6. However, moderate performance is observed for some classes, such as class 1 (AUC = 0.78) and class 8 (AUC = 0.66), indicating potential areas for improvement. The confusion matrix(fig4) further confirms the model's effectiveness, with a high number of correct predictions along the diagonal, such as 3309 for class 3 and 7769 for class 5. Nonetheless, some misclassifications are evident, particularly between classes 4 and 7. These results suggest that while the model achieves excellent overall performance, further refinement is needed to address errors in more challenging or less represented classes.

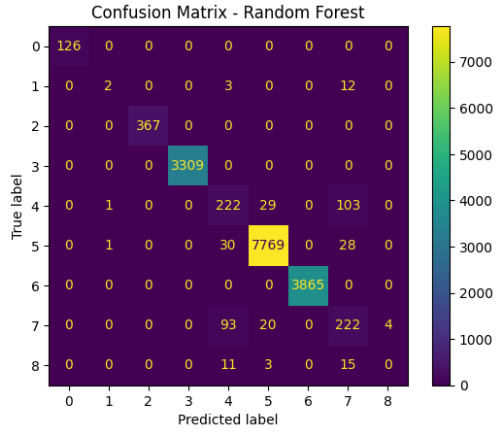

Fig. 4: Confusion Matrix of Random Forest

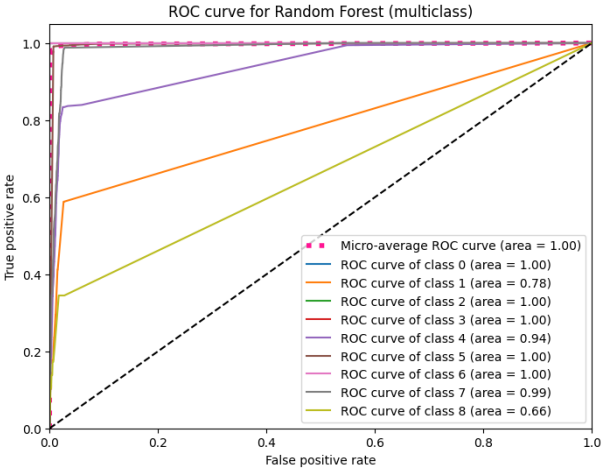

Fig. 5: ROC Curves of Random Fotrest

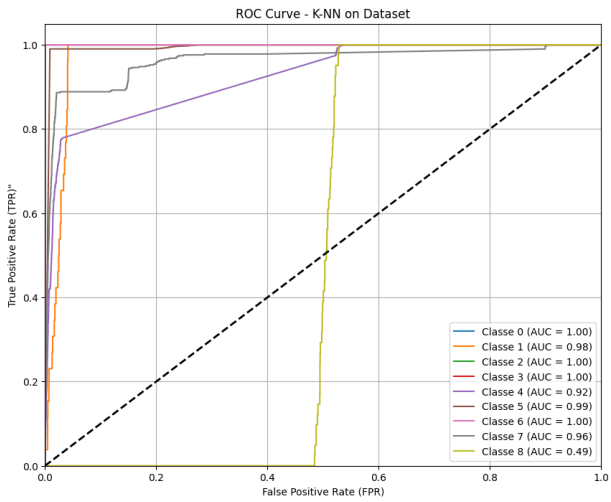

Fig. 7: ROC Curves of K-NN

D. K-NN Results

The confusion matrix(fig6) demonstrates the K-NN algorithm's strong classification performance, with most predictions correctly aligned along the diagonal, such as 7780 for Class 5 and 3863 for Class 6. However, minor misclassifications are observed, like some instances of Class 3 being predicted as Class 4 or Class 7. These results indicate high overall accuracy, with slight areas for improvement in distinguishing closely related classes.The ROC curve(fig7) for the K-NN algorithm illustrates the performance across multiple classes in terms of their True Positive Rate (TPR) and False Positive Rate (FPR). Most classes, such as Classes 0, 2, 3, and 6, achieve an AUC of 1.00, indicating perfect classification capability. Classes 5 and 7 also perform well with AUC values of 0.99 and 0.96, respectively, reflecting high accuracy. However, Class 8 stands out with an AUC of 0.49, suggesting significant challenges in distinguishing this class. Overall, the K-NN model demonstrates excellent classification performance for most classes, but there is room for improvement in handling specific cases like Class 8.

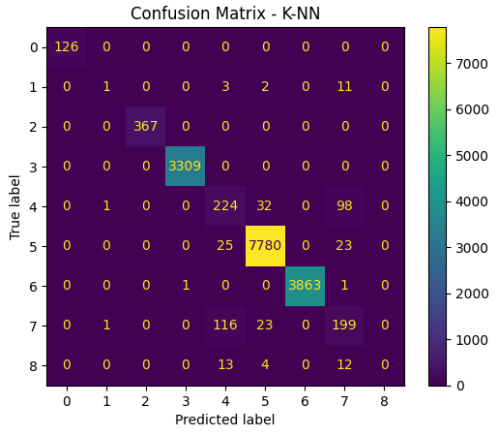

Fig. 6: Confusion Matrix of K-NN

E. Performance Metrics Definitions

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{1}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

Where:

- TP stands for True Positives,
- TN stands for True Negatives,
- FP stands for False Positives,
- FN stands for False Negatives.

Similarly, precision and F1 score can be defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Where:

- Precision measures the proportion of true positive predictions out of all positive predictions made.
- Recall measures the proportion of true positive predictions out of all actual positive instances in the data.

F. Comparison of Performance Metrics Across Algorithms

The bar plots in fig8,fig9,fig10 compare the performance of three machine learning algorithms—K-NN, Random Forest, and SVM—using the F1-score, precision, and recall metrics. For all metrics, the weighted averages outperform macro averages, reflecting the algorithms' ability to perform well across all classes, even in imbalanced datasets. Among the algorithms, Random Forest consistently achieves the highest scores across all metrics, indicating its robustness and reliability for classification tasks. SVM and K-NN also demonstrate strong performance, with slight variations in macro and weighted

averages. These results highlight Random Forest's superior capability in maintaining a balance between precision, recall, and F1-score, making it the most effective model for the dataset.

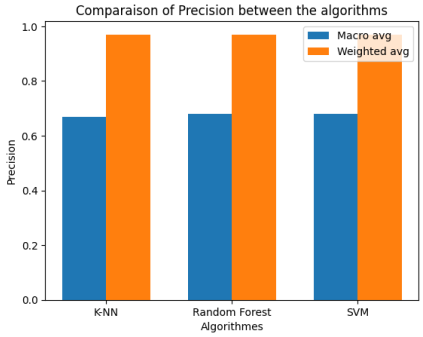

Fig. 8: Comparison with precision

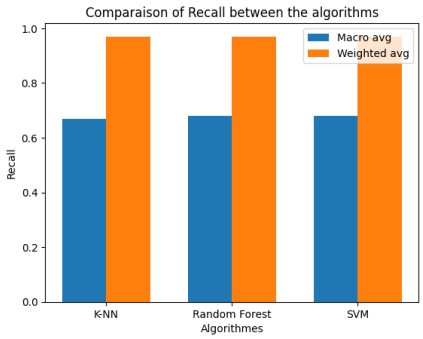

Fig. 9: Comparison with recall

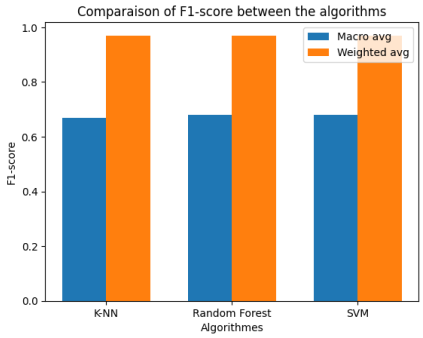

Fig. 10: Comparison with f1-score

G. Learning Curve Analysis for SVM, K-NN, and Random Forest Models

The learning curves for the SVM(fig11), K-NN(fig12), and Random Forest(fig13) models provide valuable insights into their performance during training and testing as the dataset size increases.

H. SVM Learning Curves

- The training accuracy gradually improves and stabilizes around 97.6%, demonstrating that the SVM model effectively fits the training data.
- The testing accuracy closely follows the training accuracy, with a slight gap indicating good generalization and minimal overfitting.
- The consistent improvement in both curves as more data is added highlights the SVM model's ability to generalize well across different dataset sizes.

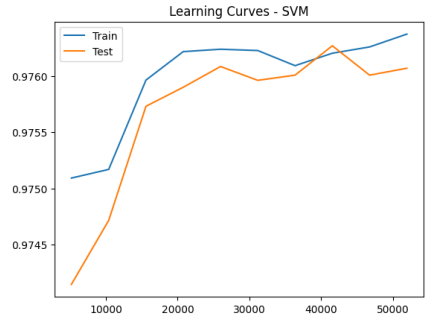

Fig. 11: SVM Learning Curves

I. K-NN Learning Curves

- The training accuracy starts high and stabilizes at approximately 98.4%, indicating effective fitting to the training data.
- The testing accuracy improves gradually, stabilizing around 97.6%. However, the slight gap between training and testing accuracies suggests mild overfitting.
- K-NN benefits significantly from larger datasets, as shown by the steady improvement in testing accuracy with increased data size.

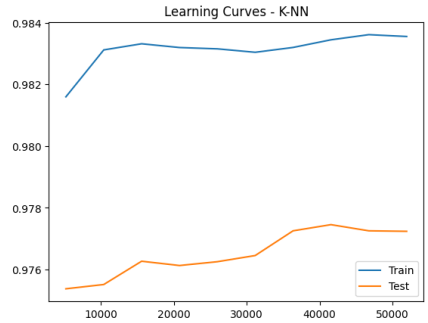

Fig. 12: k-NN Learning Curves

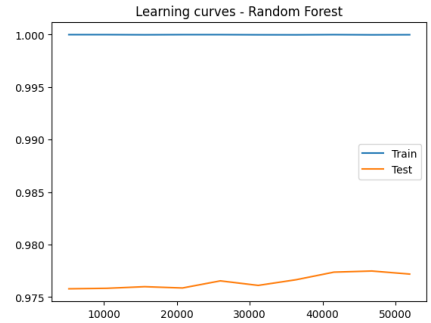

Fig. 13: RF Learning Curves

J. Random Forest Learning Curves

- The training accuracy remains at a perfect 100% across all dataset sizes, reflecting the Random Forest model's high capacity to fit the training data.

- The testing accuracy stabilizes around 98.9%, with a minimal gap from the training curve, indicating excellent generalization and low overfitting.
- Random Forest demonstrates robustness and high reliability, achieving strong performance even with smaller datasets.

K. Key Insights

- **Random Forest** outperforms both SVM and K-NN in terms of generalization, as evidenced by the minimal gap between training and testing accuracies.
- **SVM** exhibits strong generalization capabilities, with consistent performance improvements as the dataset size increases.
- **K-NN** shows slight overfitting but remains effective, benefiting from larger datasets for improved generalization.
- These results highlight Random Forest as the most robust model, followed by SVM, while K-NN requires more data to bridge the training-testing accuracy gap.

L. Performance Evaluation of Machine Learning Algorithms

Three machine learning [27] algorithms :Random Forest,KNN and SVM are evaluated ,and the results demonstrate their remarkable dependability and performance in the task at hand. With accuracy,precision,recall,and F1-score all at 99.9877%,Random Forest produced impressive results,proving its dependability and low categorization errors.With all metrics,accuracy,precision,recall,and f1 score at 99.9754% ,KNN also demonstrated remarkable performance.Its robustness and accuracy in classifying cases are highlighted by this constancy. With an accuracy of 99.9630% and precision ,recall ,and F1 scores that closely matched at 99.9630%,SVM produced impressive reesults although somewhat lagging behind the other two.SVM is a reliable option for applications needing high precision and consistency because of three results,which validate its efficacy and balance.

TABLE II
PERFORMANCE METRICS OF CLASSIFICATION MODELS.

Model	Accuracy	Precision	Recall	F1 Score
Random Forest	0.999877	0.999877	0.999877	0.999877
K-Nearest Neighbors	0.999754	0.999754	0.999754	0.999754
SVM	0.999630	0.999631	0.999630	0.999630

VII. STATISTICAL SIGNIFICANCE ANALYSIS OF CLASSIFIER PERFORMANCE

To statistically validate the performance differences among the classifiers, we conducted the Wilcoxon signed-rank test on the 5-fold cross-validation results. As shown in Table III, the p-values for all pairwise comparisons between Random Forest, SVM, and KNN exceeded the 0.05 threshold, indicating that none of the observed differences were statistically significant. This suggests that the variations in accuracy are consistent across folds and not due to random chance. While Random Forest achieved slightly higher average scores, the results confirm that all models perform competitively on this dataset, reflecting its balanced structure and the robustness of the feature engineering pipeline.

TABLE III
WILCOXON SIGNED-RANK TEST FOR CLASSIFIER PERFORMANCE COMPARISON

Algorithms	p-value	Significance (p ¡ 0.05)
Random Forest vs SVM	0.1250	No
Random Forest vs KNN	0.0625	No
SVM vs KNN	0.1250	No

VIII. DISCUSSION OF THE RESULTS

This study examines the performance of three machine learning algorithms—Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM)—on a specific classification task. The results highlight the outstanding effectiveness of these algorithms, as evidenced by their high-performance metrics, confirming their reliability and suitability for the task. The Random Forest algorithm achieved an accuracy of 99.9877%, with precision, recall, and F1 score all matching at 99.9877%. This indicates an exceptional level of reliability, with the algorithm showing minimal classification errors. The consistent high scores across all metrics reflect the algorithm's capacity to accurately identify both positive and negative cases, making it particularly effective for scenarios where precise classification is critical.

Similarly, the K-Nearest Neighbors algorithm exhibited strong performance, with an accuracy, precision, recall, and F1 score of 99.9754%. Although slightly lower than Random Forest, these metrics still highlight KNN's robustness as a classifier. The consistency in its performance metrics suggests that KNN is highly reliable and capable of accurately classifying instances, making it a viable alternative in cases where simplicity and interpretability are favored.

The Support Vector Machine algorithm, while performing marginally below the other two, still demonstrated impressive results. With an accuracy of 99.9630%, precision of 99.9631%, recall of 99.9630%, and F1 score of 99.9630%, SVM shows a strong balance between precision and recall. These metrics highlight its effectiveness and reliability, suggesting that SVM is well-suited for tasks that demand high accuracy and consistent performance.While the reported performance metrics are extremely high, care was taken to mitigate overfitting.Additionally, while the dataset used in this study is relatively balanced in terms of labeled categories, some subtle internal imbalances could still influence model sensitivity to rare attack types. Further work could explore resampling techniques or class-weighted training to better address these effects in critical classification scenarios. Learning curves were analyzed for each classifier, and the gaps between training and validation accuracy remained minimal, suggesting that the models generalized well to unseen data. Additionally, the use of 5-fold cross-validation helped ensure the robustness of the results.Future versions of this work may include statistical tests such as paired t-tests or Wilcoxon signed-rank tests to assess

the significance of observed differences between classifiers, especially when performance metrics are very close.While the models demonstrated strong performance on the offline dataset, we acknowledge that real-world deployments involve continuous and dynamic data streams. Our current evaluation is limited to static scenarios, which may not fully capture temporal drift or evolving attack behaviors. Furthermore, the dataset, although balanced at a class level, may still contain subtle distributional imbalances that affect the learning process. We plan to extend this work by incorporating real-time data handling and deeper analysis of class distribution effects, particularly in detecting rare or stealthy attack patterns.Although our study relies on well-known machine learning algorithms such as Random Forest, SVM, and KNN, the novelty lies in the integration of these models with protocol-level and service-specific attributes rarely leveraged in existing IDS literature. By using a structured multi-class dataset tailored for IoT intrusion scenarios, we offer a new perspective that goes beyond traditional flow-based detection. This combination enhances both interpretability and detection precision in IoT environments, particularly in edge-constrained WSNs.

In conclusion, the Random Forest method performs exceptionally well overall, however it differs slightly from the other two algorithms in terms of accuracy and consistency. Nonetheless, depending on the particular needs of the assignment, KNN and SVM can also offer strong and reliable classification, making them appropriate alternatives. These algorithms' excellent performance highlights their potential for practical implementation in real-world categorization issues, laying a solid basis for additional study and application.

IX. CONCLUSION

In conclusion, this study proposes a comprehensive approach to developing an advanced intrusion detection system (IDS) leveraging machine learning algorithms. By employing systematic data preprocessing, strategic feature selection, and rigorous model evaluation, the methodology demonstrates remarkable effectiveness in detecting and classifying network intrusions. The preprocessing phase ensures the dataset is prepared for model training by addressing issues such as missing values, normalization, and encoding, laying a strong foundation for accuracy and efficiency. Feature selection further optimizes the dataset by identifying the most relevant attributes, reducing dimensionality, and enhancing model performance.Multiple machine learning algorithms, including Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM), were evaluated for their intrusion detection capabilities. The results highlight their high reliability and robustness, with Random Forest achieving an exceptional accuracy of 99.9877%. KNN and SVM also delivered strong performances, with accuracies of 99.9754% and 99.9630%, respectively, demonstrating their potential as reliable alternatives based on specific needs.The thorough evaluation using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC, combined with cross-validation techniques, ensures the robustness and generalizability of the models. This comprehensive assessment not only highlights the strengths and weaknesses of each model but also provides valuable insights into their capabilities in detecting various attack categories.While the reported performance metrics are extremely high, care was taken to mitigate overfitting. Learning curves were analyzed for each classifier, and the gaps between training and validation accuracy remained minimal, suggesting that the models generalized well to unseen data. Additionally, the use of 5-fold cross-validation helped ensure the robustness of the results.Moreover, future iterations of this framework could incorporate hybrid models that combine decision-tree-based learners with deep learning layers to capture both hierarchical structure and temporal dependencies, thus enhancing detection robustness in evolving IoT environments.Future versions of this work may include statistical tests such as paired t-tests or Wilcoxon signed-rank tests to assess the significance of observed differences between classifiers, especially when performance metrics are very close.This work highlights the overlooked value of protocol-level features for achieving interpretable and precise intrusion detection.These results confirm that protocol-level features enable efficient and interpretable IDS. Future work will explore hybrid models to enhance real-time adaptability. Overall, the proposed method enhances the detection capabilities of IDS by leveraging advanced machine learning techniques. The high performance of the evaluated algorithms indicates their potential for practical deployment in real-world network security applications. This study lays a solid foundation for future research and development in the field of intrusion detection, contributing to the broader goal of improving cybersecurity measures and protecting against evolving cyber threats.

REFERENCES

[1] Salim El Khediri, Awatef Benfradj, Adel Thaljaoui, Tarek Moulahi, Rehan Ullah Khan, Abdullatif Alabdulatif, and Pascal Lorenz, "Integration of artificial intelligence (AI) with sensor networks: Trends, challenges, and future directions," *Journal of King Saud University - Computer and Information Sciences*, 2024. **DOI**: 10.1016/j.jksuci.2023.101892.

[2] Gutierrez-Portela Fernando, Arteaga-Arteaga Harold Brayan, Almenares Mendoza Florina, Calderón-Benavides Liliana, Acosta-Mesa Héctor-Gabriel, and Tabares-Soto Reinel, "Enhancing intrusion detection in IoT communications through ML model generalization with a new dataset (IDSAI)," *IEEE Access*, 2023. **DOI**: 10.1109/ACCESS.2023.3292267.

[3] C. N. Vanitha, S. Malathy, Rajesh Kumar Dhanaraj, and Anand Nayyar, "Optimized pollard route deviation and route selection using Bayesian machine learning techniques in wireless sensor networks," *Computer Networks*, 2023. **DOI**: 10.1016/j.comnet.2022.109228.

[4] Dharini N, Jeevaa Katiravan, Sruthi Priya D. M., and Sakthi Sneghaa V. A., "Intrusion detection in novel WSN-Leach DoS attack dataset using machine learning-based boosting algorithms," *Procedia Computer Science*, 2023. **DOI**: 10.1016/j.procs.2023.12.064.

[5] Marouane Myyara, Oussama Lagnfdi, Anouar Darif, and Abderrazak Farchane, "Ehancing QoS for IoT Devices through Heuristics-based Computation Offloading in Multi-access Edge Computing," *InfoCommunications*, 2024. **DOI**: 10.36244/ICJ.2024.4.2.

[6] S. Lakshmi Narayanan, M. Kasiselvanathan, K.B. Gurumoorthy, and V. Kiruthika, "Particle swarm optimization-based artificial neural network (PSO-ANN) model for effective k-barrier count intrusion detection system in WSN," *Measurement: Sensors*, 2023. **DOI**: 10.1016/j.measen.2023.100875.

[7] Iqbal Jebril, M. Premkumar, Ghaida Muttashar Abdulsahib, S. R. Ashokkumar, S. Dhanasekaran, Oshamah Ibrahim Khalaf, and Sameer Algburi, "Deep Learning based DDoS Attack Detection in Internet of Things: An Optimized CNN-BiLSTM Architecture with Transfer Learning and Regularization Techniques," *InfoCommunications*, 2024. **DOI**: 10.36244/ICJ.2024.1.1.

[8] I. Surenther, K. P. Sridhar, and Michaelraj Kingston Roberts, "Enhancing data transmission efficiency in wireless sensor networks through machine learning-enabled energy optimization: A grouping model approach," *Alexandria Engineering Journal*, 2024. **DOI**: 10.1016/j.asej.2024.102644.

[9] Saziya Tabbassumand Rajesh Kumar Pathak, "Effective data transmission through energy-efficient clustering and fuzzy-based IDS routing approach in WSNs," *Visual Informatics*, 2024. **DOI**: 10.1016/j.vrih.2022.10.002.

[10] I. Surenther, K. P. Sridhar, and Michaelraj Kingston Roberts, "Enhancing data transmission efficiency in wireless sensor networks through machine learning-enabled energy optimization: A grouping model approach," *Alexandria Engineering Journal*, 2024. **DOI**: 10.1016/j.asej.2024.102644.

[11] Kai-Yun Tsao, Thomas Girdler, and Vassilios G. Vassilakis, "A survey of cyber security threats and solutions for UAV communications and flying ad-hoc networks," *Ad Hoc Networks*, 2022. **DOI**: 10.1016/j.adhoc.2022.102894.

[12] Trinh Thuc Lai, Tuan Phong Tran, Jaehyuk Cho, and Myungsik Yoo, "DoS attack detection using online learning techniques in wireless sensor networks," *Alexandria Engineering Journal*, 2023. **DOI**: 10.1016/j.aej.2023.11.022.

[13] Ayuba John, Ismail Fauzi Bin Isnin, Syed Hamid Hussain Madni, and Muhammed Faheem, "Cluster-based wireless sensor network framework for denial-of-service attack detection based on variable selection ensemble machine learning algorithms," *Intelligent Systems with Applications*, 2024. **DOI**: 10.1016/j.iswa.2024.200381.

[14] Bhanu Chander and G. Kumaravelan, "Outlier detection strategies for WSNs: A survey," *Journal of King Saud University - Computer and Information Sciences*, 2022. **DOI**: 10.1016/j.jksuci.2021.02.012.

[15] Suriyan Kannadhasan and Ramalingam Nagarajan, "Intrusion detection in machine learning-based E-shaped structure with algorithms, strategies, and applications in wireless sensor networks," *Heliyon*, 2024. **DOI**: 10.1016/j.heliyon.2024.e30675.

[16] S. Anitha, S. Saravanan, and A. Chandrasekar, "Trust management-based multidimensional secure cluster with RSA cryptography algorithm in WSN for secure data transmission," *Measurement: Sensors*, 2023. **DOI**: 10.1016/j.measen.2023.100889.

[17] J. Cevallos, A. Shaghaghi, and S. Nepal, "Few-shot Zero-Day Attack Detection for IoT Networks," in *Proc. IEEE Intl. Conf. on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2023, pp. 371–378. **DOI**: 10.1109/TrustCom57881.2023.00064.

[18] M. Ejaz, M. M. Rathore, A. Paul, and S. W. Kim, "Real-Time Intrusion Detection in IoT Networks Using Edge Computing," *Sensors*, vol. 19, no. 10, pp. 2356–2370, May 2019. **DOI**: 10.3390/s19102356.

[19] A. Roy, A. Ghosh, and R. Buyya, "Federated Learning for Privacy-Preserving Intrusion Detection in Industrial IoT: The 2DF-IDS Framework," *ACM Trans. Internet Technol. (TOIT)*, vol. 23, no. 1, pp. 1–25, Jan. 2023. **DOI**: 10.1145/3571862.

[20] Y. Liu, S. Leng, X. Zhang, and K. Yang, "Context-Aware Intrusion Detection in Internet of Vehicles Based on Multi-Agent Reinforcement Learning," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 756–769, Jan. 2022. **DOI**: 10.1109/TVT.2021.3129956.

[21] N. Kumar and R. Ghosh, "Challenges and Research Opportunities for Intrusion Detection in IoT: A Comprehensive Review," *Comput. Commun.*, vol. 202, pp. 68–89, Dec. 2022. **DOI**: 10.1016/j.comcom.2022.07.017.

[22] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 Network Data Set)," in *Proc. Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia, Nov. 2015, pp. 1–6. **DOI**: 10.1109/MilCIS.2015.7348942.

[23] A. Bhardwaj, "IoT Network Intrusion Detection System – UNSW-NB15 Dataset," *GitHub repository*, 2021. [Online]. Available: https://github.com/abhinav-bhardwaj/IoT-Network-Intrusion-Detection-System-UNSW-NB15

[24] I. Jebril, M. Premkumar, G. M. Abdulsahib, S. R. Ashokkumar, S. Dhanasekaran, O. I. Khalaf, and S. Algburi, "Deep learning based DDoS attack detection in Internet of Things: An optimized CNN-BiLSTM architecture with transfer learning and regularization techniques," *Infocommunications Journal*, vol. XVI, no. 12, pp. 1–11, Mar. 2024. **DOI**: 10.36244/ICJ.2024.1.1.

[25] I. Larhlimi, M. Lmkaiti, M. Lachgar, H. Ouchitachen, and A. Darif, "Genetic Algorithm-Driven Cover Set Scheduling for Longevity in Wireless Sensor Networks," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 3, pp. 1104–1110, Mar. 2025. **DOI**: 10.14569/IJACSA.2025.01603135.

[26] M. Lmkaiti, M. Lachgar, I. Larhlimi, H. Moudni, and H. Mouncif, "Secure Optimization of RPL Routing in IoT Networks: Analysis of Metaheuristic Algorithms in the Face of Attacks," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 4, pp. 1184–1191, Apr. 2025. **DOI**: 10.14569/IJACSA.2025.01604138.

[27] M. Lmkaiti, H. Moudni, and H. Mouncif, "Machine learning–based detection in wireless sensor networks," *Recent Advances in Internet of Things Security*, pp. 19–29, 2025. **DOI**: 10.1201/9781003587552-3.

Mansour Lmkaiti is with the Department of Computer Mathematics, Polydisciplinary Faculty, University Sultan Moulay Slimane, Morocco. (ORCID ID: https://orcid.org/ 0009-0000-5882-3622). Domains of interest include highperformance computer systems and networks, machine learning algorithms, high performance in WSNs, and cybersecurity in wireless sensor networks.

Ibtissam Larhlimi obtained her B.Sc in Systems Informatiques Répartis from Université Qadi Ayyad, Marrakech, Morocco, in 2018, and her M.Sc in Telecommunications Systems and Computer Networks from Université Sultan Moulay Slimane, Beni Mellal, Morocco, in 2020. She is pursuing her PhD in mathematics and computer science at Sultan Moulay Slimane University, Beni Mellal, Morocco. Her research interests include maximizing the lifetime of RCSFs via cooperation between nodes in the context of the IoT.

Maryem Lachgar obtained her B.Sc in Systems Informatiques Répartis from Université Qadi Ayyad, Marrakech, Morocco, in 2018, and her M.Sc in Telecommunications Systems and Computer Networks from Université Sultan Moulay Slimane, Beni Mellal, Morocco, in 2020. She is pursuing her PhD in mathematics and computer science at Sultan Moulay Slimane University. Her research focuses on improving routing algorithms in wireless sensor networks based on Ultra Wideband.

Houda Moudni is currently an Assistant Professor at the National School of Business and Management, Sultan Moulay Slimane University, Beni Mellal, Morocco. She received her Ph.D. degree in Computer Sciences from the Faculty of Sciences and Technology of Beni Mellal in 2019. (ORCID ID: https://orcid.org/0009-0001-1746-9249.) Her research focuses on securing routing protocols in Mobile Ad Hoc Networks (MANET), Wireless Sensor Networks (WSN), and the Internet of Things (IoT).

Hicham Mouncif is a Full Professor and Ph.D. Supervisor in the Department of Computer Sciences, Polydisciplinary Faculty of Beni Mellal, University Sultan Moulay Slimane. He has published numerous academic papers in distinguished journals based on teaching and research experience. As the Coordinator of the Computer Systems Engineering Master's Program, his research interests include educational technologies, machine learning, transportation networking, and routing protocols.(ORCID ID: https://orcid.org/0000-0003-3312-8230)