IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 November 2023, accepted 27 December 2023, date of publication 2 January 2024,
date of current version 10 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3349248

RESEARCH ARTICLE

Stochastic Gradient Descent Intrusions Detection for Wireless Sensor Network Attack Detection System Using Machine Learning

HADEEL M. SALEH[1], HEND MAROUANE[2], AND AHMED FAKHFAKH[3]
[1]Continuing Education Center, University of Anbar, Ramadi 31006, Iraq
[2]National School of Electronics and Telecommunications (ENET'COM), NTS'COM Laboratory, Safax University, Sfax 3029, Tunisia
[3]Digital and Numeric Research Center of Safax (CRNS), Sfax 3029, Tunisia

Corresponding author: Hadeel M. Saleh (hadealms89@gmail.com)

ABSTRACT Communication in cyber-physical systems relies heavily on Wireless Sensor Networks (WSNs), which have numerous uses including ambient monitoring, object recognition, and data transmission. However, they are vulnerable to cyberattacks because they are connected to the IoT. In order to combat the difficulties associated with WSN intrusion detection, this research employs machine learning techniques, notably the Gaussian Nave Bayes (GNB) and Stochastic Gradient Descent (SGD) algorithms. The effectiveness of recommendation systems is improved with the introduction of context awareness. To lessen the burden on the computer, we first do a principal component analysis and singular value decomposition on the raw traffic data. On the WSN-DS dataset, the suggested SG-IDS model achieved a 96% accuracy rate, outperforming state-of-the-art algorithms with higher rates of 98% accuracy, 96% recall, and 97% F1 measurement. In an evaluation of an IoMT dataset, the SG-IDS performed admirably, with an accuracy of 0.87 and a precision of 1.00 in intrusion detection tasks.

INDEX TERMS Intrusion detection, wireless sensor network, machine learning, accuracy, Internet of Things.

I. INTRODUCTION

Wireless sensor networks (WSNs) have become increasingly common with the advent of the Internet of Things (IoT). This is due to the IoT's ability to interconnect all objects and hence drastically alter people's daily routines [1], [2]. Strict precautions are necessary to ensure the security of IoT networks. To guarantee the safety of the WSN, many safeguards, such as encryption, authentication, and other concepts, have been implemented. By contrast, threats that can circumvent common security measures have emerged as a result of the development of a wide range of attack methodologies. The safety of sensitive information is of paramount importance in large-scale WSN deployment. Protecting WSN systems requires severe security techniques and is essential [3], [4]. Unfortunately, passive defense measures are insufficient for ensuring complete WSN security. Thus, the availability of preventive safety technologies is essential [5]. An intrusion detection system (IDS) is a powerful tool for active defense [6]. Data-driven IDS can proactively detect attacks even when traditional defenses are not available. However, as the volume of data carried over a network grows, the challenge for an IDS to do real-time analysis of that data grows as well. Consider how rapidly and efficiently data processing must occur in a WSN while thinking about the advantages of an IDS [7]. Behaviors include sudden spikes in network activity and the appearance of new, unexpected WSN parameters are hallmarks of abnormal network usage. Wormholes, sinkholes, flooding, and jamming are just some of the attacks that disrupt normal network operations [8]. It may be difficult for the classifier to quickly distinguish normal and abnormal patterns in network traffic because of the large number and variety of involved data and non-involved data [9]. Difficulty in detecting suspicious behavior

The associate editor coordinating the review of this manuscript and approving it for publication was Zijian Zhang.

owing to noise and other irrelevant characteristics in the network. Such difficulty increases the time and effort required to investigate and decreases the likelihood of success [10]. Machine learning has inherent limitations that are typically determined by the data and features used in a model or algorithm. This highlights the importance of feature selection in machine learning. The widespread incorporation of computers into more and more aspects of human life has led to the widespread availability of datasets with tens of thousands of feature space dimensions. However, a finite number of characteristics can capture the entire image. The effectiveness of machine learning algorithms is severely hindered by the presence of a large number of superfluous and redundant attributes in this data subset. The combination of feature selection and machine learning has been a significant success in the scientific community, and the resulting technology has found significant applications in areas such as network traffic monitoring and security [11]. The time spent on the intrusion-detection process must be minimized without sacrificing the accuracy or detection rate. Owing to these and other differences between WSNs and traditional computer networks, traditional methods of network intrusion detection can no longer be used to safeguard WSNs. There are several variations in the terminal types, data transfer, network design, and other areas. First, a WSN IDS must reliably identify the known and unknown threats. Second, an IDS for a WSN must be lightweight, meaning that it will not significantly slow the WSN infrastructure [11]. This research proposes a new approach that is suitable for the detection of WSNs. The primary contributions of this study are as follows:

(1) The high volume and diversity of data that must be processed by the wireless sensor network are the root cause of the computationally expensive intrusion detection approach and the poor detection performance of intrusion activity. As a result, different feature selection methods, such as principal component analysis (PCA) and singular value decomposition (SVD), were investigated in this paper concerning intrusion detection in WSNs.
(2) The purpose of this research is to propose an intrusion detection model called SG-IDS by comparing and contrasting the performances of two different classification strategies: Stochastic Gradient Descent and Guessing Naive Bayes under WSN.
(3) To detect traffic attacks under a WSN with fewer false positives, an SG-IDS is utilized. Traditional methods of WSN intrusion detection suffer from several drawbacks that this model aims to remedy, including poor detection performance, slow real-time performance, and excessive results compared to other similar research.

The remaining sections of this research are organized as follows. The topic of intrusion detection in WSN is covered in Section II. Research of interest is presented in Section III. In Section IV, a plan for protecting sensor networks from intrusion is presented. The experimental environment is demonstrated in Section V. The results and analysis of the experiments are presented in Section VI. Future objectives are outlined in Section VII.

II. DETECTION OF INCURSIONS IN WSN

Two distinct types of attacks can be launched against a WSN: passive and aggressive. Destructive attacks are often referred to as ‘‘active attacks. Any entity that poses an immediate and direct threat to a system is considered an adversary. The effective information sent by the source to the destination of the destination station sent by the source station can be retrieved using a passive assault that does not disrupt the regular data connection. If malicious actors gain access to legitimate data, they can cause serious problems for the network and endanger data security. The release of confidential information does not delay the transfer of data [12]. Eavesdropping on a conversation that takes place between two nodes is an example of a passive attack, as opposed to an active attack, which exploits the broadcasting capability of the wireless communication medium. The motivation for an invasion determines whether it is external or internal. To steal private information from a WSN, it is necessary to access powerful wireless receiving and transmitting equipment. These attacks typically employ techniques, such as replay, injection, eavesdropping, and interference. This internal threat has gone on an offensive after losing a key node in the network. An internal attack can be launched by two distinct types of nodes: independent nodes, which make use of network resources without directly affecting other nodes, and compromised nodes, which can cause harm to other nodes in the network [13], and malicious sensor nodes, which eavesdrop on, interfere with, or control the communication of the entire network by masquerading as normal nodes. The energy, processing power, communication bandwidth, and storage space are limited in WSNs. Therefore, the architecture of the intrusion-detection system must be adjusted to meet the demands of each application scenario and environmental design.

III. RELATED WORK

Traditional wired network intrusion detection system (IDS) solutions are incompatible with the expansion of wireless systems [14], particularly WSN and Ad Hoc networks. The need for an intrusion detection system in a WSN is highlighted. Any unusual activity is investigated using intrusion detection systems that employ anomaly detection. A broad variety of anomaly detection systems have been developed as a result of these discoveries. Many of these systems for spotting outliers are different takes on well-established methods like artificial immunity analysis, clustering, ML algorithms, and SL models for learning statistics. Multilevel semi-supervised ML (MSML) is a framework proposed by Yao et al. [15] concerning intrusion detection models that make an effort to overcome these difficulties. First, the data were subjected to four separate procedures: fine-grained classification (FC), finding of patterns, model updates, and extraction of pure clusters. To discover all pure clusters, we first need to define

the term ''pure clusters'' and then provide a hierarchical semi-supervised k-means approach. A joint analysis based on the Hilbert Huang Transformation (HHT) was developed by Chen et al. [16] to detect LDoS attacks on WSNs. Using the nodes with the highest trust value was the primary focus of this study to develop an effective attack detection framework. The primary objective was to decrease the traffic volume, travel time, and energy consumption. Ifzarne et al. [17] improved the security of WSNs by combining both the Support Vector Machine (SVM) method of categorization and the Cuckoo Search Optimization (CSO) method. The goal of this study is to develop an appropriate and efficient method for predicting network intrusions using available datasets. The parameters of the SVM classification model were successfully parallelized over several nodes using the map-reduction technique, allowing the method to meet the target. The limitations of this investigation include subpar results, sluggish response time, and excessive misclassification rate. Liu et al. [18] utilized the EM technique of Expectation Maximization to identify anomalous data in the NSL-KDD repository. In this post, we looked at several different types of attacks, including synfloods, land, ping of death, sweeping, and UDP floods. Hemanand et al. [19] advocated deploying the existing Glowworm Swarm Optimization method across IoT sensors to identify power-hungry devices and allocate resources equitably. This would allow smart and sustainable energy management. When evaluating a network's performance, the routing protocol should be taken into account, as suggested by Jayalakshmi et al. [20], who argued in favor of strengthening network security by using cryptography at every node. As proposed by Tauqeer et al. [26], based on these suggestions, NIDSs can incorporate feature selection models to enhance their functionality. The development of this concept was motivated by the multitude of optimization strategies available, such as particle swarm optimization (PSO), grey wolf optimization (GWO), firefly algorithm (FFA), and genetic algorithm (GA). The proposed paradigm aims to improve the NIDS performance by utilizing 13 sets of rules derived from the algorithms mentioned above. These deployments were accomplished with Python's Anaconda open-source and its wrapper-based techniques. The UNSW-NB15 dataset and the SVM and J48 ML classifiers were used to assess the quality of the proposed model.

Vinayakumar et al. [21] collected data for the MQTTset dataset, which is optimized for the MQTT protocol used by many IoT devices. Researchers have created a fake detection system that combines the official dataset with cyber-attacks against an MQTT network to prove that the dataset is reliable. These findings demonstrate that MQTTset can be utilized to educate ML models for detection systems that guarantee the security of IoT ecosystems. Kumar et al. [22] developed a novel approach to detecting intrusions based on unauthorized use. This system can identify all five types of network attacks (exploit, DOS, probe, generic, and normal attacks). KDD99 or NSL-KDD 99 data collection is also commonly utilized in IDS-related activities. These records are considered useless and outdated for detecting modern threats. This study uses the UNSW-NB15 dataset as a non-online data source to develop an integrated classification-based method for identifying cybercrime.

As shown by the research carried out by Chandre et al. [23], the success rate of an attack on MQTT-based IoT system can be estimated using one of several readily accessible machine learning models. Evaluation criteria such as precision, accuracy, and F1 score were used to make direct comparisons between the models' performance. Based on the outcomes, it was clear that random forest achieved near-perfect accuracy. Hemanand et al. [24] developed a clever IDS by combining a linear support vector machine (LSVM) and Cuckoo Search Greedy Optimization (CSGO) models to optimize wireless sensor network safety. This model was tested on well-known network datasets, such as NSL-KDD and UNSW-NB15. The first step in normalizing the attributes is to perform dataset preprocessing, which involves removing any unnecessary data, making educated guesses about the values that are absent, and applying any necessary filters. The CSGO algorithm needs to be given the optimal number of features, which is calculated during the pre-processing stage, for it to be able to select the best possible features. The last step is to predict whether the label should be considered normal or abnormal by utilizing a machine-learning classification algorithm based on LSVM. During the process of evaluating the findings, a multitude of performance measurements was utilized to verify and assess the efficacy of the proposed security model.

The dataset utilized by Almomani [25] was simulated in an NS-2 network simulator using the LEACH routing protocol. This simulation aimed to collect data from the network and preprocess it, resulting in the generation of 23 characteristics that categorized the condition of the relevant sensor. In addition, the simulation incorporated the implementation of five different types of Service (DoS) assaults. The constructed CNN-LSTM model was assessed over 25 epochs, yielding accuracy, precision score, and recall score values of 0.944, 0.959, and 0.922, respectively. These scores are measured on a scale ranging from 0 to 1.

In order to spot cyberattacks, Tauqeer et al. [26] suggested a trifecta of Machine Learning algorithms: Random Forest, Gradient Boosting, and Support Vector Machine (SVM). The most effective models for detecting cyberattacks are those trained with machine learning. The WUSTL EHMS 2020 dataset includes major in-the-middle, data injection, and spoofing attacks, and is used to test proposed Machine Learning models. The outcome analysis demonstrates that the proposed Machine Learning models are superior to the state of the art.

A new lightweight IDS technique is developed by Taouali et al. [27] to successfully counter a wide variety of cyberattacks in IoMT networks. Pre-processing, feature selection, and judgment are the three stages that make up the proposed anomaly-based intrusion detection system. Cleaning and standardizing data is done in the pre-processing

stage. In order to improve detection results and minimize the dimensionality of retrieved features, the proposed method employs two data-driven kernel approaches, namely kernel principal component analysis and kernel partial least square techniques, during the feature selection phase. Therefore, the kernel extreme learning machine is employed in the decision step to determine if the traffic flow is benign or malicious. An up to date IoMT dataset called WUSTL-EHMS-2020 is considered for evaluating and discussing the produced results, to verify the efficacy of the established detection technique. 99.9% accuracy, 99.8% specificity, 100% sensitivity, and 99.9% F-score were all attained by the proposed approach.

An extreme Gradient Boosting (XGBoost) approach is proposed by Dhanya and Chitra [28] to identify malicious software in databases. Differential Evolution (DE), an intelligent evolutionary technique, is used to optimize the XGB algorithm's hyperparameters. The experiment was run on the WUSTL EHMS 2020 Dataset for IoMT CyberSecurity, and after hyperparameter optimization, it achieved an accuracy of 97.39 percent. When it came to malware identification, the DE-optimized XGB Classifier excelled in both accuracy and speed.

IV. WSN INTRUSION DETECTION ARCHITECTURE

There are three main parts to intrusion detection technology for WSNs: data collection, detection, and reaction. The detection system receives data from the environment gathered by the information-collecting module. The detection module has an analyzer that examines and evaluates the acquired data traffic information to determine whether an intrusion has occurred within the WSN. When something out of the ordinary is found, the detection module relays the information to the response module. Figure (1) demonstrates how WSN can be utilized for detecting intrusions. Sensor nodes (SN), cluster heads (CH), and sink nodes (sink) are all types of nodes in a such network [29], [30].

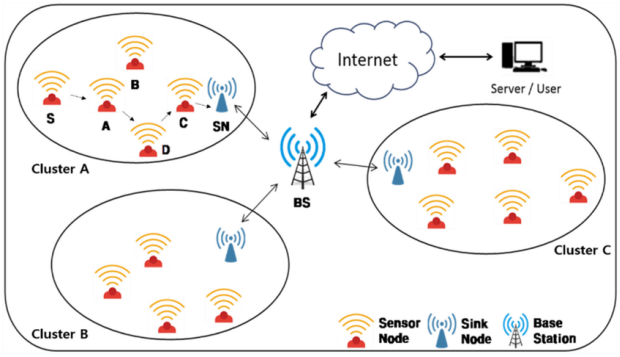


FIGURE 1. Basic Wireless sensor network intrusion detection architecture [30].

To reduce communication burden and energy consumption, this network employs distributed intrusion detection. The leader of a cluster in distributed detection coordinates the various computational operations being performed all over the world. The communication overhead can be lowered by sending less data to the cluster head, and the processing expense of the cluster head can be offloaded to normal sensor nodes [31]. Many researchers seeking superior IDS performance in WSN have turned to more intricate data mining strategies. However, because of the high processing cost that they entail, such methods are not suitable for application in real-time systems that utilize wireless sensor networks. The large feature dimensions of the input data, availability of duplicate data, and insufficient data preparation are the most important contributors to the high computational cost of an IDS.

V. THE PROPOSED RESEARCH METHODOLOGY

Feature selection is used to reduce the number of potential features from a large pool to a more manageable size. Pre-processing techniques like principal component analysis and singular value decomposition are frequently used to enhance the detection accuracy of the classification algorithm, reduce the computational load of the IDS, and preserve as much information as feasible. Algorithm (1) presents a data analysis module that utilizes these methods to assess whether observed behavior constitutes an intrusion. Figure (2) provides an overview of the architectural configuration of the proposed system. The first focuses on different approaches to data manipulation in this context.

These steps in algorithm 1 and algorithm 2 describe the proposed algorithm procedure. Firstly, load the input called the WSN-DS data set and called WUSTL EHMS 2020 Dataset, the output is Intrusion Classification. Then define the parameters needed for PCA, and define the parameters required for GNB, set up the hyper parameters for SGD such as the learning rate and the number of iterations performed here.

Furthermore, in order to store true labels and predicted labels, lists will be created. Then performing SVD and PCA. The next step is to split the data set into training (70) % and testing (30) % and moving on to training the models. This step consists of two phases which are train GNB and train SGD. Finally, evaluation performance should be performed in order to Accuracy, precision, recall, and F1 score.

The recommended intrusion detection method uses numerous machine-learning methods to improve intrusion classification in Wireless Sensor Networks. First, the input data, WSN-DS, was loaded. A sequence of activities was then performed under well-defined hyperparameters. Preprocessing with Principal Component Analysis (PCA) reduces dataset dimensionality, making feature extraction easier. The data are then enhanced, and relevant properties are identified using Singular Value Decomposition (SVD). The dataset had two sets: training and testing. The dataset comprised 70% training data and 30% testing data. Two models, GNB and SGD, were trained throughout training. After computing the mean and covariance matrix for each class, GNB calculates the class probabilities. The Stochastic Gradient Descent (SGD) model is trained iteratively using scrambled training data and gradient descent to update the model parameters using the estimated loss gradient.

Algorithm 1 The SG-IDS Algorithm Specification

Input: Load the dataset called WSN-DS.
Output: Intrusion Classification
Start
Step 1: Define the parameters needed for PCA, such, as the number of components (num_components).
Step 2: Define the parameters required for GNB including probabilities, mean vectors and covariance matrices.
Step 3: Set up the hyperparameters for SGD like learning rate and the number of iterations.
Step 4: Create lists to store true labels and predicted labels.
Step 5: Perform PCA;

- Apply PCA to reduce the dimensionality of our dataset down to num_components.

Step 6: Perform SVD;

- Use SVD to further preprocess our data and select features.

Step 7: Split the dataset;

- Divide the dataset into training (70%) and testing (30%) sets.

Step 8: Moving on to training the models;
Firstly, lets train GNB (Gaussian Naive Bayes);

- For each class in our dataset
- Calculate mean and covariance matrix for each class.
- Compute probability for each class.

Secondly, train SGD (Stochastic Gradient Descent);
For each iteration

- Randomly shuffle our training data.
- For each data point (x,y) in the training data
- Calculate gradient of loss function with respect, to model parameters.
- Update model parameters using descent; parameters = learning_rate ∗ gradient

Step 9: Test the trained models;
For every data point (x_test, y_test), in the testing set.

- Apply PCA and SVD transformations
- Make predictions using GNB;
- Determine class probabilities using the GNB formula.
- Select the class with the probability as the predicted label.
- Make predictions using SGD;
- Calculate the decision boundary based on parameters trained with SGD
- Classify based on this decision

Step 10: Evaluate Performance;
For each predicted label and true label;

- Print evaluation metrics Accuracy, precision, recall and F1 score

End

In the assessment stage, the trained models were evaluated. PCA and SVD altered each data point in the testing set to align the data for prediction. The GNB model predicts labels using class probabilities by selecting the most likely class. The Stochastic Gradient Descent (SGD) model is predicted by setting the decision boundary using the SGD-trained parameters. For each projected and real label, the accuracy, precision, recall, and F1 score were calculated to measure the performance. The invasion detection accuracy of the algorithms was fully demonstrated using these measures.

This method is often referred to as ''data engineering.'' This is an essential part of the learning process. The three stages of data processing are cleaning, normalization,

Algorithm 2 The SG-IOMT Algorithm Specification

Input: Load the dataset called WUSTL EHMS 2020 Dataset.

Output: Intrusion Classification

Star

Step 1: Define the parameters needed for PCA, such, as the number of components (num_components).

Step 2: Define the parameters required for GNB including probabilities, mean vectors and covariance matrices.

Step 3: Set up the hyperparameters for SGD like learning rate and the number of iterations.

Step 4: Create lists to store true labels and predicted labels.

Step 5: Perform PC;

- **Apply PCA to reduce the dimensionality of our dataset down to num_components.**

Step 6: Perform SV;

- **Use SVD to further preprocess our data and select features.**

Step 7: Split the datase;

- **Divide the dataset into training (70%) and testing (30%) sets.**

Step 8: Moving on to training the models.

Firstly, lets train GNB (Gaussian Naive Bayes);

- **For each class in our dataset**
- **Calculate mean and covariance matrix for each class.**
- **Compute probability for each class.**

Secondly, train SGD (Stochastic Gradient Descent);

For each iteration

- **Randomly shuffle our training data.**
- **For each data point (x,y) in the training data**
- **Calculate gradient of loss function with respect, to model parameters.**
- **Update model parameters using descen; parameters = learning_rate * gradien**

Step 9: Test the trained model;

For every data point (x_test, y_test), in the testing se;

- **Apply PCA and SVD transformation**
- **Make predictions using GNB;**
- **Determine class probabilities using the GNB formul**
- **Select the class with the probability as the predicted label**
- **Make predictions using SGD;**
- **Calculate the decision boundary based on parameters trained with SGD**
- **Classify based on this decision.**

Step 10: Evaluate Performance.
For each predicted label and true label;

- **Print evaluation metrics Accuracy, precision, recall and F1 score**

End

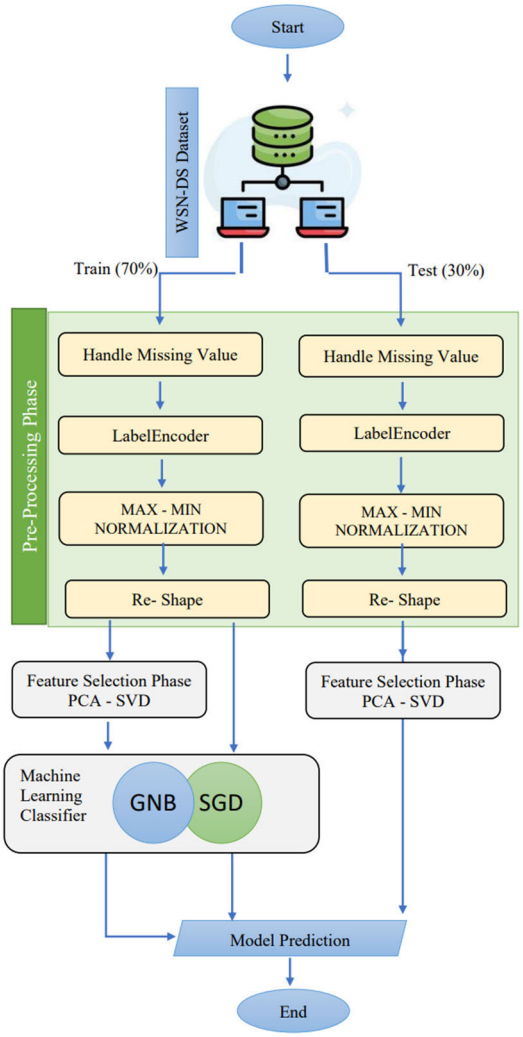


FIGURE 2. WSN-DS dataset algorithm framework specifications.

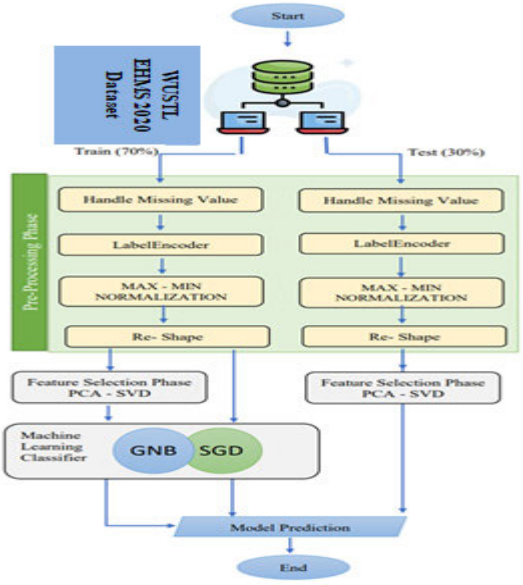


FIGURE 3. WUSTL EHMS 2020 Dataset algorithm framework specifications.

TABLE 1. Characteristic of wsn dataset.

Feature number	Symbol	Feature name	Description
1	id	Node Id	A unique ID number of the sensor node
2	Time	Time	The run-time of the node in the simulation
3	Is_CH	Is CH	Describes if the node is a CH or not
4	Who_CH	Who CH	Cluster head ID
5	Dist_To_CH	Distance to CH	Distance between node and CH
6	ADV_S	ADV CH sends	Number of the advertise CH's broadcast messages sent to nodes
7	ADV_R	ADV CH receives	Number of advertise messages received by the nodes from CH
8	JOIN_S	Join request send	Number of join request messages sent by the nodes to the CH
9	JOIN_R	Join request receive	Number of join request messages received by CH from nodes
10	SCH_S	ADV SCH sends	messages of TDMA schedule broadcast sent to the nodes
11	SCH_R	ADV SCH receives	Number of scheduled messages received by the CH
12	Rank	Rank	Node order in TDMA scheduling
13	DATA_S	Data sent	Number of data packets sent from the node to its CH
14	DATA_R	Data received	Number of data packets received by the node from the CH
15	Data_Sent_To_BS	Data sent to BS	Number of data packets that are sent from node to the BS
16	dist_CH_To_BS	Distance CH to BS	Distance between CH and BS
17	send_code	Send code	The sending code of the cluster
18	Consumed_Energy	Energy consumption	Energy consumed
19	Attack type	Attack type	Type of attacks or normal traffic

and feature selection. A filter-based method influenced by principal component analysis and singular value decomposition was used to extract the most important properties. The model can then be trained by using the training set when the necessary feature vector is selected. After the model has been trained, it can be checked against the validation set for accuracy. The validated model was then applied to the test dataset for analysis.

A. THE DATA PRE-PROCESSING

1) COLLECTING AND MAPPING INFORMATION

The provided data includes a label attribute consisting of alphabetical characters that must be converted to numeric values to exclude it from the methodology. The attack classification comprises five distinct types, namely "Normal," "Blackhole," "Grayhole," "Flooding," and "TDMA." Because quantifying such information is not feasible, ordinal numbers ranging from 0 to 4 are employed to arrange the data logically. Please refer to Table (1) for the required alterations and adjustments.

The second dataset is the WUSTL EHMS 2020 Dataset which serves as a pivotal resource for researchers delving into the realm of cybersecurity within the Internet of Medical Things (IoMT). This real-world dataset, meticulously collected from an Enhanced Healthcare Monitoring System (EHMS) testbed [32], intricately combines both network flow metrics and patients' biometric data, establishing a unique and comprehensive foundation for the in-depth exploration of IoMT cybersecurity challenges. Comprising 44 features, including 35 network flow metrics capturing elements such as IP addresses, port numbers, protocol types, and packet lengths, alongside eight patients' biometric features encompassing vital health indicators, the dataset culminates in a singular label feature discerning the nature of network flows as benign or malicious as illustrated in Table (2). This amalgamation of diverse characteristics positions the WUSTL EHMS 2020 Dataset as an invaluable asset for researchers, fostering applications ranging from intrusion and anomaly detection to the detailed characterization of various IoMT cybersecurity attacks.

TABLE 2. Characteristic of IOMT dataset.

Feature	Description
1. Dir	Direction
2. Flgs	Flags
3. SrcAddr	Source Address
4. DstAddr	Destination Address
5. Sport	Source Port
6. Dport	Destination Port
7. SrcBytes	Source Bytes
8. DstBytes	Destination Bytes
9. SrcLoad	Source Load
10. DstLoad	Destination Load
11. SrcGap	Source Gap
12. DstGap	Destination Gap
13. SIntPkt	Source Packet Time
14. DIntPkt	Destination Packet Time
15. SIntPktAct	Source Packet Actual Time
16. DIntPktAct	Destination Packet Actual Time
17. SrcJitter	Source Jitter
18. DstJitter	Destination Jitter
19. sMaxPktSz	Source Maximum Packet Size
20. dMaxPktSz	Destination Maximum Packet Size
21. sMinPktSz	Source Minimum Packet Size
22. dMinPktSz	Destination Minimum Packet Size
23. Dur	Duration
24. Trans	Transmission
25. TotPkts	Total Packets
26. TotBytes	Total Bytes
27. Load	Network Load
28. Loss	Loss
29. pLoss	Packet Loss Ratio
30. pSrcLoss	Source Packet Loss Ratio
31. pDstLoss	Destination Packet Loss Ratio
32. Rate	Data Transfer Rate
33. SrcMac	Source MAC Address
34. DstMac	Destination MAC Address
35. Packet_num	Packet Number
36. Temp	Temperature
37. SpO2	Oxygen Saturation
38. Pulse_Rate	Heart Rate
39. SYS	Systolic Blood Pressure
40. DIA	Diastolic Blood Pressure
41. Heart_rate	Heart Rate
42. Resp_Rate	Respiration Rate
43. ST	ST Segment in ECG
44. Label	Data Label or Classification

2) LABEL ENCODER

The nominal and ordinal feature names are in a set of categories represented as strings. Some labels might have to order information (ordinal qualities), whereas others might not (nominal features). Throughout the data pre-processing phase, it is vital to encode labels as numbers to guarantee that the learning algorithm correctly interprets the characteristics. LabelEncoder uses a numerical encoding system to assign values to the labels.

3) MAXIMUM AND MINIMUM NORMALIZATION

Several classification techniques are significantly impacted by the data's range of features, from less than one to hundreds of thousands, necessitating the normalization of continuous data. Here, we employed the outliers in Eq. (1) to provide a baseline for comparison. where xj is the original data for the j-dimensional feature, Minj is the minimal value of the feature, Maxj is its maximum value, and xj is the normalized data of the feature [33].

$$x_j^* = \frac{x_j - Min_j}{Max_j - Min_j} \tag{1}$$

B. FEATURES EXTRACTION

1) PRINCIPAL COMPONENT ANALYSIS

When faced with the difficulty of identifying patterns in high-dimensional data, many researchers turn to principal component analysis (PCA). The goal of PCA is to use a smaller number of typical feature images (called Eigenobjects) to represent both recognized and unfamiliar faces. PCA is useful for detecting and validating facial features, as shown by statistical data. In order to use principal component analysis (PCA), a two-dimensional matrix of face images must be transformed into a one-dimensional vector. A one-dimensional vector can be oriented in either a row or column without affecting its value [22], [23].

2) SINGULAR VALUE DECOMPOSITION

Another method of data splitting is singular value decomposition (SVD). They are used for several purposes in signal processing and statistics, such as finding patterns and extracting features from matrices. However, PCA cannot extract features from a single signal, nor can it disclose information about features contained in a signal with varying frequencies. SVD can be a more useful approach for feature extraction than principal component analysis because frequency differences may mask genuine differences between physiological states [34], [35].

C. CLASSIFICATION MODEL

To detect intrusions, the SG-IDS classification algorithm was combined with data from wireless sensor networks that were preprocessed using a sequence backward feature selection technique. A fast, decentralized, high-performance SG-IDS is a gradient-based boosting framework. The foundation of SG-IDS is a simplified histogram-based training method that uses fewer features and samples overall.

1) GAUSSIAN NAIVE BAYES

The Machine Learning (ML) classification approach, known as Gaussian Naive Bayes (GNB), employs a probabilistic strategy and Gaussian distribution. In a Gaussian Naive Bayes, every input parameter (or predictor) is assumed to have an independent predictive power over the outcome. Based on this final prediction, we calculated the likelihood that the dependent variable would fall into each of the predetermined categories. When the competing groups' probabilities are identical, the group with a higher likelihood will prevail. Feature probabilities, as shown in Eqs. (2), is assumed

to follow a Gaussian distribution as follows:

$$p\left(x_i|y\right) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{\left(x_i-\mu_y\right)^2}{2\sigma_y^2}\right) \tag{2}$$

The continuous variable X's variance and mean are calculated for each Y category using the aforementioned procedures. Figure 4 shows the Naive Bayes classifier wiche is an important part of machine learning and probabilistic modeling. The Naive Bayes classifier is a learning system that uses rules from math, called Bayes' theorem. It guesses probabilities by making ideas about features being alone or separate. The picture makes it easy to see the main parts and steps used in classifying things with Naive Bayes. The information was classified in accordance with their degree of similarity. The GNB not only considers the distance from the mean when determining this proximity to the mean, it also considers how this distance relates to the class variance [36].

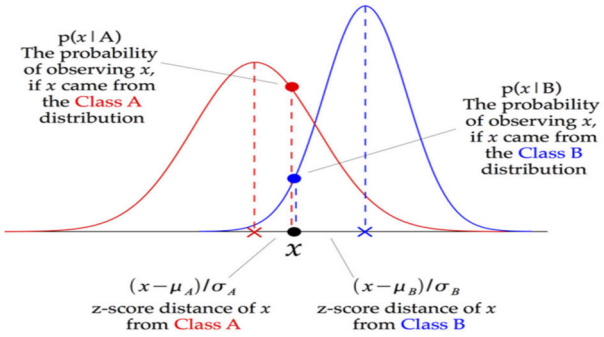


FIGURE 4. Graphical representation of the naive Bayes classifier.

2) STOCHASTIC GRADIENT DESCENT

This model is a powerful method of training linear classifiers. It is sufficient to replace the real item with a less precise estimate of the gradient. To estimate the gradient of the cost function, stochastic gradient descent gives a gradient to each component of the learning process. Several parameter changes were performed to account for the expected variations. Whenever new training data was added, the model parameters were updated. Stochastic gradient descent yields vastly better outcomes than the conventional method when working with big datasets [37]. This model of facilitation works well. The simplified SGD revisions in Eq. (3) are as follows:

$$\theta^{(t+1)} = \theta^{(t)} - \alpha_t \nabla l_i\left(\theta^{(t)}\right) \tag{3}$$

Both are representations of the size of the learning set employed to fine-tune the parameters and t represents the number of iterations. Here, a new value was arbitrarily assigned to index I before each iteration. In practice, however, it is typical practice to jumble the samples before analysis [38].

VI. EXPERIMENTAL SETTINGS

The WSN-DS dataset [39], which is publicly available, was used in this study. Designed to work with WSNs, this dataset includes indicators of potential intrusion. Regarding the WSN-DS, the four most common types of DoS attacks are black holes, gray holes, floods, and schedule attacks. Comprehensive statistics are shown in Table 3. The number of randomly selected samples from the training set was 224796 (approximately 70%), whereas the number of randomly selected samples from the testing set was 149865 (30%).

TABLE 3. WSN-DS Dataset Class Label Description.

Class	Description
Normal	Logs of a Typical Link
Blackhole	DoS attacks are launched against the LEACH protocol when an attacker advertises themselves as a CH at the very beginning of the attack.
Grayhole	An attacker launches a denial-of-service attack against the LEACH protocol by advertising themselves as a CH to other nodes at the very beginning of the attack.
Flooding	An attacker can target the LEACH protocol in a variety of ways.
Scheduling	Attacks on LEACH's scheduling infrastructure occur during initialization.

CM was used to assess the dataset's accuracy, recall, precision, and F-measure. According to sources [40], [41], equations (4)– (7) illustrate how to strike a balance between the number of false positives and the ratio of false-negative results to the total. Precision is defined as the fraction of training data that is properly labeled. The proportion of "good" constituents that were accurately categorized into the "good" cluster. Accuracy refers to the proportion of false positives that arise when employing a detection model that initially misidentifies certain elements as negative. The F-score is equivalent to the arithmetic mean [42], [43], [44].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

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

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F_1 = 2 * \frac{precision * recall}{precision + recall} = \frac{2TP}{2TP + FP + FN} \tag{7}$$

VII. RESULTS ANALYSIS AND DISCUSSION

This research test demonstrated the effectiveness of the proposed model. Multiple independent experiments were performed. The first goal of the current study is to compare the performance of SG-IDS with and without a feature extraction phase using PCA/SVD with 10 or 15 features. The second goal was to evaluate SG-IDS's efficacy concerning other machine-learning classification strategies. The third goal of this study is to examine SG-IDS's performance

of SG-IDS in the face of four different types of attacks. For a network to be usable and eventually integrated with its intrusion detection mechanism, the precision and recall metrics must reach a sufficiently high-performance level for detection purposes. When applied to the WSN-DS dataset without first selecting features, standard classification methods such as Gaussian Naive Bayes (GNB) and Stochastic Gradient Descent (SGD) are compared with DNN [21] and Deep CNN [23] in Table (4).

TABLE 4. Measurements from several classifications on the WSN-DS dataset with no feature selection phase.

Algorithms		Measures			
		Accuracy	Precision	Recall	F1-score
DNN [21]		0.98	0.93	0.87	0.90
Deep CNN [23]		97	94	92	90
SG-IDS	GNB	0.82	0.81	0.82	0.79
	SGD	0.95	0.98	0.95	0.96

Sample sets can undergo feature selection/dimensionality reduction with the help of the feature selection module, leading to improved estimator accuracy or enhanced performance on high-dimensional datasets. Ten or 15 features were used by the two algorithms (PCA and SVD) in this study. The outcomes of applying feature selection with ML algorithms to the WSN-DS dataset are depicted in Figures (5) – (7). Two efficient supervised machine learning algorithms SG-IDS-ML that have been demonstrated with high accuracy, precision, recall, and F1-score were used in the study: Gaussian Naive Bayes (GNB) and Stochastic Gradient Descent (SGD). The two algorithms yielded a correct classification of instances with accuracies of 0.87 each demonstrating their effectiveness. However, GNB and SGD reached a recall score of 0.87 and obtained a precision score of 1 showing no false positives. A balanced tradeoff between precision and recall, F1-scores of 0.93 for both algorithms. In this case, it shows that SG-IDS-ML algorithms such as SGD and GNB are good tools for identifying intrusions and consequently classification of traffic incidents.

Fascinating findings are revealed while evaluating several algorithmic approaches under diverse feature selection techniques. Without feature selection, GNB and SGD do not perform equally well. The accuracy stands at 0.81, precision is at 0.82, recall is at 0.79, while F1-score is at 0.82. Yet another approach for uniform outcomes in GNB, as well as SGD, could incorporate a PCA comprising of ten factors or an SVD comprising of ten factors. They provide for an accuracy of 0.81 and 0.98, a precision rate of 0.82 and 0.95, a recall level of 0.79 and 0.96, and finally, an F1 score of 0 Nevertheless, using PCA with 15 factors leads to a peculiarity on

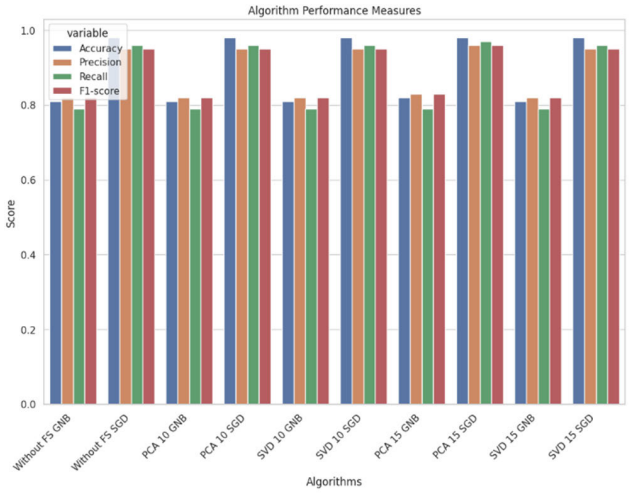


FIGURE 5. Comparison of multiple classification measurements for WSN-DS.

GNB's accuracy at 82, while SGD retains its high accuracy of 98. For precision, recall, and F1-score of both algorithms, they were reasonably stable across the three feature selection methods with 83, 79, and 83 as well as 96, 97, and 96, respectively. In summary, these findings demonstrate that feature selection techniques either maintain or improve the effectiveness of intrusion detection. The statistical summary highlights the data with a mean accuracy of 0.896 and a low standard deviation (0.088594). In all this, it suggests a uniform degree of correctness even as they were experimented on.

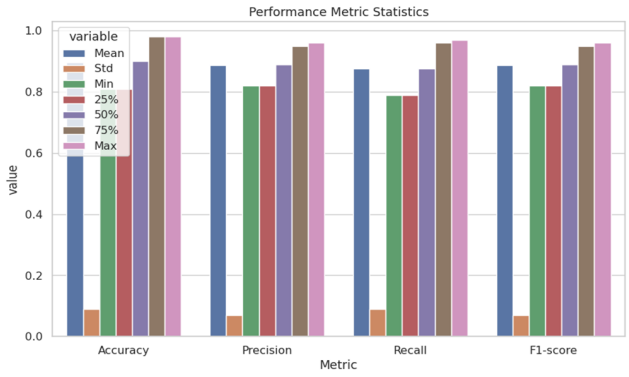


FIGURE 6. Performance Metric Statics for WSN-DS.

The mean of the precision and F1-score equals 0.887, indicating a balance of the correct identification with the precision and recall tradeoff. Despite the high standard deviation, the variability is relatively low. The recall (mean = 0.876) measures the ability of the model to capture true positives. Quartile values highlight reliable models for different features or experiments. These statistical insights cumulatively imply a robust and effective system. Therefore, the intrusion detection algorithms utilized were efficient and reliable.

FIGURE 7. Comparison of multiple classification measurements for IomT.

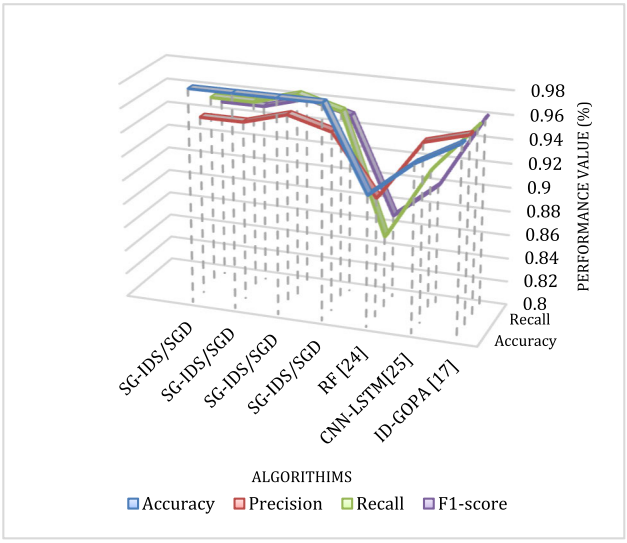


FIGURE 8. Evaluation of Several Classification Metrics.

Gaussian Nave Bayes (GNB) and Stochastic Gradient Descent (SGD) feature selection methods were used to evaluate the proposed algorithms against the state-of-the-art; the feature ranking for these algorithms is presented in Table (5). The proposed model outperformed the competition. Tables (5) and Figure (7) both show the results of the experiments. A confusion matrix (CM) was used to assess the classification abilities of our system. The methods used in this study are compared to those discovered in the literature review in Table (5). The results showed that IDS-ML performed the best when comparing ML methods for the feature selection scheme, and it was also the best option when comparing ML methods for the multiclass configuration.

TABLE 5. IDS-ML Algorithm vs. Other Methods for WSN-DS.

Algorithms	Feature Extraction Technique	Accuracy	Precision	Recall	F1-score
ID-GOPA [17]	chis-squared, and information gain ratio	0.96	0.96	0.96	0.96
CNN-LSTM [25]	CNN	0.94	0.95	0.92	NA
RF [24]	Information Gain	0.91	0.90	0.86	0.87
SG-IDS/SGD	PCA10	0.98	0.95	0.96	0.95
SG-IDS/SGD	PCA15	**0.98**	**0.96**	**0.97**	**0.96**
SG-IDS/SGD	SVD10	0.98	0.95	0.96	0.95
SG-IDS/SGD	SVD15	0.98	0.95	0.96	0.95

Moreover, figure (8) shows the ways of selecting features and algorithms compared when it comes to classifying things. Importantly, the Gaussian Naive Bayes (GNB) and Stochastic Gradient Descent (SGD) methods keep performing well. In the case of using PCA and SVD with 10 or 15 parts, it changes how good the predictions are. They affect accuracy, precision, recall, and F-measure score. Specifically, SGD always does better than GNB in every situation. This shows how strong it is. This number gives a quick look at how the system works in different situations. It helps select the best settings for stopping hacker attack problems. The aforementioned research has yielded certain findings, as posited by scholars. SGD has better accuracy and recall than the other approaches on the wireless sensor network dataset WSN-DS, as shown in Tables (4) and (5). The gradient size can be made proportional to the sample size when the SGD is used. It is generally accepted that a more precise model yields a smaller gradient.

The effects of feature selection on the algorithm's accuracy, F-measure, and other metrics are depicted in Figures (4)–(7). The feature selection technique outperformed the other methods in this context. The ability to consider feature dependencies and the connection between searching for feature subsets and selecting a model is crucial when dealing with the WSN-DS dataset. It is easy to eliminate certain unnecessary internally dependent qualities because the other three approaches do not consider the classifier's interaction with the data. However, when the data comprising these characteristics are processed as a whole, the discrimination performance of those features is low, even though the features themselves provide significant potential for discrimination. The wrapper learning algorithm considers the prediction accuracy while weighing the benefits of a potential subset. Classifiers and feature selection work together to allow for the selection of a subset of traits that will prove useful during the learning process.

Compared to existing approaches, such as ID-GOPA [17], Table (5) demonstrates that the proposed SG-IDS algorithm performs better than RF approach [24], and CNN-LSTM [25]. This is because, initially, features are selected based on the traffic data gathered by the sensor nodes. To simplify the traffic characteristics, a technique that

utilizes principal component analysis and singular value decomposition has been developed. The goal is twofold: to reduce the dimensions of traffic data and improve model accuracy. Existing methods for selecting features and classifying intrusion detection systems in wireless sensor networks face several drawbacks, including inadequate detection performance, limited real-time capability, and overly complex models. This approach effectively addressed these issues by handling them separately. This method ensures that the model is not overfitted by incorporating robust detection capabilities and high-quality real-time performances.

VIII. CONCLUSION

Combining feature selection algorithms with machine-learning techniques, current state-of-the-art intrusion detection systems. By reducing the number of features and dimensionality of the model, the feature selection technique helps in generalization and prevents overfitting. Conversely, it can aid in elucidating the connections between traits and their related values. Start by testing a variety of machine learning methods in a controlled environment. In terms of precision, the SG-IDS is the head and shoulder above the competition. Because it is a decision-based learning algorithm within a gradient boosting framework, it is particularly well suited to these tasks owing to its fast-training efficiency, minimal memory use, high accuracy (up to 96 percent), and ability to analyze massive amounts of data. To significantly improve SG-IDS, this study compares many intrusion detection algorithms for WSNs. The high processing cost of an IDS can be avoided by performing thorough feature extraction during data pre-processing to minimize dimensionality and remove redundant data. Therefore, the problem is no longer an issue. Next, SG-IDS was used to improve accuracy and memory. Experiments and research on analogous systems show that this scheme has a high detection rate and low false alarms and requires little calculation. It can identify intruders in wireless sensor networks.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

AUTHOR CONTRIBUTIONS

Hadeel M. Saleh developed the approach, software, and performed the formal analysis; Hend Marouane oversaw the work and provided feedback. Ahmed Fakhfakh's role was supervision.

REFERENCES

[1] E. Tsukerman, "How machine learning is revolutionizing intrusion detection," in *Designing a Machine Learning Intrusion Detection System*. Berkeley, CA, USA: Apress, 2020, doi: 10.1007/978-1-4842-6591-8_2.

[2] N. G. Narkar and N. M. Shekokar, "Evaluation of intrusion detection system with rule-based technique to detect malicious web spiders using machine learning," in *Design of Intelligent Applications Using Machine Learning and Deep Learning Techniques*. Chapman & Hall, 2021, pp. 273–288.

[3] D. Chatterjee, "An efficient intrusion detection system on various datasets using machine learning techniques," in *Machine Learning Techniques and Analytics for Cloud Security*, 2021, pp. 103–128.

[4] J. E. Mulepa and D. G. Selvam, "Proficient intrusion detection system using machine learning using machine learning," *Int. J. Adv. Res. Sci., Commun. Technol.*, pp. 499–506, 2023, doi: 10.48175/ijarsct-9072.

[5] I. Batra, S. Verma, and M. Alazab, "A lightweight IoT-based security framework for inventory automation using wireless sensor network," *Int. J. Commun. Syst.*, vol. 33, no. 4, Mar. 2020, Art. no. e4228, doi: 10.1002/dac.4228.

[6] O. A. Osanaiye, A. S. Alfa, and G. P. Hancke, "Denial of service defence for resource availability in wireless sensor networks," *IEEE Access*, vol. 6, pp. 6975–7004, 2018, doi: 10.1109/ACCESS.2018.2793841.

[7] A. Y. Hussein and A. T. Sadiq, "Meerkat clan-based feature selection in random forest algorithm for IoT intrusion detection," *Iraqi J. Comput., Commun., Control Syst. Eng.*, vol. 22, no. 3, pp. 1–10, 2022, doi: 10.33103/uot.ijccce.22.3.2.

[8] T.-T.-H. Le, T. Park, D. Cho, and H. Kim, "An effective classification for DoS attacks in wireless sensor networks," in *Proc. 10th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2018, pp. 689–692, doi: 10.1109/ICUFN.2018.8436999.

[9] D. Selvamani and V. Selvi, "A comparative study on the feature selection techniques for intrusion detection system," *Asian J. Comput. Sci. Technol.*, vol. 8, no. 1, pp. 42–47, Feb. 2019, doi: 10.51983/ajcst-2019.8.1.2120.

[10] P. Li, W. Zhao, Q. Liu, X. Liu, and L. Yu, "Poisoning machine learning based wireless IDSs via stealing learning model," in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl.*, 2018, pp. 261–273, doi: 10.1007/978-3-319-94268-1_22.

[11] S. K. Pandey, "An anomaly detection technique-based intrusion detection system for wireless sensor network," *Int. J. Wireless Mobile Comput.*, vol. 17, no. 4, p. 323, 2019, doi: 10.1504/IJWMC.2019.103110.

[12] G. Liu, H. Zhao, F. Fan, G. Liu, Q. Xu, and S. Nazir, "An enhanced intrusion detection model based on improved kNN in WSNs," *Sensors*, vol. 22, no. 4, p. 1407, Feb. 2022, doi: 10.3390/s22041407.

[13] P. Michiardi and R. Molva, "Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in in *Proc. 6th Joint Work. Conf. Commun. Multimedia Secur. (IFIP TC6/TC11)*, Portorož, Slovenia. Boston, MA, USA: Springer, Sep. 2002, pp. 107–121.

[14] M. Zhou, Y. Liu, Y. Wang, and Z. Tian, "Anonymous crowdsourcing-based WLAN indoor localization," *Digit. Commun. Netw.*, vol. 5, no. 4, pp. 226–236, Nov. 2019, doi: 10.1016/j.dcan.2019.09.001.

[15] H. Yao, D. Fu, P. Zhang, M. Li, and Y. Liu, "MSML: A novel multilevel semi-supervised machine learning framework for intrusion detection system," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1949–1959, Apr. 2019, doi: 10.1109/JIOT.2018.2873125.

[16] H. Chen, C. Meng, Z. Shan, Z. Fu, and B. K. Bhargava, "A novel low-rate denial of service attack detection approach in ZigBee wireless sensor network by combining Hilbert–Huang transformation and trust evaluation," *IEEE Access*, vol. 7, pp. 32853–32866, 2019, doi: 10.1109/ACCESS.2019.2903816.

[17] S. Ifzarne, H. Tabbaa, I. Hafidi, and N. Lamghari, "Anomaly detection using machine learning techniques in wireless sensor networks," *J. Phys., Conf. Ser.*, vol. 1743, no. 1, Jan. 2021, Art. no. 012021, doi: 10.1088/1742-6596/1743/1/012021.

[18] J. Liu, B. Kantarci, and C. Adams, "Machine learning-driven intrusion detection for Contiki-NG-based IoT networks exposed to NSL-KDD dataset," in *Proc. 2nd ACM Workshop Wireless Secur. Mach. Learn.*, Jul. 2020, pp. 25–30, doi: 10.1145/3395352.3402621.

[19] D. Hemanand, D. S. Jayalakshmi, U. Ghosh, A. Balasundaram, P. Vijayakumar, and P. K. Sharma, "Enabling sustainable energy for smart environment using 5G wireless communication and Internet of Things," *IEEE Wireless Commun.*, vol. 28, no. 6, pp. 56–61, Dec. 2021, doi: 10.1109/MWC.013.2100158.

[20] D. S. Jayalakshmi, D. Hemanand, G. M. Kumar, and M. M. Rani, "An efficient route failure detection mechanism with energy efficient routing (EER) protocol in MANET," *Int. J. Comput. Netw. Inf. Secur.*, vol. 13, no. 2, pp. 16–28, Apr. 2021, doi: 10.5815/IJCNIS.2021.02.02.

[21] 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, doi: 10.1109/ACCESS.2019.2895334.

[22] V. Kumar, D. Sinha, A. K. Das, S. C. Pandey, and R. T. Goswami, "An integrated rule based intrusion detection system: Analysis on UNSW-NB15 data set and the real time online dataset," *Cluster Comput.*, vol. 23, no. 2, pp. 1397–1418, Jun. 2020, doi: 10.1007/s10586-019-03008-x.

[23] P. R. Chandre, P. N. Mahalle, and G. R. Shinde, "Intrusion prevention framework for WSN using Deep CNN," *Turkish J. Comput. Math. Educ. (TURCOMAT)*, vol. 12, no. 6, pp. 3567–3572, 2021.

[24] D. Hemanand, G. Reddy, S. S. Babu, K. R. Balmuri, T. Chitra, and S. Gopalakrishnan, "An intelligent intrusion detection and classification system using CSGO-LSVM model for wireless sensor networks (WSNs)," *Int. J. Intell. Syst. Appl. Eng.*, vol. 10, no. 3, pp. 285–293, Oct. 2022.

[25] O. Almomani, "A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms," *Symmetry*, vol. 12, no. 6, p. 1046, Jun. 2020, doi: 10.3390/sym12061046.

[26] H. Tauqeer, M. M. Iqbal, A. Ali, S. Zaman, and M. U. Chaudhry, "Cyberattacks detection in IoMT using machine learning techniques," *J. Comput. Biomed. Informat.*, vol. 4, no. 1, pp. 13–20, Dec. 2022, doi: 10.56979/401/2022/80.

[27] O. Taouali, S. Bacha, K. Ben Abdellafou, A. Aljuhani, K. Zidi, R. Alanazi, and M. F. Harkat, "Intelligent intrusion detection system for the Internet of Medical Things based on data-driven techniques," *Comput. Syst. Sci. Eng.*, vol. 47, no. 2, pp. 1593–1609, 2023, doi: 10.32604/csse.2023.039984.

[28] L. Dhanya and R. Chitra, "An optimal differential evolution based XGB classifier for IoMT malware classification," in *Proc. Int. Conf. Adv. Intell. Comput. Appl. (AICAPS)*, Feb. 2023, pp. 1–6, doi: 10.1109/aicaps57044.2023.10074030.

[29] S. S. Wali and M. N. Abdullah, "Efficient energy for one node and multi-nodes of wireless body area network," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 12, no. 1, p. 914, Feb. 2022, doi: 10.11591/ijece.v12i1.pp914-923.

[30] K. Cho and Y. Cho, "HyperLedger fabric-based proactive defense against inside attackers in the WSN with trust mechanism," *Electronics*, vol. 9, no. 10, p. 1659, Oct. 2020, doi: 10.3390/electronics9101659.

[31] A. Ghosal and S. Halder, "A survey on energy efficient intrusion detection in wireless sensor networks," *J. Ambient Intell. Smart Environ.*, vol. 9, no. 2, pp. 239–261, Feb. 2017, doi: 10.3233/AIS-170426.

[32] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, "Intrusion detection system for healthcare systems using medical and network data: A comparison study," *IEEE Access*, vol. 8, pp. 106576–106584, 2020, doi: 10.1109/ACCESS.2020.3000421.

[33] N. Jameel and H. S. Abdullah, "Intelligent feature selection methods: A survey," *Eng. Technol. J.*, vol. 39, no. 1B, pp. 175–183, Mar. 2021, doi: 10.30684/etj.v39i1b.1623.

[34] F. J. Ferri, P. Pudil, M. Hatef, and J. Kittler, "Comparative study of techniques for large-scale feature selection," in *Machine Intelligence and Pattern Recognition*, vol. 16. North-Holland, 1994, pp. 403–413.

[35] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014, doi: 10.1016/j.compeleceng.2013.11.024.

[36] K. Guolin, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Y. Liu, "LightGBM: A highly efficient gradient boosting DecisionTree," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 3147–3155.

[37] H. M. Fadhil, M. N. Abdullah, and M. I. Younis, "A framework for predicting airfare prices using machine learning," *Iraqi J. Comput., Commun., Control Syst. Eng.*, vol. 22, no. 3, pp. 1–16, 2022, doi: 10.33103/uot.ijccce.22.3.8.

[38] Q. Li, C. Tai, and E. Weinan, "Stochastic modified equations and dynamics of stochastic gradient algorithms I: Mathematical foundations," *J. Mach. Learn. Res.*, vol. 20, pp. 1474–1520, Jan. 2019.

[39] I. Almomani, B. Al-Kasasbeh, and M. AL-Akhras, "WSN-DS: A dataset for intrusion detection systems in wireless sensor networks," *J. Sensors*, vol. 2016, pp. 1–16, Aug. 2016, doi: 10.1155/2016/4731953.

[40] Q. Liu, D. Wang, Y. Jia, S. Luo, and C. Wang, "A multi-task based deep learning approach for intrusion detection," *Knowl.-Based Syst.*, vol. 238, Feb. 2022, Art. no. 107852, doi: 10.1016/j.knosys.2021.107852.

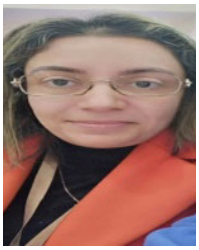
[41] H M. Fadhil, N. Q. Makhool, M. M. Hummady, and Z. O. Dawood, "Machine learning-based information security model for botnet detection," *J. Cybersecurity Inf. Manage.*, vol. 9, no. 1, pp. 68–79, 2022.

[42] I. M. Bapiyev, B. H. Aitchanov, I. A. Tereikovskyi, L. A. Tereikovska, and A. A. Korchenko, "Deep neural networks in cyber attack detection systems," *Int. J. Civil Eng. Technol.*, vol. 8, no. 11, pp. 1086–1092, Nov. 2017.

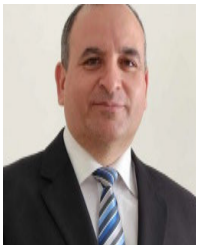
[43] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *J. Inf. Secur. Appl.*, vol. 44, pp. 80–88, Feb. 2019, doi: 10.1016/j.jisa.2018.11.007.

[44] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," *IEEE Access*, vol. 8, pp. 32464–32476, 2020, doi: 10.1109/ACCESS.2020.2973730.

HADEEL M. SALEH is currently a Lecturer with solid academic training in computer science master's degree, with a focus on artificial intelligence. The Presidency of the University has a Center for Continuing Education, where she shares a wealth of knowledge that she has acquired in her professional career. Through her deep knowledge of computer science and information technology, she is at loggerheads with the new crop of professionalism in these fields. She is an important person in her organization and academic circle, due to her educational dedication and zeal for the development of AI.



HEND MAROUANE received the Diploma degree in wireless communication from the Engineering School of Communications (SUP'COM), Tunisia, in 2002, and the Engineering and master's degrees and the Ph.D. degree in engineering from the National School of Engineering of Safax (ENIS), Tunisia, in 2010. She is currently an Assistant Professor with the National School of Electronics and Telecommunication (ENET'COM) of Sfax, Tunisia. Her research interests include wireless networks, advanced protocols for mobile communication, and signal processing. She is a member of the NTS'COM Laboratory, ENET'COM.



AHMED FAKHFAKH received the Engineering degree in electrical engineering from ENIS, in 1997, the D.E.A. and Ph.D. degrees in electronics from Bordeaux University, in 1998 and 2002, respectively, and the H.D.R. Diploma degree in electrical engineering from ENIS, in 2009. From 2002 to 2016, he was a member of the Laboratory of Electronics and Information Technologies (LETI), ENIS. Since 2016, he has been a member of the SM@RTS Laboratory, Digital Research Center of Sfax (CRNS), and the Head of the Research Team "Design and Implementation of Communicating Systems."