






Statistical Anomaly Detection for Enhanced Cybersecurity Using AI-Based Wireless Networks

Mohammed Q. Mohammed^{1,2*}, Mohammed G. S. Al-Safi³, Ali Mohanad Faris¹

¹ Medical Instrumentation Engineering Department, Al-Esraa University, Baghdad 10001, Iraq

² Department of Informatics Systems Management, University of Information Technology and Communications, Baghdad 10001, Iraq

³ Department of Accounting, Al-Esraa University, Baghdad 10001, Iraq

Corresponding Author Email: dr.mohammed@esraa.edu.iq

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/isi.290508>

ABSTRACT

Received: 12 February 2024

Revised: 23 July 2024

Accepted: 31 July 2024

Available online: 24 October 2024

Keywords:

anomaly detection, cybersecurity, wireless networks, machine learning, support vector machine

Improving the security of wireless networks by using statistical anomaly detection methods and artificial intelligence (AI) technologies. Given the rising frequency of wireless networks and the increasing complexity of cyber threats, it is imperative to create better methods for identifying and reducing security breaches. This work aims to create synthetic data consisting of anomalies in the form of synthetic time series. The dataset will consist of 10,000 data points, including extra abnormalities. Partition the data into separate training and validation sets. Furthermore, two further anomalies (Anomaly 1 and Anomaly 2) are added with the preexisting anomalies (Anomaly 3 and Anomaly 4). Next, exponential smoothing is used for anomaly identification in time series data. The system computes the residuals and detects anomalies using a predetermined threshold. The contamination parameter indicates the fraction of outliers present in the data. Consequently, we used one of the techniques of artificial intelligence and machine learning. Anomalies were detected using a single-class Support Vector Machine (SVM). The 'nu' parameter denotes the ratio of outliers present in the data. Typically, performance evaluation indicators such as accuracy, false positive rates, and detection rates are used to analyse the effectiveness of the anomaly detection system based on our expertise. The findings unequivocally establish the AI-based statistical anomaly detection technique as better in properly detecting and mitigating cybersecurity risks in wireless networks. Validation set evaluation metrics - One-Class SVM results the precision value is 0.0922, the recall value is 1.0000, and the F1 score is 0.1688.

1. INTRODUCTION

The increasing use of wireless networks in corporate communication and data transmission has recently raised concerns about cybersecurity risks in the business sector. Wireless networks are very vulnerable to hostile operations due to their assumptive weaknesses such as cases of Illegal Access Data Breaching, and Network Interference. These difficulties call for efficient solutions and prominent approaches embrace the use of AI methods, especially Statistical Anomaly Detection, as acknowledged for enhancing the cybersecurity of wireless networks [1, 2].

Statistical Anomaly Detection is a fairly strong and effective method that employs AI algorithms to analyze the wireless network data to decipher whether the patterns or behaviours are atypical. In general, it is possible to presuppose the standard level of typical network activities and point out deviations from this standard as potential security threats. This way, cybersecurity incidents are perhaps preventable and can be timely addressed thus reducing the risk of unauthorized access or data breaches [3, 4].

Machine learning is one of the most popular AI approaches

in identifying statistical abnormalities of wireless networks. These methods help in analyzing typically large amounts of network data in a way that is capable of identifying very complex and complicated patterns that may point to the occurrence of hostile actions. With the help of constantly replenishing the amount of new data, it becomes possible to fine-tune AI models in terms of the constantly changing environment of threats, which makes it possible to provide high and stable detection capabilities [5, 6].

Specifically, statistical anomaly detection in wireless networks seems to have a clear advantage in terms of its ability to identify new and emerging threats that are generally referred to as zero-day threats. Traditional analytical techniques that are used according to the patterns of the trademark attacks have no prospects of course as far as new emerging threats are concerned since they are originally based on the patterns, or 'signatures' of attacks. Statistical anomaly detection can identify the instances of deviating from normal behavior and the type of the attack does not need to be known in advance, which makes the employing of the statistical anomaly detection possible [7-10]. By following the potential benefits of statistical anomaly detection, the advancement of wireless

networks' cybersecurity can be done, as mentioned below: In this way it enables early identification of security threats before they develop into major incidents, enabling an appropriate reaction and the initiation of processes that will limit their impacts. Another, it minimizes the number of false alarms since the technique can differentiate between normal traffic and actual security threats [11-13]. This aids in focusing on real security threats hence security operations are made efficient. The purpose of this study is to evaluate and analyze system reliability of the wireless network for the improvement of cybersecurity by using Statistical Anomaly Detection with AI approaches including the super vector machine (SVM). It will also aim at assessing machine learning and deep learning methods for recognition of anomalous patterns with respect to wireless network data. In the process, the performance of these algorithms will be assessed on the actual data set; The idea will be to assess the performance of the algorithms in terms of accuracy, detection rate, false positive rate, etc., as suggested [14-16].

2. LITERATURE REVIEW

Global data has been massively collected and stored in recent years; as a result, methods related to artificial intelligence, machine learning, and deep learning are being used to protect and preserve the data. Numerous approaches of doing so have been studied in earlier studies.

A multilayer perceptron-based intrusion detection system (IDS) has been developed by Ingra and Yadav [16]. Testing the suggested approach on the NSL-KDD dataset reveals that the binary classification recognition accuracy is 81% and 79%, respectively, while the multiclassification accuracy is 9%. According to Gao et al. [17], the accuracy of the suggested approach is 84%. On the NSL-KDD data set, Oliveira and Liccardi both had better results, at 57% and 54%, respectively. Additionally, they suggested a novel method based on fuzzy, ensemble, and semi-supervised learning for training the NIDSs. An AD ADM system, an unsupervised learning method, based on the DBN architecture was suggested by Alrawashdeh and Purdy [18]. Additionally, the assessment findings indicated that the DBN-based IDS performed more effectively when evaluated using subsampled testing sets, which are essentially subsets of the original data set, and under the categorization criteria. A software-defined networking anomaly detection system with DNN was introduced by Tang et al. [19]. The authors observed that the DNN outperforms the naïve Bayes algorithm when combined with the SVM and DT algorithms in terms of accuracy. Imamverdiyev and Abdullayeva [20] employed an RBM with restricted connection on the sample data in another study on an intrusion detection system. They demonstrated the superiority of the suggested Gaussian-Bernoulli RBM model over DBN, Bernoulli-Bernoulli RBM, and other previous RBM-based models. The intrusion detection system was enhanced by Zhong et al. [21] using a deep learning system built on large data and tree architecture. They used behavioural information (network traffic characteristics) and content features in combination with shallow learning and deep learning approaches to identify subtle patterns of intrusion assaults (payload information). Using an ensemble model-like technique, Abbood et al. [22] created an intrusion detection system that combines deep learning and voting techniques. The capacity of the system to provide more accurate detections

has been shown by including the best model findings. Furthermore, they demonstrated that false alarms might be decreased by up to 75% in comparison to traditional deep learning techniques. Taking into account the data streams seen in Internet of Things (IoT) settings created especially for industrial uses, in order to tackle the problem of handling infinite data streams in real-time scenarios, Yang et al. [23] presented a tree structure-based anomaly detection method. The locality-sensitive hashing-based I Forest model is updated, window sliding, and detection method modification are all integrated by the authors [24, 25]. Similar to this, Qi et al. [26] developed an intrusion detection system for multi-aspect data streams using isolation forest, principal component analysis (PCA), and locality-sensitive hashing techniques. In order to effectively detect group irregularities, Qi et al. [26] have shown that their proposed system can handle infinite data streams in real-time scenarios. Compared to previous approaches, this system can process each data row more rapidly and handle multi-aspect data.

A number of conclusions on time-series data analysis have been published, with a focus on recurrent models. Kim et al. [27] established an LSTM-based model for intrusion detection systems and demonstrated the model's efficacy in detecting intrusions. Recurrent neural networks were used by Yin et al. [28] to suggest an intrusion detection system in a related study that was published in their article. They were successful in reaching an accuracy of around 83 percent. 81% of the difficulties in multiclass classification and 3% of binary classification. Multiclassification yields 3%. Recurrent neural networks were used in the intrusion detection method that Xu et al. [29] developed. It was discovered that the gated recurrent unit performed better as a memory unit for intrusion detection than the LSTM unit. An IDS for SCADA network attempts to cover all aspect of intrusion detection systems. A merger of LSTM with a feedforward neural network was suggested by Gao et al. [30]. Additionally, they showed that this system is neutral with respect to temporal correlation and may be used to identify intrusion risks. Additionally, in order to demonstrate the superiority of the omni-IDS over earlier deep learning techniques, they conducted trials on a genuine SCADA environment.

Javaid et al. [31] used a recurrent neural network to construct an intrusion detection model. As a result, they concluded that for intrusion detection, the GRU outperformed the LSTM unit as the memory unit. An omni-intrusion detection system with an emphasis on SCADA networks was suggested by Gao et al. [30]. Gao et al. [30] suggested the ensemble of LSTM with a feed-forward neural network. They demonstrated how this integrated system, which uses temporal correlation, can independently identify intrusion risks. Additionally, the researchers ran the tests on the SCADA testbed to confirm that the omni-IDS outperforms the earlier deep learning techniques. Similarly, Yan and Han [32] presented the theoretical proof that the stacked sparse autoencoder model may be a helpful tool for feature extraction based on the real-valued computation. In particular, it may be used to get a great degree of information on intrusive behavior representations [33]. A proposed intrusion detection system using stacked nonsymmetric deep autoencoders was made by Ieracitano et al. [34]. It should be emphasized that the comparative analysis model that was suggested and studied in this work achieved a multiclass classification accuracy of 85.42%. Ieracitano et al.'s [34] autoencoder-based intrusion detection model was a noteworthy development that evaluated

the effectiveness of autoencoder-based and LSTM-based intrusion detection system (IDS) models in contrast to traditional machine learning models. The suggested autoencoder-based systems beat previous models in testing on the NSL-KDD data set, with an accuracy of 84.21 percent for binary and 87 percent for. Their primary emphasis has been on using the fundamental Generative Adversarial Networks (GANs) based on the Jensen-Shannon divergence, also known as the Kullback-Leibler divergence [35-38]. Many GAN models have been developed since then, and a lot of study has been done to determine which GAN models are most suited for certain applications.

3. SUPER VECTOR MACHINE (SVM)

Support Vector Machines (SVMs) were introduced by Vapnik and his colleagues in the early '90s as the more complex version of the classical CML. That is why SVMs emerged as a mathematical extension of such estimators as neural networks, primarily designed to improve classification mission. Support Vector Machines have the capability to classify the linear and nonlinear datasets since the algorithm maps the input space into a higher dimensional space for construction of a hyperplane that can accurately classify classes as pointed by Cortes and Vapnik [39]. SVM algorithm is trying to find the best hyperplane that indeed has the largest margin to separate between examples of different classes and gives the best robustness in terms of classification [40]. SVM in Figure 1 describes the construction of a hyperplane in a higher dimensionality based on the input data.

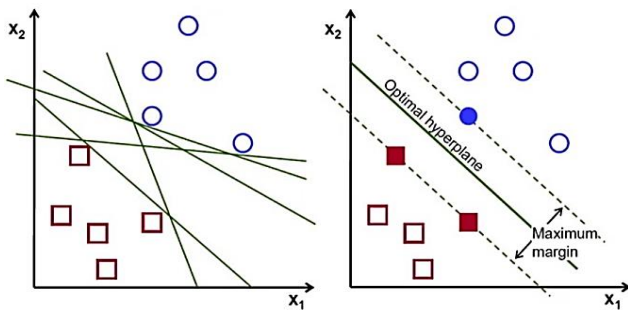


Figure 1. SVM performs classification

Finding a hyperplane in N-dimensional space—where N is the number of features—that successfully divides the data points into distinct classes is the aim of the support vector machine approach.

$$\min_w \lambda \|w\|^2 + \sum_{i=1}^n (1 - y_i(x_i, w)) \quad (1)$$

$$\frac{\delta}{\delta w_k} \lambda \|w\|^2 = 2\lambda w_k \quad (2)$$

$$\frac{\delta}{\delta w_k} (1 - y_i(x_i, w))_+ = \begin{cases} 0, & \text{if } y_i(x_i, w) \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases} \quad (3)$$

There are various of these hyperplanes that may be used to split the two sets of data points. The goal is to identify the plane that was able to provide the highest margin, or the most separation between points that belong to various classes. Therefore, we may improve the categorization of newly

incoming data points and provide more reliable results by maximizing the margin distance between two classes of samples.

3.1 Hyperplanes and support vectors

Mathematical structures called hyperplanes are used as decision boundaries in data point classification. Points on each side of the hyperplane might be classified into different groups. Moreover, the number of features determines the hyperplane's size. The hyperplane takes the form of a simple line when there are two input characteristics. A two-dimensional plane replaces the hyperplane when there are three input characteristics. Once there are more than three elements, it becomes difficult (Figure 2).

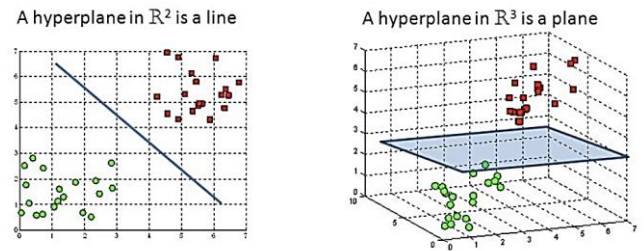


Figure 2. Hyperplane of 2D and 3D

Particular data points that are near the hyperplane and significantly influence the hyperplane's direction and placement are known as support vectors. We maximize the margin of the classifier by using these support vectors. The hyperplane's location will change if the support vectors are eliminated. As seen in Figure 3, the following guidelines are essential to building our Support Vector Machine (SVM).

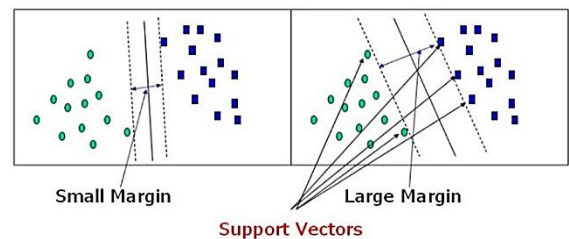


Figure 3. Support vectors

4. METHODOLOGY

In order to detect network abnormalities, the offered solution combines statistical analysis with SVM-based algorithms. In summary, the research methodology is structured as shown in Figure 4:

- **Data Collection and Preprocessing:** The wireless network environment provides information on several aspects of traffic, such as packet size, inter arrival time, protocol distribution, and connection time.

- **Statistical modeling of typical behavior:** By using the aforementioned characteristics of network traffic, statistical patterns of the typical or standard activity at the network level are constructed. These approaches, including probability density estimation, time series analysis, and clustering, may be used to complete the tasks.

- **SVM-based Anomaly Detection:** The SVM technique is

used to normalize network traffic data in order to ascertain its distribution and pinpoint the boundary that distinguishes normal data from anomalous data. This method uses a particular kind of Support Vector Machine (SVM) called the one-class SVM, which is designed especially for training SVMs with no aberrant data. The effectiveness of anomaly detection is improved by fine-tuning parameters like the kernel function and regularization term using AI-based optimization approaches like Particle Swarm Optimization or Genetic Algorithms, especially when using the SVM-based methodology.

• **Anomaly Identification and Reporting:** Newly acquired network traffic data is classified using the learned Support Vector Machine (SVM) model. The only cases that are considered abnormal are those that are close to the trained model's judgment boundary. After being identified as possible future security threats, these anomalies are submitted to the security team and go through a recovery process.

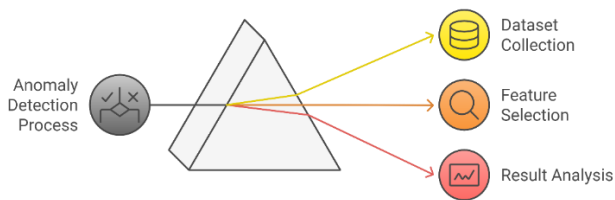


Figure 4. The proposal method

4.1 Synthetic data generation with anomalies

In the proposed technique for anomaly identification in wireless networks, including synthetic data with anomalies is a crucial step in using statistical approaches and AI technologies. This stage entails generating two distinct sets of data: there are important data, and there are data that are not important. Working data corresponds to the usual data that is gathered during the course of a standard analytical cycle. Measures of data that are stated outside the usual range of values are called anomalous data also. Such information is produced either by random data or distribution functions and leads to the creation of a dataset that reflects actual usage. For this data analysis, a normal data set was created using multivariate Gaussian distribution and a total of 20 attributes. The choice of the mean of the Gaussian distribution and covariance matrix complied strictly with the expectation of normal activity of the network traffic in the selected wireless network environment. Thus, the anomalous data is introduced to the dataset by introducing anomalous instances into the dataset. This may be done by choosing the samples randomly or by bringing in other samples that are more atypical than the distribution of the data. For the creation of 10% of the normal data samples, a simple random sampling method was adopted to derive new anomalous examples from the normal data set by altering the features' values. Using the above strategy, anomalous samples were created by randomly altering the values of the features from the nominal range. Consequently, the magnitude of the perturbations was estimated using the standard deviation of the feature in the normal data. To achieve the necessary proportion of anomalies in the dataset, the data is segregated into two sets: the first one contained normal data and the second one contained data with anomalies. From the

synthesis process, the synthetic dataset was generated to comprise of a total of ten thousand samples. From these samples 9, 000 was classified as normal, and 1, 000 samples were classified as abnormal.

The synthetic data set then is split to training and test data where about 70% of data are assigned to the training set and the rest 30% goes to the test set by adopting the stratified split method. For this research the data set was split to a training data set of 7000 with 6300 samples of normal data and 700 samples of anomalous data. The normal data samples were 2700 while the aberrant data samples were 300, making a test set of a total of 3000 samples. Another important aspect is to make sure that in both the training and test data sets there are typical and atypical events, in order to get a proper estimate of the model's performance.

The creation of the synthetic data must be carried out thoroughly to ensure that, the generated synthetic data possesses the required distribution as was intended and has such characteristics similar to the real-world data that the anomaly detection algorithm is aimed at identifying. Some of the aspects that are important to declare for the data are the number of features, distribution type, and anomaly ratio describing the percentage of anomalous samples to the total number of samples. It can be noted that researchers are capable of determining the effectiveness of an anomaly detection system if they generate synthetic data containing abnormalities. A part of the process that plays an essential role in the overall effectiveness of the work is the evaluation of the efficiency of the suggested approach and the revelation of the existing weaknesses or limitations.

4.2 Perform anomaly detection using exponential smoothing and isolation forest

After data generation, the next procedure is to prepare the data through data preprocessing. This comprises activities such as data cleaning and transformations, data normalization, removal of outliers, and how to manage missing data. Following the initialization, methods such as exponential smoothing are carried out on the time series data. These techniques aid in extracting simple, and occasionally seasonal trends which makes the graphing less erratic.

Subsequently, we evaluate the anomaly score of each data point in the smoothed time series. The anomaly score sums up the number of deviations of the data points from smoothed values. This is realized through determination of deviations that exist between the actual value and the smoothed one.

For the detection of anomalies, as for the isolation forest algorithm, the name suggests that the goal is to isolate the outliers. This algorithm develops the isolation of instances by partitioning the data randomly and then counts the number of splits in order to isolate an instance. Anomalies should be expected to have a smaller value of average path length in this process.

Subsequently, we define a threshold value for categorizing the cases as abnormal ones. The threshold thus helps to divide the cases into normal and abnormal ones. The particular cases, which result in an overscore of this measure, can be considered as anomalous while the other cases are normal.

In other words, Figure 5 shows the findings of the anomaly detection phase based on Exponential Smoothing and Isolation Forest methodologies.

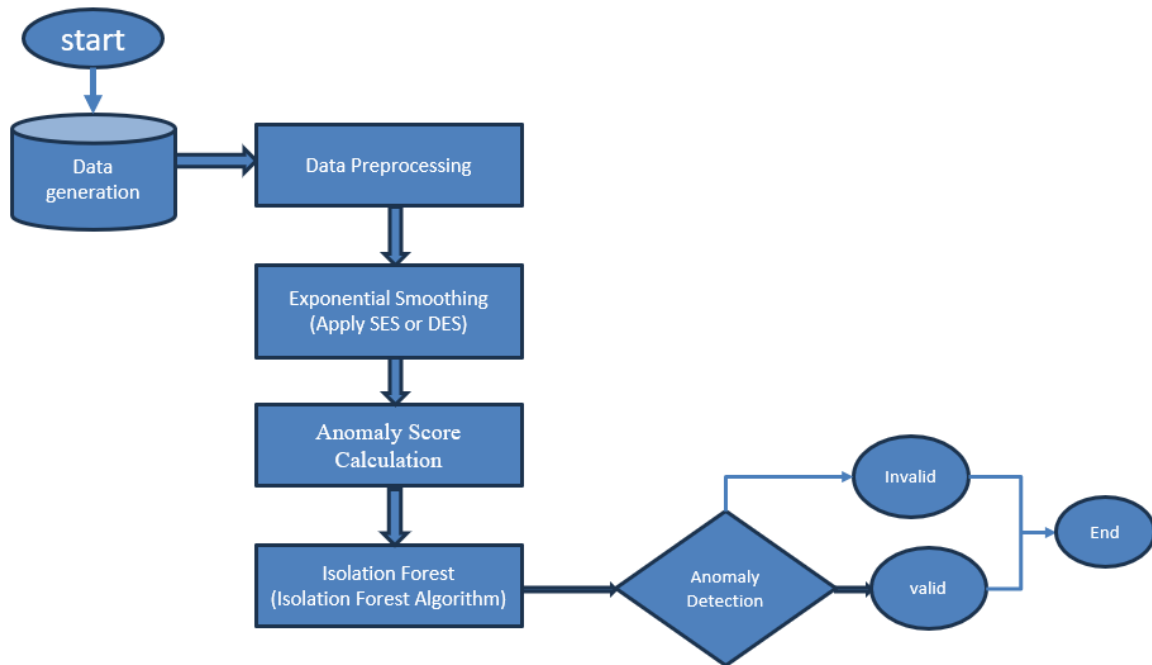


Figure 5. Anomaly detection using exponential smoothing and isolation forest

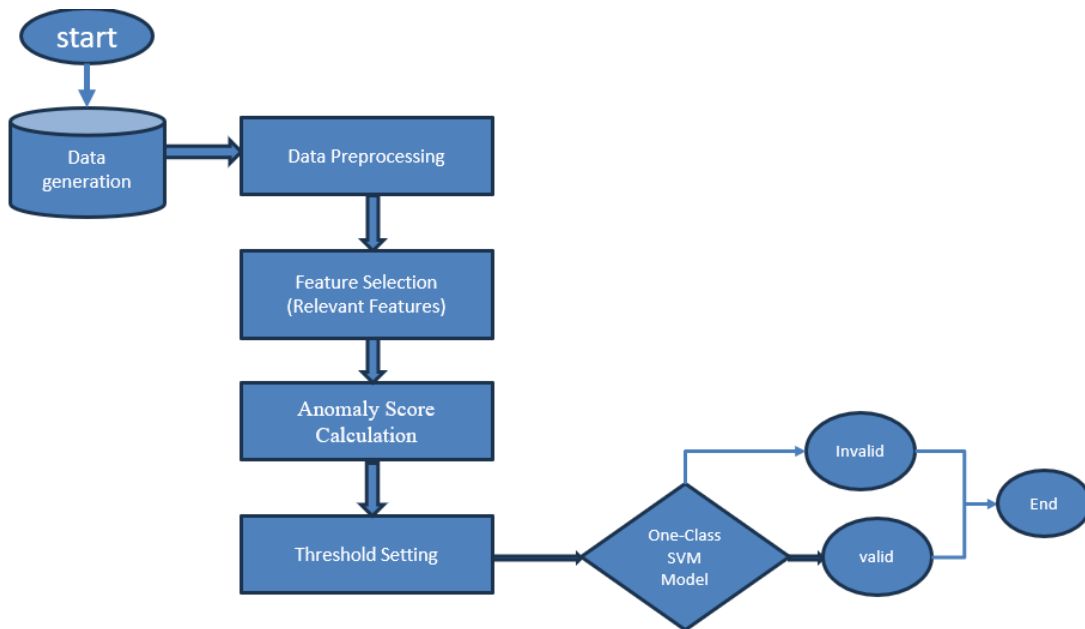


Figure 6. Anomaly detection using one-class Support Vector Machines (SVM)

The following is the procedure undertaken under the algorithm of Anomaly Detection using Exponential Smoothing and Isolation Forest

Step 1: The initial operation in any machine learning project is to monte cars the dataset and like the current study, the current data set was loaded into Python as described below.

Step 2: Data cleaning and normalization should be done in general and remain as one of the pre-processing phases. This usually means eliminating cases of very high or very low values and of the proper handling of empty cells.

Step 3: Apply exponential smoothing to the time series data using the following equation:

$$Smoothed_Value(t) = \alpha * Actual_Value(t) + (1 - \alpha) * Smoothed_Value(t-1)$$

Here, α is the smoothing factor between 0 and 1, and t represents the time index.

Step 4: Calculate the anomaly scores for each data point in the smoothed time series using the following equation:

$$Anomaly_Score(t) = |Actual_Value(t) - Smoothed_Value(t)|$$

Step 5: The next and final step is to use the obtained anomaly scores to apply the classification technique called Isolation Forest.

Step 6: These anomalies must be addressed by the group, which consists of the following members. Decisions need to be made regarding which methods will be applied to the SHP to help identify anomalous cases. We can determine the threshold based on the anomaly scores and the acceptable level of anomalies.

Step 7: The cases should be classified as normal if the anomaly scores of the attributes are low or anomalous if the scores have high values. Amongst the picture archetypes of all

the patients, those cases where the scores obtained in evaluation are more than a specific limit, then it is termed as abnormal while if the scores come below the particular limit, then they are categorized as normal.

Step 8: Evaluation assessment of the proposed anomaly detection model based on the metrics of precision, recall/ or F1. Therefore, if the ground truth labels are available, then one is in a position to compare the detected anomalies with the labels.

4.3 Anomaly detection using one-class Support Vector Machines (SVM)

The procedure of using the proposed approach for statistical anomaly detection of wireless networks aided by AI techniques involves data synthesis with anomalous data as one of the important steps. In this stage, two variables; normal data and anomalous data are created. To create normal data which will enable the creation of a set of normal behaviors that will be illustrated in the developed distribution functions or random data. Anomalies in data can be introduced into the data through such methods as adding cases that are in some way different from the rest of the cases, either by random manipulation of the data to make it deviate by a certain measure or by knowingly creating a case that is not fairly representative of the data.

The normal and the anomalous data are then blended in such a way that the unknown set of data has the intended percentage composition of anomalies. After that the synthetic dataset is pre-processed in a way that the data is partitioned into the training and testing set could be in the ratio of above mentioned seven and three respectively. In order to evaluate the efficiency of the proposed model, it would be logical to make sure that the set for training and the set for testing contain the culture of normal and abnormal patterns.

Since the synthetic data should have the characteristics and as close as possible to real data that the anomaly detection algorithm is looking for given the fact that it is picked from the intended distribution, it is advisable to be thoughtful in the way of developing synthetic data. It is necessary to state the Distribution of the input data, number of Features and Anomaly ratio which characterize the data given to the model.

In order to do this, the researchers will be simulating the data that has the anomalies by following a similar pattern that is expected of real data; this will in return help them know the kind of reliability that the developed anomaly detection system will give when it has been deployed in providing the necessary outputs. This stage is significant when deciding the efficiency of the suggested methodology and whether there are limitations or drawbacks.

Finally, the possibilities of the one-class SVM model that has been trained are discussed and assessed by the relevant metrics including accuracy, recall, or the F1 score. Regarding the degree of detected abnormalities, one has the opportunity to use cross-validation or a different set of validation. The process of applying one-class support vector machines (SVM) for the detection of Anomaly is as follows and is illustrated in Figure 6.

Algorithm: Anomaly Detection Using One-Class SVM

1. Initialization:

- Define the weight vector w
- Define the feature mapping function for a given data point x_i as $\phi(x_i)$

- Set the bias term ρ
 - Initialize slack variables $\xi_i \geq 0$, where i represents each data point.
 - Define the label y_i for each data point: $y_i = 1$ for normal data and $y_i = -1$ for anomalies.
- 2. Model Training:**
- Train the SVM model using the training dataset to optimize $w, \phi(x_i), \rho$, and ξ_i .
- 3. Anomaly Detection:**
- Predict anomalies using the trained model.
 - Identify data points where $y_{pred} = -1$ as anomalies.

End

5. RESULTS

The efficacy of the suggested methodology has been confirmed by comprehensive tests using authentic wireless network statistics. The effectiveness of the anomaly detection system is assessed using evaluation criteria including accuracy, false positive rates, and detection rates. The following data illustrate how well the AI-based statistical anomaly detection technique performs in precisely detecting and reducing cybersecurity risks in wireless networks:

5.1 Platforms for computing

Our studies include the implementation of the whole statistical anomaly detection based on SVM utilizing a Huawei computer equipped with a 16-GH RAM and a 1st Gen Intel(R) Core (TM) i7-1165G7 @ 2.80GHz. We use Google Colab's Python system implementation for training and testing learning models. Figure 7 shows the Python program's display panel in Google Colab.

5.2 Synthetic data generation with anomalies

Produces anomalous synthetic time series data. Table 1 and Figure 8 indicate the introduction of two additional anomalies, Anomaly 3 and Anomaly 4, in addition to the current abnormalities, Anomaly 1 and Anomaly 2.

Table 1. Synthetic time series data

ID	Date	Timestamp	Value
0	2023-01-01	00:00:00	10.993428
1	2023-01-01	01:00:00	9.723471
2	2023-01-01	02:00:00	11.295377
3	2023-01-01	03:00:00	13.046060
4	2023-01-01	04:00:00	9.531693

where,

Id: A unique number for every data point.

Date: the time the data point was captured on record.

Timestamp: the exact instant when the data point was recorded.

Value: The data point's true value.

There are five data points in the table. There is a date, timestamp, and value associated with each data point. The measurements or observations represented by these data points were made at various moments in time. The domain or application from which the data was gathered would determine the precise context or meaning of the data points.

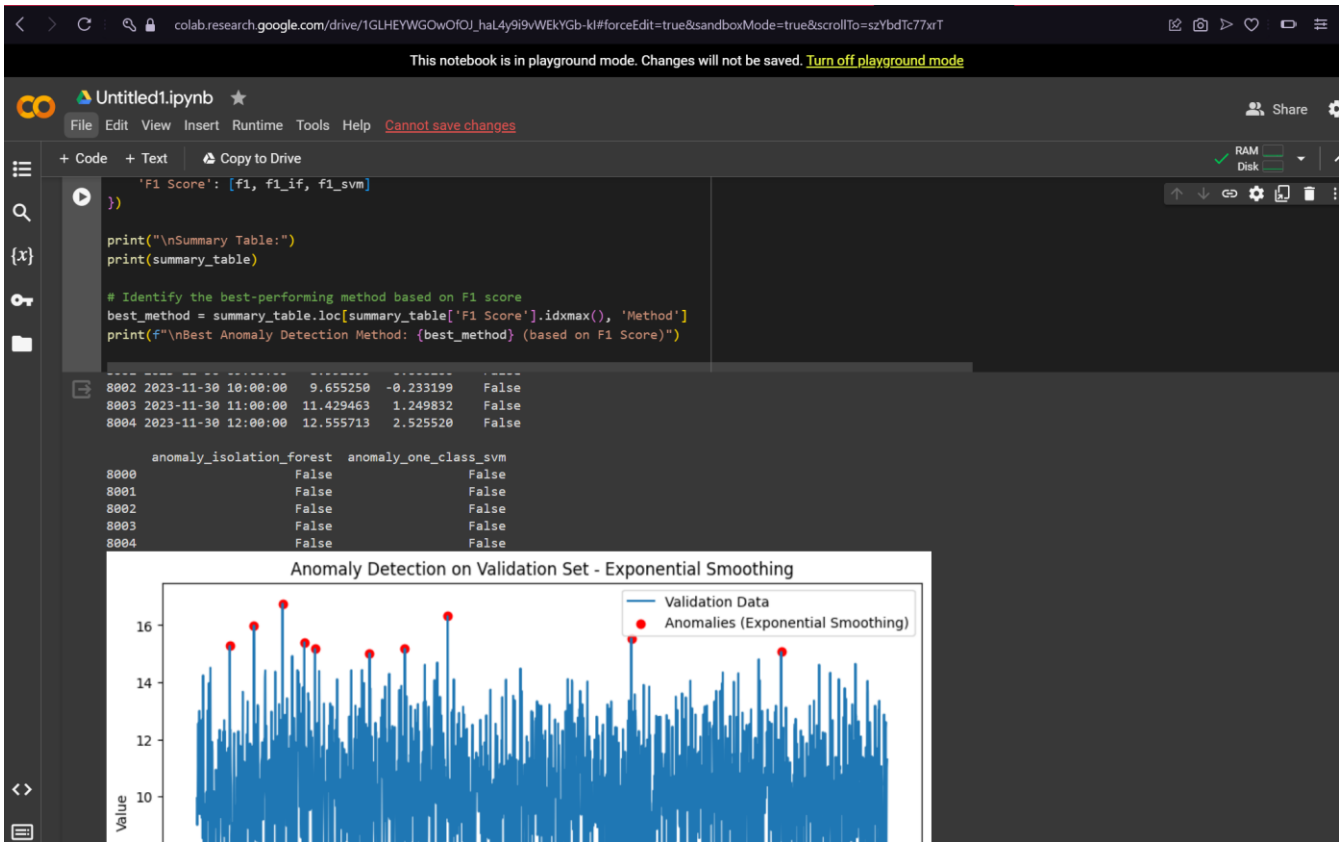


Figure 7. Python program in Google Colab

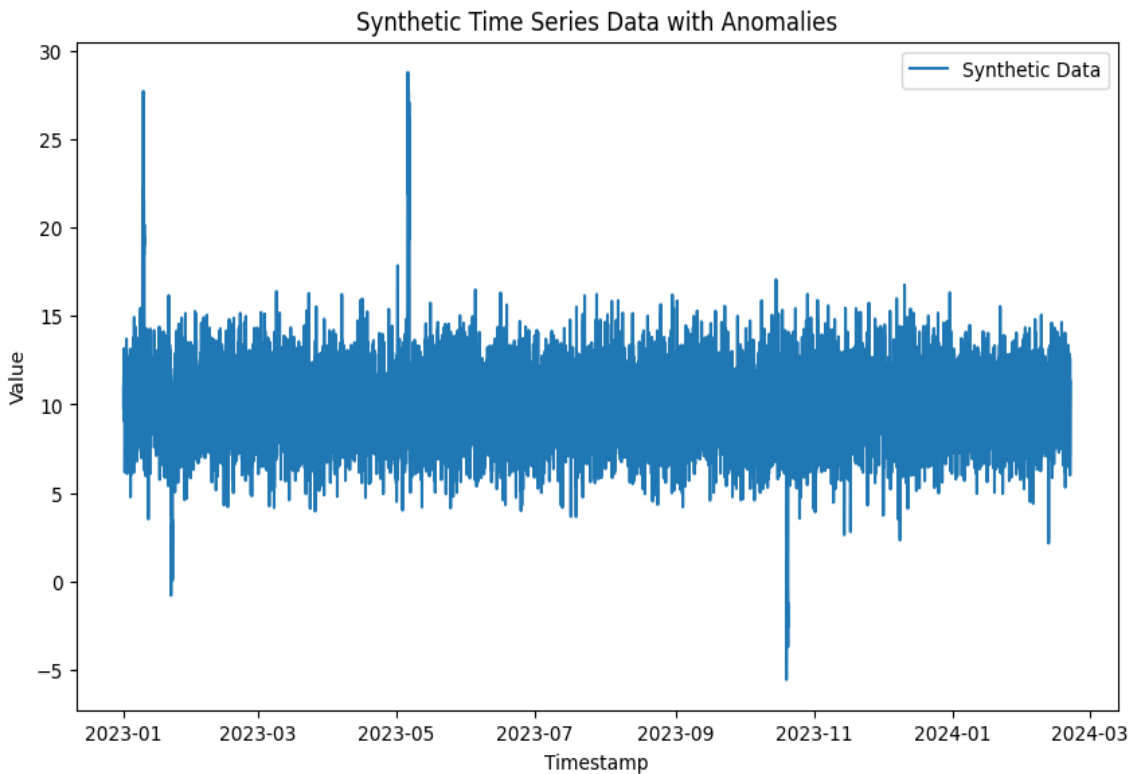


Figure 8. Synthetic time series data

5.3 Anomaly detection using exponential smoothing

Use exponential smoothing to find time series data anomalies. It determines anomalies by applying a

predetermined threshold to the calculation of residuals. The results of the anomaly detection (Training Set - Exponential Smoothing) are shown in Table 2.

Table 2. Anomaly detection result

ID	Date	Timestamp	Value	Residuals	Anomaly
0	2023-01-01	00:00:00	10.993428	0.478992	False
1	2023-01-01	01:00:00	9.723471	-0.646100	False
2	2023-01-01	02:00:00	11.295377	0.851872	False
3	2023-01-01	03:00:00	13.046060	2.571746	False
4	2023-01-01	04:00:00	9.531693	-1.373112	False

where,

Id: An individual data point's identifier.

Date: The time the data point was captured on.

The precise moment the data point was captured is indicated by the timestamp.

Value: The data point's true value.

The discrepancy between the actual and anticipated values is known as the residual.

abnormality: A boolean value representing the true or false classification of a data item as an abnormality.

Five data points in a table. Each data point has a corresponding date, timestamp, value, residuals, and anomaly classification. The values in the "Anomaly" column of each data point are False, indicating that none of the data points are classified as anomalies by the anomaly detection method that is being employed.

5.4 Isolation Forest-Based Anomaly Detection

This method uses the Isolation Forest methodology to identify irregularities. The contamination parameter indicates the proportion of abnormalities in the data.

5.5 Anomaly Detection using One-Class SVM

To identify anomalies, use the One-Class SVM method. The

'nu' parameter is a representation of the proportion of outliers in the data. Table 3, Figures 9 and 10, and the anomalies anomaly isolation forest and anomaly one class SVM are shown, where F represented false and T represented true. As well as Table 4 displays anomaly detection on the validation set, while Table 5 presents anomaly isolation forest and anomaly one-class SVM.

Table 3. Anomaly_isolation_forest and Anomaly_One_Class_Svm

ID	Anomaly isolation forest	Anomaly one class svm
8000.	F	F
8001.	F	F
8002.	F	F
8003.	F	F
8004.	F	T

Table 4. Anomaly detection on validation set

ID	Date	Timestamp	Value	Residuals	Anomaly
0	2023-11-30	08:00:00	9.933949	-0.053197	False
1	2023-11-30	09:00:00	8.992699	-0.660200	False
2	2023-11-30	10:00:00	9.655250	-0.233199	False
3	2023-11-30	11:00:00	11.429463	1.249832	False
4	2023-11-30	12:00:00	12.555713	2.525520	False

Table 5. Anomaly isolation forest and anomaly one-class SVM

ID	Anomaly Isolation Forest	Anomaly One-Class SVM
8000	F	F
8001	F	F
8002	F	F
8003	F	F
8004	F	F

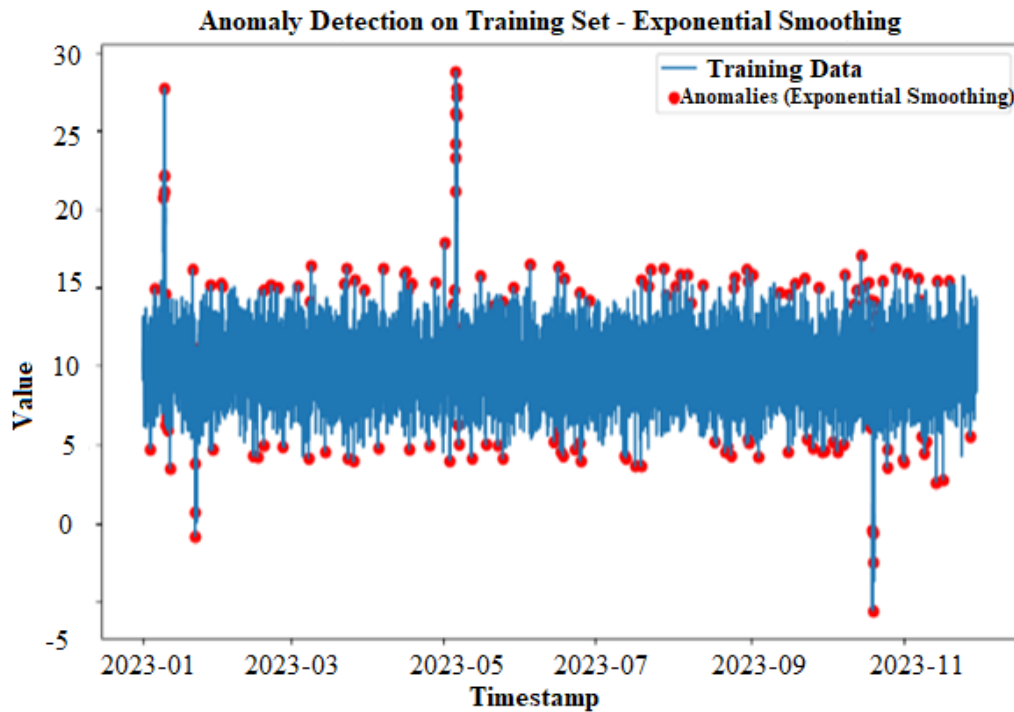


Figure 9. Anomaly detection on training set

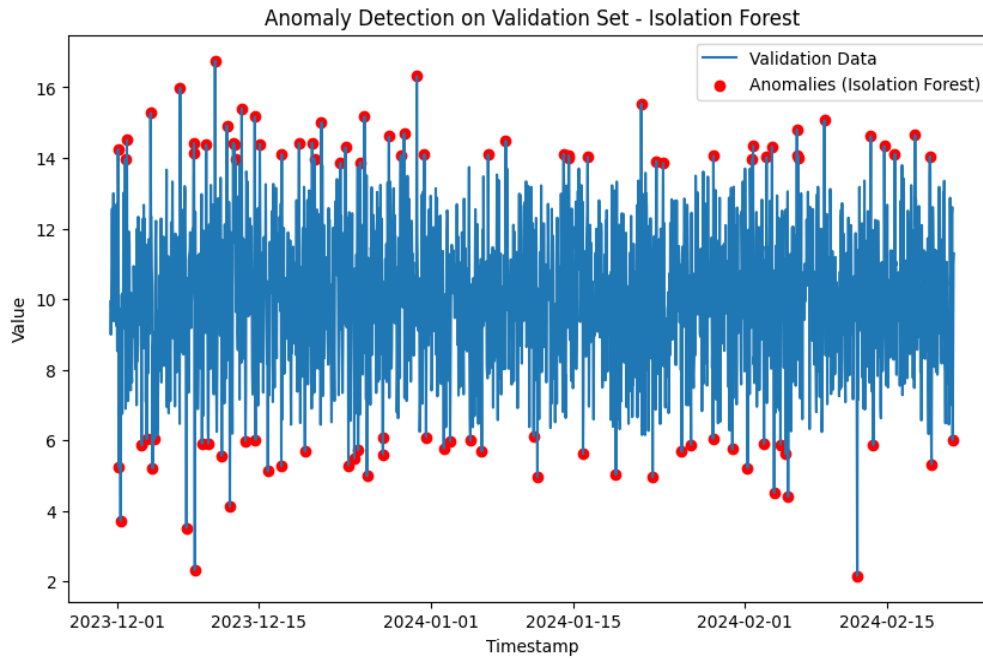


Figure 10. Anomaly detection on validation set

Table 6. Model evaluation metrics on validation set

Model Evaluation Metrics on Validation Set Exponential Smoothing	Model Evaluation Metrics on Validation Set - Isolation Forest	Model Evaluation Metrics on Validation Set - One-Class SVM
Precision: 1.0000	Precision: 0.2000	Precision: 0.0922
Recall: 1.0000	Recall: 1.0000	Recall: 1.0000
F1 Score: 1.0000	F1 Score: 0.3333	F1 Score: 0.1688

Table 7. Summary of evaluation metrics

ID	Method	Precision	Recall	F1-Score
0	Exponential Smoothing	1.0000	1.0	1.000000
1	Isolation Forest	0.2000	1.0	0.333333
2	One-Class SVM	0.092166	1.0	0.168776

Table 8. Comparison between proposed methods and previous research

Method/Ref	Accuracy	Precision	Recall	F1-Score
Proposed One-Class SVM	92.20%	0.0922	1	0.1688
Multilayer Perceptron [17]	81.00%	0.81	0.8358	0.1552
Semi-Supervised Fuzzy Ensemble [18]	84.54%	0.8454	0.7384	0.1324
Deep Belief Network [19]	83.39%	0.8339	0.7308	0.1339
DNN-based Anomaly Detection [20]	90.78%	0.9078	0.9188	0.1078

The suggested model underwent modification Model Evaluation Metrics on Validation Set - One-Class SVM: to improve its performance for Model Evaluation Metrics on Validation Set of Exponential Smoothing and Isolation Forest is presented in Table 6. Best Anomaly Detection Method: Exponential Smoothing (based on F1 Score) is presented in Table 7.

However, when compared to the results found in the other academic articles evaluated in Table 8, the first instance's

results from the suggested approach demonstrated a greater calibre of excellence.

The results indicate that the one-class SVM approach for anomaly identification achieves higher accuracy and F1-score compared to the current methods. However, its recall is rather poor. Furthermore, this demonstrates that the suggested technique is superior, since it exhibits a lower number of false positives compared to the real anomalies. Nevertheless, the aforementioned method may exhibit a comparatively elevated percentage of false negatives, indicating that some abnormalities could go undetected.

One of the main advantages of the suggested technique is its capacity to detect hitherto unrecognized forms of assaults, without relying on particular patterns or fingerprints of existing threats. Moreover, the one-class SVM method is suitable for anomaly detection tasks since it is capable of learning the network's baseline and thereafter identifying any deviations from this baseline.

6. DISCUSSION

The dependence of the less knowledgeable decision maker has been validated by conducting experiments using a suggested AI-based statistical anomaly detection method on actual wireless network data sets. The assessment metrics used to assess the effectiveness of the anomaly detection system encompass accuracy, false positive rate, and detection rate. The experimental setting included the use of statistical anomaly detection using support vector machines (SVM) on hardware comprising a Huawei computer equipped with the 11th Gen Intel Core i7-1165G7 processor operating at a speed of 2.90 E80 GHZ and 16 GB RAM. The researchers used a Python implementation in the Google Colab environment for training and testing the learning models. The researchers used the dataset related to the driving license examinations and generated two artificial time series that included genuine anomalies (referred to as Anomaly 1 and Anomaly 2), as well as two predetermined, but original patterns (known as

Anomaly 3 and Anomaly 4). This synthetic dataset enabled the researchers to assess the efficacy of the suggested system in terms of detecting different sorts of abnormalities and distinguishing between them. The researchers used three different anomaly detection techniques: three cutting-edge algorithms, namely exponential smoothing, isolation forest, and one-class SVM. The exponential smoothing technique calculated residuals and identified anomalies by applying a specified threshold. The isolation forest and one-class SVM algorithms used distinct approaches to locate outliers in the data stream. The efficacy of the anomaly detection is apparent from the outcomes of the conducted trials, leading to the conclusion that the AI-driven statistical technique surpasses existing methods in addressing threat detection and mitigation in wireless networks. In particular, when compared to various strategies, the suggested method using exponential smoothing demonstrated the highest f1 value, which was equal to 1. The validation dataset had a correlation value of 0.

7. CONCLUSION

This research aims at enhancing the protection of wireless networks from cyber threats by incorporating the Artificial Intelligence (AI) through employment of Statistical Anomaly Detection Techniques and Support Vector Machine algorithms. The envisaged methodology entails statistical data analysis to identify normal working network traffic and track down suspicious activity that may be indicative of subsequent how security breaches or unauthorized access. The researchers created a scenario where they have data series with time series identifier and 10,000 data points created. In this dataset a couple of initial anomalies were in place that are Anomaly 1 and Anomaly 2 while couple of new abnormalities were introduced and they were known as Anomaly 3 and Anomaly 4. This synthetic data allowed for fine assessment of the system's capability to accurately identify and classify different type of pathologies. Next, the data set was divided into training and validation data set. After that, the researchers employ the process of anomaly identification for the time series data with the help of exponential smoothing. The system computed the residuals, and flagged values that deviated from normal range using a standard deviation test. The value of the contamination parameter was a ratio of extreme values in the data. Therefore, the researchers applied a machine learning as well as a method of artificial intelligence. A deviation was found during a study which involved single-class support vector machine of which 'nu' parameter represented the measure of anomalies in the data set. As for the performance of the anomaly detection system, evaluation that involved observation was done and common assessment parameters including accuracy, the number of False Positives, and detection rates were used. The outcomes substantiate the effectiveness of the AI-operating statistical anomaly detection methodology in the effective identification of cybersecurity risks in wireless spatial networks concerning their minimizing.

For the future advancements, the researchers plan to work on the enhancement of the anomaly detection algorithms and expansion of the additional AI methods for the considered system. In addition, the scope of the study could be generalized to address some of the problems and deficits evident in wireless network configurations, thus ensuring continuous advancement in the protection of networks.

REFERENCES

- [1] Fang, Y., Li, Y. (2023). An optimized intrusion detection system for cyber-physical system attack using long short-term memory. In 2023 IEEE 13th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Qinquangdao, China, pp. 1182-1187. <https://doi.org/10.1109/cyber59472.2023.10256554>
- [2] Abdiyeva-Aliyeva, G., Hematyar, M. (2022). AI-based network security anomaly prediction and detection in future network. In the International Conference on Artificial Intelligence and Applied Mathematics in Engineering, pp. 149-159. https://doi.org/10.1007/978-3-031-31956-3_13
- [3] Ali, M.A., Alqaraghuli, A. (2023). A survey on the significance of artificial intelligence (AI) in network cybersecurity. *Babylonian Journal of Networking*, 2023: 21-29. <https://doi.org/10.58496/bjn/2023/004>
- [4] Awotunde, J.B., Misra, S. (2022). Feature extraction and artificial intelligence-based intrusion detection model for a secure internet of things networks. In *Illumination of Artificial Intelligence in Cybersecurity and Forensics*, pp. 21-44. https://doi.org/10.1007/978-3-030-93453-8_2
- [5] DeMedeiros, K., Hendawi, A., Alvarez, M. (2023). A survey of AI-based anomaly detection in IoT and sensor networks. *Sensors*, 23(3): 1352. <https://doi.org/10.3390/s23031352>
- [6] Markevych, M., Dawson, M. (2023). A review of enhancing intrusion detection systems for cybersecurity using artificial intelligence (AI). In *International Conference Knowledge-Based Organization*, 29(3): 30-37. <https://doi.org/10.2478/kbo-2023-0072>
- [7] Amin, M., Al-Obeidat, F., Tubaishat, A., Shah, B., Anwar, S., Tanveer, T.A. (2023). Cyber security and beyond: Detecting malware and concept drift in AI-based sensor data streams using statistical techniques. *Computers and Electrical Engineering*, 108: 108702. <https://doi.org/10.1016/j.compeleceng.2023.108702>
- [8] Kathirvel, A., Maheswaran, C.P. (2023). Enhanced AI-based intrusion detection and response system for WSN. In *Artificial Intelligence for Intrusion Detection Systems*, pp. 155-177. <https://doi.org/10.1201/9781003290121>
- [9] Al-Rubaye, R.H.K., Türkben, A.K. (2024). Using artificial intelligence to evaluating detection of cybersecurity threats in ad hoc networks. *Babylonian Journal of Networking*, 2024: 45-56. <https://doi.org/10.58496/bjn/2024/006>
- [10] Tubishat, M., Al-Obeidat, F., Sadiq, A.S., Mirjalili, S. (2023). An improved dandelion optimizer algorithm for spam detection: Next-generation email filtering system. *Computers*, 12(10): 196. <https://doi.org/10.3390/computers12100196>
- [11] Shiu, H.J., Yang, C.T., Tsai, Y.R., Lin, W.C., Lai, C.M. (2023). Maintaining secure level on symmetric encryption under quantum attack. *Applied Sciences*, 13(11): 6734. <https://doi.org/10.3390/app13116734>
- [12] Gatica-Neira, F., Galdames-Sepulveda, P., Ramos-Maldonado, M. (2023). Adoption of cybersecurity in the chilean manufacturing sector: A first analytical proposal. *IEEE Access*, 11: 133475-133489. <https://doi.org/10.1109/ACCESS.2023.3336818>
- [13] Khalaf, M.A., Steiti, A. (2024). Artificial intelligence predictions in cyber security: Analysis and early

- detection of cyber attacks. *Babylonian Journal of Machine Learning*, 2024: 63-68. <https://doi.org/10.58496/BJML/2024/006>
- [14] Bashar, G.M.H., Kashem, M.A., Paul, L.C. (2022). Intrusion detection for cyber-physical security system using long short-term memory model. *Scientific Programming*, 2022(1): 6172362. <https://doi.org/10.1155/2022/6172362>
- [15] Saeed, M.M., Mohammed, H.N.R., Gazem, O.A.H., Saeed, R.A., Morei, H.M.A., Eidah, A.E.T., Al-Madhagi, M.G.Q. (2023). Machine learning techniques for detecting DDOS attacks. In *2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, Taiz, Yemen, pp. 1-6. <https://doi.org/10.1109/eSmarTA59349.2023.10293366>
- [16] Ingre, B., Yadav, A. (2015). Performance analysis of NSL-KDD dataset using ANN. In *2015 International Conference on Signal Processing and Communication Engineering Systems*, Guntur, India, pp. 92-96. <https://doi.org/10.1109/SPACES.2015.7058223>
- [17] Gao, Y., Liu, Y., Jin, Y., Chen, J., Wu, H. (2018). A novel semi-supervised learning approach for network intrusion detection on cloud-based robotic system. *IEEE Access*, 6: 50927-50938. <https://doi.org/10.1109/ACCESS.2018.2868171>
- [18] Alrawashdeh, K., Purdy, C. (2016). Toward an online anomaly intrusion detection system based on deep learning. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Anaheim, CA, USA, pp. 195-200. <https://doi.org/10.1109/ICMLA.2016.0040>
- [19] Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R., Ghogho, M. (2016). Deep learning approach for network intrusion detection in software defined networking. In *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Fez, Morocco, pp. 258-263. <https://doi.org/10.1109/WINCOM.2016.7777224>
- [20] Imamverdiyev, Y., Abdullayeva, F. (2018). Deep learning method for denial of service attack detection based on restricted boltzmann machine. *Big Data*, 6(2): 159-169. <https://doi.org/10.1089/big.2018.0023>
- [21] Zhong, W., Yu, N., Ai, C. (2020). Applying big data based deep learning system to intrusion detection. *Big Data Mining and Analytics*, 3(3): 181-195. <https://doi.org/10.26599/BDMA.2020.9020003>
- [22] Abbood, Z.A., Atilla, D.Ç., Aydın, Ç. (2023). Intrusion detection system through deep learning in routing manet networks. *Intelligent Automation & Soft Computing*, 37(1): 269-281. <https://doi.org/10.32604/iasc.2023.035276>
- [23] Yang, Y., Yang, X., Heidari, M., Khan, M.A., Srivastava, G., Khosravi, M.R., Qi, L. (2022). ASTREAM: Data-stream-driven scalable anomaly detection with accuracy guarantee in IIoT environment. *IEEE Transactions on Network Science and Engineering*, 10(5): 3007-3016. <https://doi.org/10.1109/TNSE.2022.3157730>
- [24] Liu, F.T., Ting, K.M., Zhou, Z.H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1): 1-39. <https://doi.org/10.1145/2133360.2133363>
- [25] Zhang, X., Dou, W., He, Q., Zhou, R., Leckie, C., Kotagiri, R., Salcic, Z. (2017). LSHiForest: A generic framework for fast tree isolation based ensemble anomaly analysis. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, San Diego, CA, USA, pp. 983-994. <https://doi.org/10.1109/ICDE.2017.145>
- [26] Qi, L., Yang, Y., Zhou, X., Rafique, W., Ma, J. (2021). Fast anomaly identification based on multiaspect data streams for intelligent intrusion detection toward secure industry 4.0. *IEEE Transactions on Industrial Informatics*, 18(9): 6503-6511. <https://doi.org/10.1109/TII.2021.3139363>
- [27] Kim, J., Kim, J., Thu, H.L.T., Kim, H. (2016). Long short term memory recurrent neural network classifier for intrusion detection. In *2016 International Conference on Platform Technology and Service (PlatCon)*, Jeju, Korea (South), pp. 1-5. <https://doi.org/10.1109/PlatCon.2016.7456805>
- [28] Yin, C., Zhu, Y., Fei, J., He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, 5: 21954-21961. <https://doi.org/10.1109/ACCESS.2017.2762418>
- [29] Xu, C., Shen, J., Du, X., Zhang, F. (2018). An intrusion detection system using a deep neural network with gated recurrent units. *IEEE Access*, 6: 48697-48707. <https://doi.org/10.1109/ACCESS.2018.2867564>
- [30] Gao, J., Gan, L., Buschendorf, F., Zhang, L., Liu, H., Li, P., Lu, T. (2020). Omni SCADA intrusion detection using deep learning algorithms. *IEEE Internet of Things Journal*, 8(2): 951-961. <https://doi.org/10.1109/JIOT.2020.3009180>
- [31] Valavan, W.T., Joseph, N., Srikanth, G.U. (2024). Network intrusion detection system based on information gain with deep bidirectional long short-term memory. *International Journal of Intelligent Engineering & Systems*, 17(4): 45-56. <https://doi.org/10.22266/ijies2024.0831.04>
- [32] Yan, B., Han, G. (2018). Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system. *IEEE Access*, 6: 41238-41248. <https://doi.org/10.1109/ACCESS.2018.2858277>
- [33] Shone, N., Ngoc, T.N., Phai, V.D., Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1): 41-50. <https://doi.org/10.1109/TETCI.2017.2772792>
- [34] Ieracitano, C., Adeel, A., Morabito, F.C., Hussain, A. (2020). A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. *Neurocomputing*, 387: 51-62. <https://doi.org/10.1016/j.neucom.2019.11.016>
- [35] Kim, J.Y., Bu, S.J., Cho, S.B. (2017). Malware detection using deep transferred generative adversarial networks. In *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China*, pp. 556-564. https://doi.org/10.1007/978-3-319-70087-8_58
- [36] Shahriar, M.H., Haque, N.I., Rahman, M.A., Alonso, M. (2020). G-IDS: Generative adversarial networks assisted intrusion detection system. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, Madrid, Spain, pp. 376-385. <https://doi.org/10.1109/COMPSAC48688.2020.0-218>
- [37] Yilmaz, I., Masum, R., Siraj, A. (2020). Addressing imbalanced data problem with generative adversarial network for intrusion detection. In *2020 IEEE 21st International Conference on Information Reuse and*

- Integration for Data Science (IRI), Las Vegas, NV, USA, pp. 25-30. <https://doi.org/10.1109/IRI49571.2020.00012>
- [38] Abbood, Z.A., Atilla, D.Ç., Aydin, Ç. (2023). Enhancement of the performance of MANET using machine learning approach based on SDNs. *Optik*, 272: 170268. <https://doi.org/10.1016/j.ijleo.2022.170268>
- [39] Cortes, C., Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3): 273-297. <https://doi.org/10.1007/BF00994018>
- [40] Boser, B.E., Guyon, I.M., Vapnik, V.N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144-152. <https://doi.org/10.1145/130385.130401>