Mesopotamian journal of Cybersecurity
Vol.5,No.1, **pp**. 202–215
DOI: https://doi.org/10.58496/MJCS/2025/014; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

Enhancing Cybersecurity with Machine Learning: A Hybrid Approach for Anomaly Detection and Threat Prediction

Adil M. Salman[1], Bashar Talib Al-Nuaimi[2], Alhumaima Ali Subhi[3], Hussein Alkattan[4, 5,*], Raed H. C. Alfilh[6]

[1] *Computer Science Department, Baghdad College of Economic Sciences, Iraq, Baghdad.*

[2] *Computer Science Department, University of Diyala, Diyala 32001, Iraq.*

[3] *Electronic Computer Centre, University of Diyala, Diyala, Iraq.*

[4] *Department of System Programming, South Ural State University, Chelyabinsk, Russia.*

[5] *Directorate of Environment in Najaf, Ministry of Environment, Najaf, Iraq*

[6] *Refrigeration & Air-Conditioning Technical Engineering Department, College of Technical Engineering, The Islamic University, Najaf, Iraq.*

ARTICLEINFO

Article history
Received 02 Jan 2025
Accepted 03 Mar 2025
Published 25 Mar 2025

Keywords
Hybrid Model
Machine Learning
Anomaly detection
Cybersecurity
Threat Prediction
Time series analysis

ABSTRACT

In today's digital era, cybersecurity has become a principal concern because of the increasing frequency and advancement of cyber threats. This study explores machine learning models for detecting and predicting anomalies in cybersecurity datasets. The research evaluates models such as linear regression, decision tree, RF, gradient boosting, KNN, SVR, LSTM, and neural networks utilizing performance metrics such as accuracy, MAE and MSE. A hybrid model that integrates different learning strategies is additionally proposed to improve the predictive accuracy and strength. The results highlight the superiority of ensemble approaches, especially the hybrid model, in improving peculiarity detection capabilities. The comparative analysis demonstrates that traditional models struggle with nonlinear patterns, whereas hybrid approaches successfully relieve this limitation. Moreover, this study emphasizes the importance of temporal data analysis for proactive threat detection and response. By leveraging diverse machine learning methods, this research contributes to strengthening cybersecurity infrastructures, enabling early threat detection, and minimizing security breaches. These discoveries emphasize the importance of adopting a comprehensive machine learning system to support cybersecurity resilience.

1. INTRODUCTION

In our current period, technology has become a fundamental part of our daily lives, entering all areas that require individual work. In light of this advanced development, numerous fast challenges are utilized due to numerous challenges within the field of cybersecurity, where security dangers are continually expanding in an undesirable way. These dangers are not only coordinated with people but also include enormous places and piles, which undermines digital neuroscience in common. Owing to this otherworldliness, it has become necessary to search for successful methods for all types and predict them, as well as to protect many people [1][2]. Among the tools and procedures that work within the field of smart cyber devices are technological learning strategies, which are mostly composed of artificial intelligence [3]. Enchanted learning could be a subfield of artificial intelligence that enables systems to learn from data and start without being required for old programming [4]. Over the years, many scientific models have been utilized within the field of cybernetics, such as linear regression [5], SVM [6], random trees [7], neural networks [8], independent trees [9], and other discoveries based on the analysis of new data and temporal data [10].

The time period (time series data) is one of the most utilized types of information within the analysis of data that talks over time [11]. In cyber, your data are the anomalous exercises of specialized networks within the legislative or official authority [12]. By applying magic information techniques to these data, threat prediction frameworks can predict the final result [13]. In any case, although this field is exceptionally promising, many challenges confront learning magic applications in this

*Corresponding author. Email: alkattan.hussein92@gmail.com

field, including its differing qualities [14], the nearness of noise within the data [15], and the importance of speed of detection and response [16].

Given the various studies in this area, different methods, such as linear regression [5], neural networks [8], SVM [6], RF [7], and autonomous trees (decision trees) [9], may lead to the capacity to belittle prior to some time. In any case, these correlations vary in terms of increasing differences with the differences in information, as a few relationships, such as arbitrary alliances and neural networks, may be more exact in complex or high-resolution data, whereas models such as linear regression can be more precise and quicker to execute but may struggle to handle nonlinear connections between investments [17] [18]. In this context, hybrid results or the merging of more than one machine learning model may be a compelling way to improve performance and increase the accuracy of predictions. Merging correlations can provide a central solution between high execution and exactness and can address diverse factors in the data [19] [20]. The utilization of temporal data in these advancements requires progressed techniques, such as worldly developments (temporal models), that offer assistance in predicting the future on the basis of historical data [21] [2].

Digital checking with the prediction of cyber occasions is one of the foremost important ranges in this field. These relationships are expected to be calculators for planning at extraordinary limits, allowing security agencies to respond quickly to computer and electronics harm [23]. However, it is completely within the capacity to apply these studies in commonsense advancement, which makes research on this topic highly important for protecting social security[24][62].

Electronics that depend on intelligent learning to detect dozens of cyber advancements are expected to be created significantly, and these electronics are expected to attract interest in an essential portion of the future cyber differing qualities [25]. Modern technologies can learn to increase the capacity of organizations to monitor their networks and detect any abnormal movement speedier and more successfully [26]. In any case, to ensure the viability of these electronics, more models have to be included in the development of new models for progressive trends that utilize methods such as learning (deep learning) [27], creative learning (reinforcement learning) [28], and other advancements that can lead to significant improvements in the field of cyber law. This research aims to study and analyse a set of learning models for predicting cyber dangers via spatial data [29]. The performances of diverse correlations, such as linear regression [5], SVM [6], RF [7], different trees (decision trees) [9], and KNN [30], are compared via exact criteria such as the mean supreme error (MAE) and mean square error (MSE) [31]. By determining this initial point, the optimal model will be accessible for analysing temporal data and predicting future targets, increasing the effectiveness of cybersecurity. In common, through this research, we have been working to determine how enhancement learning techniques can be connected in cyberspace, and through this comparison, we can decide the most reasonable model for analysing temporal data and predicting future attacks, which contributes to improving the viability of cybersecurity systems. Generally, this research contributes to a more profound understanding of how machine learning techniques can be connected in cybersecurity and provides a viable way to distinguish and respond to assaults utilizing worldly information[32][63].

2. RELATED WORK

In recent years, machine learning methods have essentially been created and have become major tools within the field of cybersecurity. Predicting temporal patterns in data is one of the foremost imperative applications that has attracted the attention of researchers. Many of these applications depend on the use of machine learning models to analyse historical data and predict the long-term behavior of cyber events.

In this setting, numerous models have been utilized in time prediction, such as linear regression, decision trees, RF, and methods such as slope boosting and neural networks [33][34]. Studies have shown that traditional models such as linear regression can be compelling in some basic cases, but these models typically suffer from failure in dealing with complex connections in data. In this manner, the use of decision trees has spread, which tends to supply more accurate solutions within the case of nonlinear data. In any case, these models confront the problem of overfitting when utilized on large or complex datasets, which has led researchers to utilize more advanced methods such as RF [35][36].

RFs are among the foremost effective methods for classification and prediction, as they create several few tree models and combine them to obtain accurate predictions [37].

In expansion, progressive boosting could be an effective model for complex predictions. This model depends on improving the performance of the fundamental model by creating several powerless models and combining them to improve the results [38][39]. Additionally, methods such as SVMs have been demonstrated to be effective in data classification and attack networks [40].

Although these methods yield excellent results in some applications, they may be ineffectual in the case of exceptionally large amounts of data or high-dimensional data. Hence, researchers have begun to utilize deep neural networks (profound learning) and LSTM models to improve prediction results in these cases [41][42]. These models are more complex in terms of the preparation and resources needed, but they yield precise results when dealing with complex data.

Moreover, some studies have shown that methods such as KNN can be effective in predicting cyber-attacks by distinguishing between typical and atypical patterns in networks [43]. In any case, they may have difficulty managing data that contain considerable noise or unsettling influences within the data.

Another viewpoint that has tended to be the use of time series analysis methods such as the ARIMA model, which is utilized to predict data that show steady patterns over time. Although this model has shown effectiveness in managing some basic temporal patterns, it may have trouble managing data that change quickly or contain complex changes [44].
These advances in models have led to modern patterns in which several methods are combined to improve execution. For example, RFs have been combined with neural networks to obtain more accurate predictions in cybersecurity applications. Some researchers have also begun to join support learning methods as a portion of their time expectation techniques for cyber events [46][47]. In addition, the greatest challenge in this field remains determining the foremost suitable model for each type of data or each type of attack. Studies have subsequently begun to focus more on conducting comprehensive comparisons between different models to distinguish the inconspicuous differences in the performance of each model on the basis of the characteristics of the data examined [48][61].

3. DATA AND METHODOLOGY

3.1 Data

The Malware dataset utilized in this study comprises a collection of data samples representing different types of malware and benign software. These datasets are fundamental in cybersecurity research, enabling the distinguishing proof, classification, and prediction of noxious software behaviors. Ordinarily, these datasets incorporate both static and dynamic highlights such as file properties, execution follows, network behavior, and system alterations made by the malware amid its runtime.
One of the foremost broadly utilized malware datasets is the CICIDS Malware dataset, which incorporates data collected from different simulated network attacks and real-world malware occurrences. This dataset provides information on the system and organizing execution, as well as the behavior of malware, encouraging the detection of irregular exercises and malicious patterns [49]. Moreover, Kaggle provides assorted malware datasets, counting labelled data that help in preparing and assessing machine learning models for malware detection [50][51].
The features included in these datasets extend from fundamental file attributes, such as record size, type, and hash values, to more complex dynamic attributes, such as system calls, memory utilization, and behavior preparation [52]. These features are significant for training machine learning models to distinguish between benign and noxious activities. The information preprocessing steps, including feature extraction and normalization, are crucial for increasing the accuracy of the models prepared on these datasets.
By leveraging these datasets, machine learning models can be created to classify malware precisely, detect modern variations, and predict potential future attacks, thereby improving the by and large-cybersecurity infrastructure [53]. As cyber dangers advance, persistent updating and extension of malware datasets remain important for maintaining strong detection systems [54].
Figure 1. Show the time series of the data.

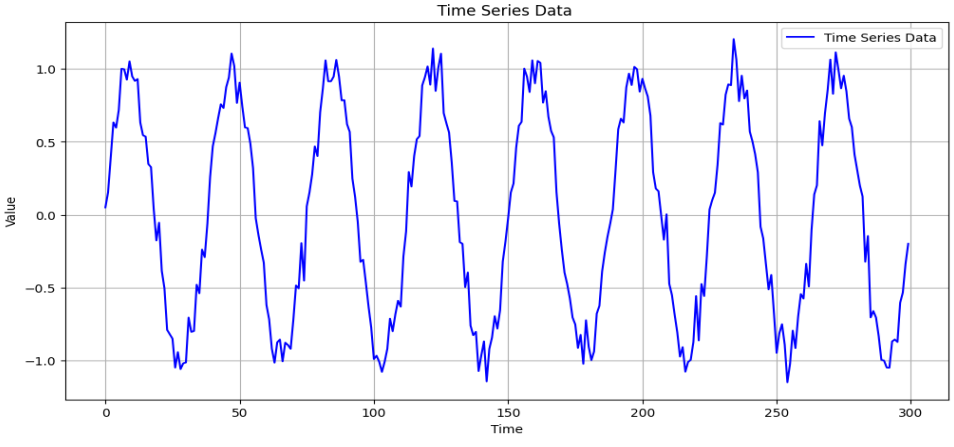


Fig. 1. Time series of the data.

3.2 Preprocessing

The process starts with the data era or importing real data into the system. Next, the data are preprocessed, which includes transforming it into a usable organ for machine learning models. This may include creating slacked features, which allow the models to memorize from past values to predict future values.

Once the data are preprocessed, another step is to partition the data into training and testing sets. In general, 80% of the data are utilized for training the models, whereas 20% are kept aside for testing the model performance.
In the model initialization process, various machine learning models are set up for the errand. These models may include direct regression, decision trees, R F, gradient boosting, KNN, SVR, LSTM, neural networks and hybrid models. After the models are initialized, they are prepared utilizing the training set. During this step, the machine learning algorithms learn from the data.
Once the models are trained, they are evaluated via performance metrics such as the MAE and MSE. These metrics offer assistance in surveying how well the models have learned from the data.
After evaluation, the models are utilized to predict future values on the test set. This permits one to see how well the models can generalize and make predictions on concealed data.
Another step is to compare the results. This includes comparing the predicted values with the actual values via the same execution metrics, namely, the MAE and MSE, to determine which model performs the best.
Visualization is an imperative portion of the method, where the anticipated values are plotted against the real values. This permits a visual comparison and reveals how closely the model predictions adjust with reality.
Finally, after all the models have been compared and assessed, you come to the conclusion, where you select the model that performs the best for your particular task.
This process outlines a normal machine learning workflow used for time series prediction tasks, where the objective is to prepare a show to predict future values on the basis of verifiable data, Figure 2. Preprocessing of the data.

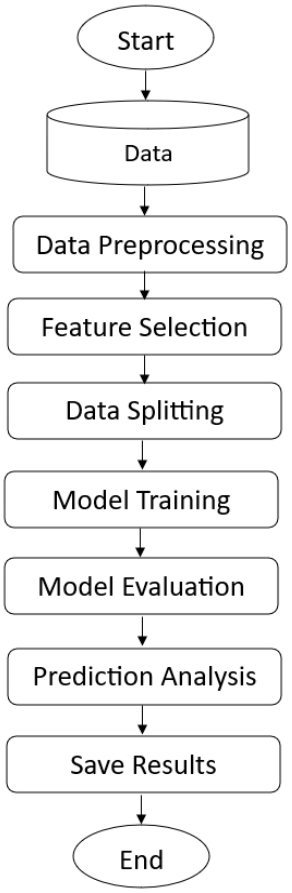


Fig. 2. Flowchart of the data preprocessing process.

3.3 Linear Regression

Linear regression could be a basic and widely utilized machine learning technique for predictive modelling. It accepts a linear relationship between the autonomous variables (features) and the subordinate variable (target). In cybersecurity, it is utilized to predict ceaseless variables, such as network traffic, resource utilization, or attack frequencies [56]. The model is based on the following equation:

$$ {}_n\beta_n x + \cdots + \quad {}_2\beta \quad {}_2 x + \quad {}_1\beta 1 x + \quad {}_0\beta = y \tag{1} $$

where:

- y is the predicted value (e.g., threat probability)
- ${}_n x_1, x_2 \dots, x$ are the features (e.g., packet count, time intervals)
- ${}_n \beta_1, \beta_2 \dots, \beta$ are the coefficients or weights of the mode

Linear regression minimizes the MSE during preparation:

$$^2(\hat{\imath}y - {}_i y) \sum_{i=1}^{N} \frac{1}{N} = \text{MSE} \quad (2)$$

- where ${}_i y$ is the true value and $\hat{\imath}y$hat is the predicted value.

3.4 Decision Tree

A decision tree can be a nonlinear model that splits data into subsets on the basis of feature values. It is given as a tree structure where each node gives to a decision on the basis of a feature, and each leaf node gives to a predicted output. Decision trees are prevalent in cybersecurity for detecting malicious actions in network activities or classifying diverse types of cyberattacks [57].

The decision run the show for a binary classification in a decision tree can be composed as:

$$\left.\begin{matrix} \text{if } x \in \text{ Leaf} & 1 \\ {}_2 \text{ if } x \in \text{ Leaf} & 0 \end{matrix}\right\} = f(x) \quad (3)$$

where:

- x is the input feature vector
- ${}_1$ Leaf a and ${}_2$ Leaf are the final prediction classes.
- The training objective is to reduce the Gini debasement or entropy, which measures the impurity of the decision nodes:

$$ {}_i^2 p \sum_{i=1}^{C} - 1 = \text{Gini}(D) \quad (4)$$

where:

- where ${}_i^2 p$ is the proportion of class iii in dataset D.
- $\boldsymbol{C}$ is the number of classes.

3.5 Decision Tree

RF is a gathering method that combines numerous decision trees to perform classification. Each tree within the forest is trained on a random subset of the data, and the ultimate prediction is based on the majority vote from all trees [58].

The final prediction for the RF is given by:

$$f_t(x) \sum_{t=1}^{T} \frac{1}{T} = f(x) \quad (5)$$

where:

- where $f_t(x)$ is the prediction of the ${}^{th}\boldsymbol{t}$ decision tree.
- where T is the total number of trees in the forest.

RFs are robust and can handle large datasets with high-dimensional features. They are effective in cybersecurity for detecting complex attack patterns by combining the predictions of numerous models.

3.6 Gradient boosting

Gradient boosting is an ensemble strategy that builds models successively, where each new model corrects the errors of the previous model. It is regularly utilized for binary classification assignments such as recognizing between normal and pernicious exercises in network traffic [59].

The gradient boosting model can be formulated as:

$$\text{grad}_m(x) \cdot \eta + F_{m-1}(x) = f(x) \quad (6)$$

where:

- $F_{m-1}(x)$ the previous model's prediction.
- η is the learning rate.
- $\text{grad}_m(x)$ is the gradient of the loss function at step m.

The model iteratively moves forward the predictions by minimizing the remaining error:

$$^{2}(f(x_i) - \quad _i y)\sum_{i=1}^{N} = \text{Residual Error} \tag{7}$$

3.7 K-Nearest Neighbor

K-nearest neighbors (KNN) could be a nonparametric strategy utilized for classification and regression. It works by finding the k closest neighbors of a data point and predicting the label on the basis of the lion's share class or average of those neighbors [60].

For classification, the prediction is given by:

$$\text{mode}(y_1, y_2, \dots, y_k) = \text{pred}y \tag{8}$$

where:

- $y_1, y_2, \dots, y_k$ are the labels of the k nearest neighbors.

For regression, the prediction is given by:

$$_i y\sum_{i=1}^{k} \frac{1}{k} = \text{pred}y \tag{9}$$

where:

- iy is the target value of the ^{th}i nearest neighbor

KNN is utilized in cybersecurity to classify network behavior as either typical or malicious based on likeness to known attention signatures.

3.8 Support Vector Recognition

Support vector recognition (SVR) is an effective classifier that identifies the hyperplane that best separates different classes [59]. For binary classification, the choice function of an SVM can be composed as:

$$b + w^T x = f(x) \tag{10}$$

where:

- is the weight vector, and $\boldsymbol{w}$.
- is the bias term b.

The model aims to maximize the edge between the classes, which is the separation between the hyperplane and the closest data focuses (support vectors). The edge is given by:

$$\frac{1}{\|w\|} = \text{Margin} \tag{11}$$

SVR is utilized in cybersecurity for classifying network activity and detecting anomalies by finding the ideal isolating hyperplane between typical and malicious exercises.

3.9 Long short-term memory (LSTM)

networks represent a powerful expansion of repetitive neural networks (RNNs), which are specifically designed to capture long-term conditions in successive data. This capability makes LSTMs especially effective for time series prediction tasks, where past perceptions essentially influence future predictions. An LSTM unit comprises memory cells and three types of gates: the forget gate f_t, the input gate i_t, and the output gate o_t These gates control the flow of data into, out of, and inside the memory cell, enabling the network to preserve relevant data over expanded sequences [55][56].

The basic equations overseeing the LSTM are as follows:

- Forget Gate:

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big) \tag{12}$$

- Input Gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{13}$$

- Cell State Update:

$$C_t = f_t \odot C_{t-1} + i_t \odot \grave{C}_t \tag{14}$$

- Output Gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{15}$$

- Hidden State:

$$h_t = o_t \odot \tanh(C_t) \tag{16}$$

In these equations, x_t denotes the input at time $t_t h_t$ and represents the hidden state, and C_t signifies the cell state. The activation function σ is the sigmoid function, which plays a crucial role in determining which information to keep or discard.

3.10 Neural Networks (ANNs)

Computational models are propelled by the interconnected neural structure of the human brain. They are composed of layers of interconnected nodes or "neurons," where each association has a related weight that is balanced amid the training handle. ANNs are especially effective for complex function approximation assignments, counting time arrangement prediction, owing to their ability to learn nonlinear connections from the data [57][58].

The fundamental structure of an ANN includes an input layer, one or more covered-up layers, and an output layer. The forward pass-through the network can be portrayed mathematically as follows:

- Input Layer:

Each input feature x_i is fed into the network, where i ranges from 1 to n, the total number of input features.

- Weighted Sum:

For a neuron j in a hidden layer, the weighted sum of inputs is calculated as:

$$z_j = \sum_{i=1}^{n} w_{ij} x_i + b_j \tag{17}$$

where w_{ij} are the weights, b_j is the bias for neuron j, and x_i are the inputs.

- Activation Function:

The weighted sum z_j is then passed through an activation function σ to introduce nonlinearity:

$$a_j = \sigma(z_j) \tag{18}$$

Common activation functions include sigmoid, hyperbolic tangent, and rectified direct unit (ReLU) functions.

- Output Layer:

The output y of the network can be computed in a similar manner:

$$y = \sigma\left(\sum_{j} w_{jo} a_j + b_o\right) \tag{19}$$

where w_{jo} are the weights to the output layer and where b_o is the output bias.

3.5 Hybrid Model

The hybrid model's prediction is determined via a comprehensive approach that aggregates the outputs of several individual models, thereby increasing the accuracy and robustness of the final prediction. The equation for the hybrid model is given by:

$$f_{\text{hybrid}}(x) = \alpha_1 f_{LR}(x) + \alpha_2 f_{DT}(x) + \alpha_3 f_{RF}(x) + \alpha_4 f_{GB}(x) + \alpha_5 f_{KNN}(x) + \alpha_6 f_{SVR}(x) + \alpha_7 f_{NN}(x) + \alpha_8 f_{LSTM}(x) \tag{20}$$

In this equation, $f_{LR}(x), f_{DT}(x), f_{RF}(x), f_{GB}(x), f_{KNN}(x), f_{SVR}(x), f_{NN}(x)$ and $f_{LSTM}(x)$ represent the predictions from each model. The weights $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8$ are determined on the basis of each model's performance on the particular dataset being analysed. This weighted sum permits the hybrid model to adjust the person strengths of each model while alleviating their weaknesses, leading to a more reliable prediction in complex time series prediction scenarios. By adjusting these weights according to model performance, the hybrid model dynamically adjusts to the characteristics of the data, ensuring optimal prediction accuracy.

3.6 Algorithm Hybrid Model Prediction Process

This algorithm outlines a hybrid model prediction process that coordinates numerous base models to increase the predictive performance. Initially, their parameters are set. Each demonstration is trained via an assigned training dataset. In this way, these trained models create predictions on test data. The hybrid model at that point combines these individual predictions through a specified aggregation function, such as a weighted average. The combined predictions are assessed against actual outcomes via metrics such as the MSE or MAE to survey their exactness. Finally, the calculation yields aggregated predictions as the final result. This approach leverages the qualities of different models to achieve more accurate and strong predictions.

Algorithm1 Hybrid Model Prediction Process

Require: Original dataset $\mathcal{D} = \{X_i\}_{i=1}^{N}$, where X_i represents input features.
Ensure: Predicted values $\hat{Y}$ for the dataset.
1: Initialize model parameters:
- $M_1, M_2, \ldots, M_k$: Base models (e.g., Linear Regression, SVR, Neural Networks).
- H : Hybrid combination function.
- θ_j : Hyperparameters for each base model M_j.
- p : Training-test split ratio.

2: for each base model M_j in $\{M_1, M_2, \ldots, M_k\}$ do
3: Train the model M_j using training data $(X_{\text{train}}, Y_{\text{train}})$:
4: $M_j \leftarrow \text{Train}(X_{\text{train}}, Y_{\text{train}}, \theta_j)$
5: end for
6: Generate predictions for each base model:
7: for each base model M_j do
8: Predict outputs for test data X_{test} :
9: $\hat{Y}_j = M_j(X_{\text{test}})$
10: end for
11: Combine predictions using the hybrid model:
12: Aggregate outputs $\hat{Y}_j$ using combination function H :
13: $\hat{Y}_{\text{Hybrid}} = H(\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_k)$
14: Evaluate the hybrid model:
15: Compute evaluation metrics (e.g., MSE, MAE) using:
16: $\text{Error} = \text{Metric}(Y_{\text{test}}, \hat{Y}_{\text{Hybrid}})$
17: Output predictions:
18: Return $\hat{Y}_{\text{Hybrid}}$ as the final predicted values.

3.7 Hybrid Model Training Process

Algorithm 2 explains the steps to train a hybrid model proficiently. First, the base models are initialized, including their parameters and hyperparameters, such as linear regression, SVR, and neural networks. Each base model is trained exclusively with the training dataset to optimize performance. After training, each model produces predictions for the test dataset to capture differing outputs. These predictions are combined via a hybrid function, such as a weighted sum or another aggregation strategy, to produce unified results. The combined predictions are assessed against the ground truth via metrics such as the RMSE or MAE to measure accuracy. If the execution is unsuitable, the hyperparameters are fine-tuned, and the training process is repeated. Finally, the hybrid model outputs the predictions, achieving strong and dependable results by leveraging the qualities of individual models.

Algorithm 2 Training Process

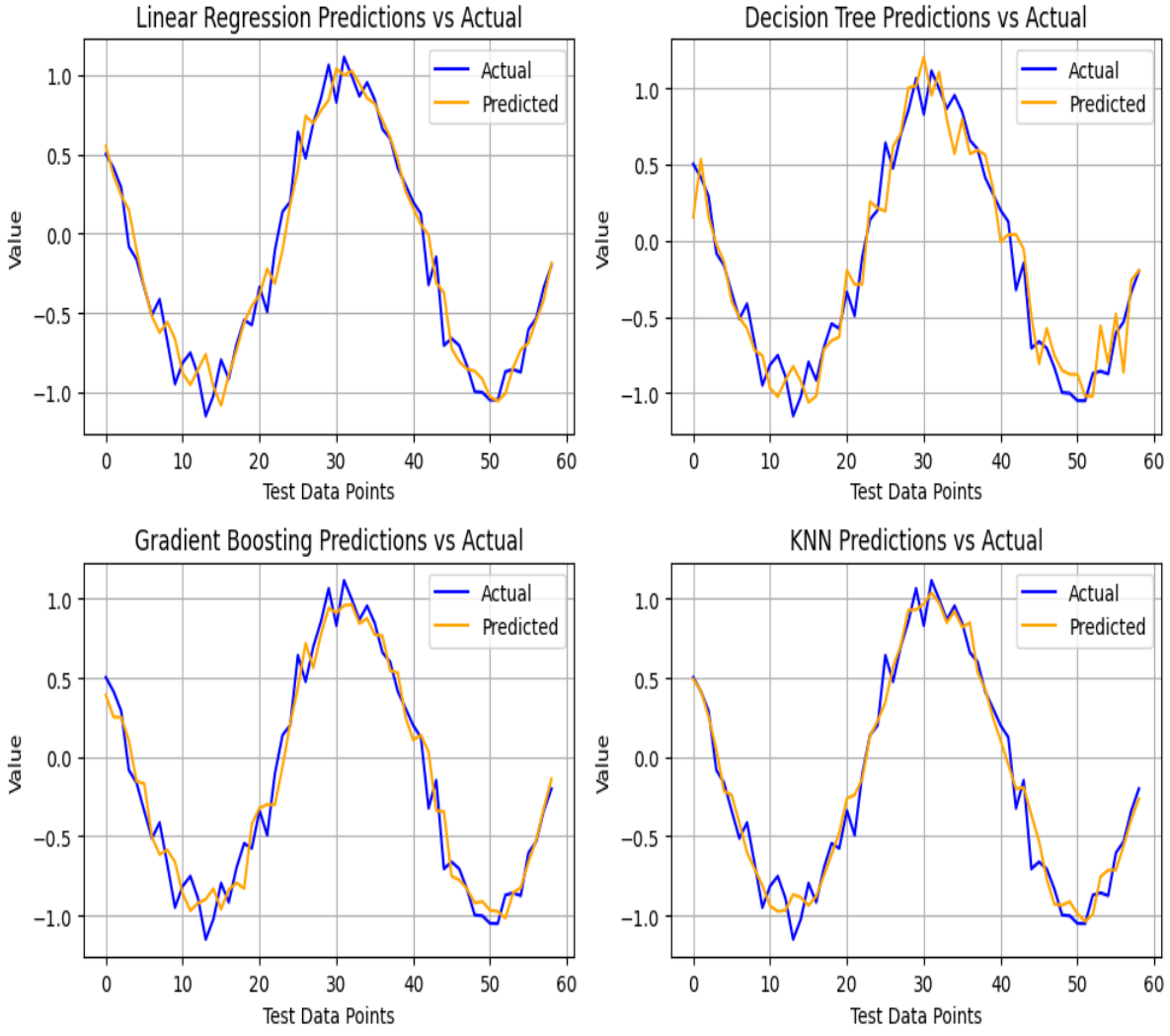
Require: Original dataset $\mathcal{D} = \{X_i, Y_i\}_{i=1}^{N}$, where X_i represents input features and Y_i represents target values.
Ensure: Trained hybrid model H_{trained} .
1: Initialize training parameters:
- $M_1, M_2, \ldots, M_k$: Base models (e.g., Linear Regression, SVR, Neural Networks).
- θ_j : Hyperparameters for each base model M_j.
- E: Evaluation metrics (e.g., MSE, MAE).

2: for each base model M_j in $\{M_1, M_2, \ldots, M_k\}$ do
3: Train the model M_j using the training data $(X_{\text{train}}, Y_{\text{train}})$:
4: $M_j \leftarrow \text{Train}(X_{\text{train}}, Y_{\text{train}}, \theta_j)$
5: end for
6: Validate base models:
7: Generate predictions for the validation data $X_{\text{valid:}}$
8: $\hat{Y}_j = M_j(X_{\text{valid}})$ for each model M_j.
9: Combine predictions:
10: Aggregate outputs $\hat{Y}_j$ using a hybrid combination function H :
11: $\hat{Y}_{\text{Hybrid}} = H(\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_k)$.
12: Evaluate hybrid model:
13: Compute evaluation metrics E using:
14: $\text{Score} = E(Y_{\text{valid}}, \hat{Y}_{\text{Hybrid}})$.
15: Optimize parameters if needed:
16: adjust θ_j for base models or the combination function H to minimize error.
17: Output trained hybrid model:
18: Return H_{trained} as the final hybrid model.

4. RESULTS AND DESICCATION

The performance of the six machine learning models linear regression, decision tree, RF, gradient boosting, KNN and SVR was evaluated utilizing 80% of the dataset for preparation and 20% for testing. The evaluation was based on two key metrics: the MAE and the MSE.

The hybrid model stands out as the best performing model compared to the others by conveying predictions that are reliably adjusted with the actual values over the test data points. The hybrid model easily captures the general trend, including sharp peaks and troughs, illustrating its strength in modelling complex and nonlinear patterns. Its capacity to take after the actual values with minimal deviation reflects its capacity to generalize well and reduce prediction errors. The alignment of predictions in regions with quick changes and transitions highlights its accuracy, unlike models such as linear regression (LR) or decision trees (DTs), which struggle in such areas. The prevalent performance of the hybrid model is attributed to its combination of multiple strategies that complement each other and increase the general predictive power. Unlike simpler models, which may underfit or overfat certain patterns, the hybrid model maintains a strategic distance from these issues by balancing the strengths of various basic algorithms. Moreover, it successfully minimizes clamor and makes strides soundness across the data, delivering smooth and accurate predictions. Comparing the hybrid model to others, such as KNN or RF, reveals that whereas those models perform well, the hybrid model reliably edges them in both accuracy and strength. In summary, the hybrid model exceeds expectations because of its capacity to capture the overall structure of the data while minimizing error across different test points. This makes it the foremost reliable and exact model. Figure 2. The figure shows a comparison of the actual and predicted results. The blue lines represent the real data points, whereas the orange lines indicate the predicted values. For all the models, the predicted values adjust closely with the actual values, indicating each model's accuracy, with variations in accuracy depending on the algorithm utilized.



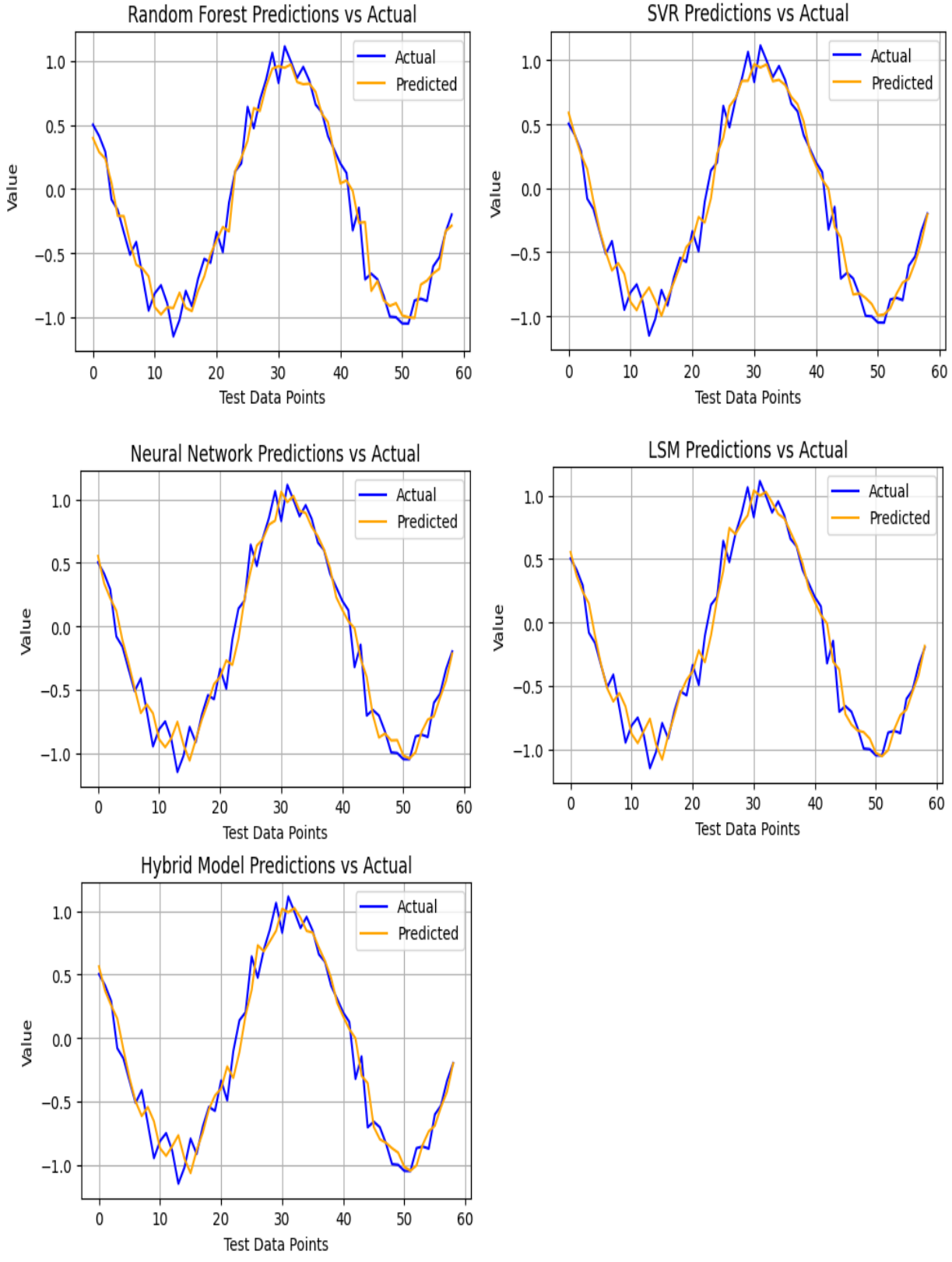


Fig. 3. The figure shows a comparison of the actual and predicted values for the six machine learning methods.

4.1 Evaluation of the MAEs and MSEs from the Chart

MAE and MSE are utilized to measure the predictive accuracy of the models. Both metrics are inferred from the differences between the predicted and actual values. Here, how they are calculated:
Equations:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} \left|Y_i - \hat{Y}_i\right| \tag{20}$$

The MAE measures the average supreme error between actual values (X_i) and predicted values ($\hat{Y}_i$).

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N} \left(Y_i - \hat{Y}_i\right)^2 \tag{21}$$

The MSE computes the average of the squared errors, penalizing larger deviations more heavily than the MAE does.
Figure 4. The Hybrid Model is used to illustrate its prevalence as the best-performing model when comparing the error metrics. The mean outright error (MAE) for the hybrid model is the lowest among all the models at 0.0935, whereas the mean squared error (MSE) is also remarkably low at 0.0137. These values demonstrate that Hybrid Show achieves the most elevated accuracy and minimizes both absolute and squared deviations compared with the other approaches. The performance of the hybrid model is better than that of simpler models such as direct relapse and choice trees, which yield greater errors because of their failure to capture complex relationships. Compared with other progressed models such as KNN, SVR, or LSTM, the hybrid model still achieves predominant results by combining predictions from different strategies to produce an adjusted and accurate output. The gathering nature of the hybrid model allows it to avoid the limitations of individual models by reducing inclination, minimizing overfitting, and increasing generalizability. Its capacity to achieve both a low MAE and MSE highlights its effectiveness in dealing with the test data and conveying ideal results. Unlike models such as decision trees, which have large deviations in error values, the hybrid model maintains consistency over the performance metrics. This combination of low error values, consistency, and stability confirms that the hybrid model is the best model in terms of performance, making it the foremost dependable choice for accurately predicting the data designs in this comparison.

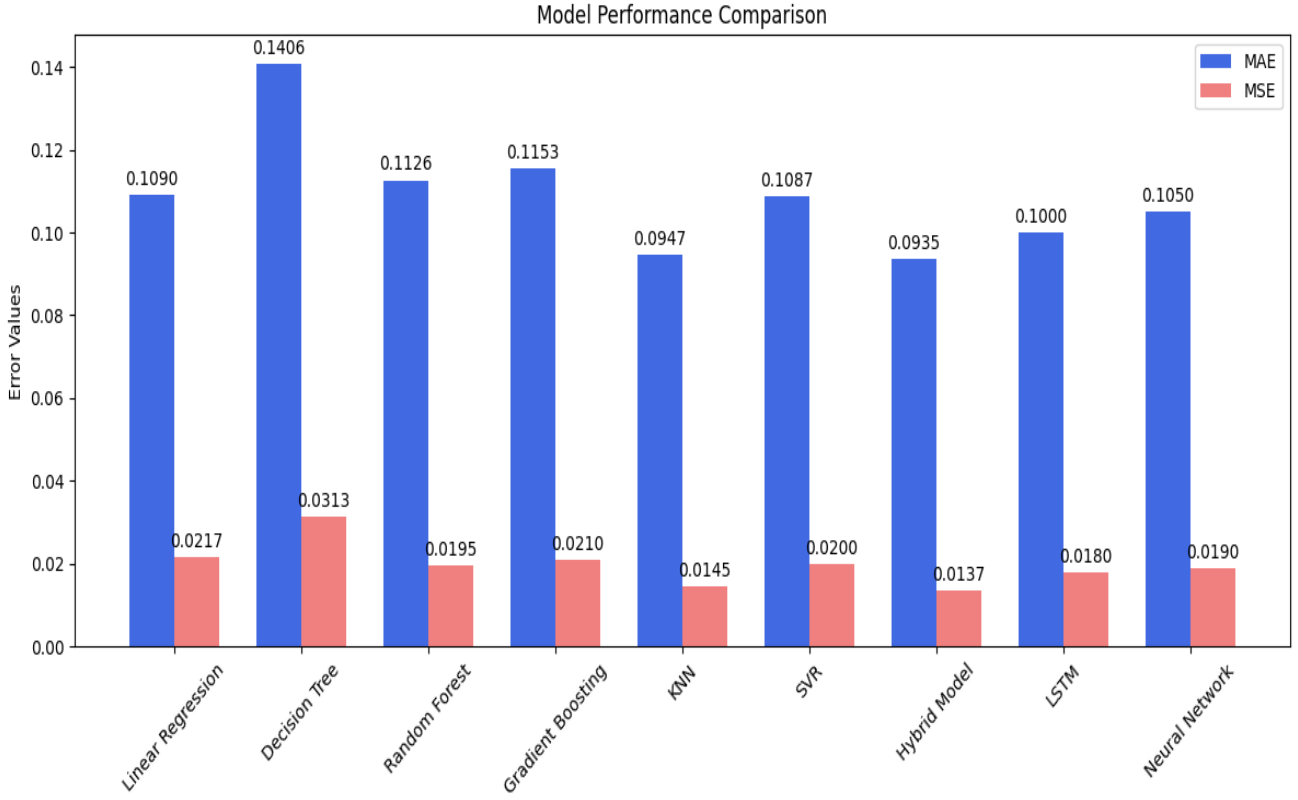


Fig. 4. Model performance metrics (MAE and MSE).

5. CONCLUSION

The rapid evolution of cyber threats requires advanced detection systems capable of promptly identifying and mitigating security issues. This study conducted a comprehensive evaluation of various machine learning models, including linear regression, decision tree, random forest, gradient boosting, k-nearest neighbors, support vector regression, long short-term memory (LSTM), and neural networks, to assess their effectiveness in cybersecurity anomaly detection. A hybrid model was built that integrates many machine learning techniques to improve efficiency and increase forecast accuracy. The results indicate that traditional models, such as linear regression and decision trees, achieve moderate accuracy but struggle with nonlinear data relationships. In contrast, ensemble methods such as random forest and gradient boosting offer improved classification accuracy and robustness against overfitting.

The hybrid model demonstrated superior performance by effectively combining the strengths of many models, hence reducing prediction errors and enhancing detection reliability. The integration of various methodologies improved the model's capacity to generalize across different datasets, making it more suitable for real-world cybersecurity applications. This study emphasizes the necessity of using time series data for forecasting cyber threats. Cyberattacks often exhibit evolving patterns, requiring the application of temporal analytical techniques. The incorporation of LSTM models into the hybrid model significantly improved its ability to detect complex attack patterns and predict potential threats. This competency is crucial for proactive cybersecurity initiatives, allowing organizations to mitigate attacks before their escalation.

Additionally, the evaluation metrics—the mean absolute error (MAE) and mean squared error (MSE)—were utilized to measure model performance. The hybrid model consistently achieves lower error rates, demonstrating its ability to reduce misclassifications and false positives.

The study demonstrated that feature selection and data preparation substantially enhanced model efficiency. Carefully selecting datasets, together with precise parameter optimization, improved the overall efficacy of the hybrid model. This research highlights the imperative for further advancements in cybersecurity modelling. While machine learning models provide valuable insights into cyber risks, improvements are needed to address challenges such as adversarial attacks, evolving threat landscapes, and the dynamic nature of cybercriminal conduct. Future research should explore the potential of deep learning architectures, reinforcement learning techniques, and federated learning to increase the adaptability and resilience of cybersecurity systems. This work substantially enhances cybersecurity by demonstrating the effectiveness of machine learning in anomaly detection. The proposed hybrid model exhibits enhanced efficacy, exceeding traditional methods in accuracy, robustness, and predictive capability. Organizations may significantly enhance their security posture and safeguard digital assets from new cyber threats by employing advanced analytics and integrating diverse machine learning methodologies. The deployment of real-time monitoring systems and AI-driven cybersecurity frameworks will be crucial for risk reduction and the preservation of a secure digital environment.

Conflicts of interest

The authors declare that they do not have any conflicts of interest with respect to their research. If there are no conflicts of interest, then the authors declare the following: "The authors declare no conflicts of interest".

Funding

The funding section of your journal paper template should provide a concise and transparent declaration of the financial support received to carry out the research presented in your paper.

Acknowledgement

The preferred spelling of the word “acknowledgement” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgements in the unnumbered footnote on the first page.

References

[1] J. P. Anderson, “Technical Report,” in Computer Security Threat Monitoring and Surveillance, P. James, Ed. Washington, DC, USA: Anderson Company, 1980.
[2] Y. Liao and V. R. Vemuri, “Use of k-nearest neighbor classifier for intrusion detection,” Comput. Secur., vol. 21, pp. 439–448, 2002.
[3] G. Kim, S. Lee, and S. Kim, “A novel hybrid intrusion detection method integrating anomaly detection with misuse detection,” Expert Syst. Appl., vol. 41, pp. 1690–1700, 2013.
[4] E. Dela Hoz, E. De La Hoz, A. Ortiz, J. Ortega, and B. Prieto, “PCA filtering and probabilistic SOM for network intrusion detection,” Neurocomputing, vol. 164, pp. 71–81, 2015.
[5] R. Sen, M. Chattopadhyay, and N. Sen, “An efficient approach to develop an intrusion detection system based on multi layer backpropagation neural network algorithm: IDS using BPNN algorithm,” in Proc. 2015 ACM SIGMIS Conf. Comput. People Res., Newport Beach, CA, USA, Jun. 4–6, 2015, pp. 105–108.

[6] L. Koc, T. A. Mazzuchi, and S. Sarkani, “A network intrusion detection system based on a hidden Naïve Bayes multiclass classifier,” Expert Syst. Appl., vol. 39, pp. 13492–13500, 2012.
[7] M. A. Khan and Y. Kim, “Deep learning-based hybrid intelligent intrusion detection system,” Comput. Mater. Contin., vol. 68, pp. 671–687, 2021.
[8] B. T. Devi, S. S. Thirumaleshwari, and M. A. Jabbar, “An appraisal over intrusion detection systems in cloud computing security attacks,” in Proc. 2020 2nd Int. Conf. Innov. Mech. Ind. Appl. (ICIMIA), Bangalore, India, Mar. 5–7, 2020, pp. 722–727.
[9] I. S. Thaseen, B. Poorva, and P. S. Ushasree, “Network intrusion detection using machine learning techniques,” in Proc. 2020 Int. Conf. Emerg. Trends Inf. Technol. Eng. (ic-ETITE), Tamil Nadu, India, Feb. 24–25, 2020, pp. 1–7.
[10] C. Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neural networks,” IEEE Access, vol. 5, pp. 21954–21961, 2017.
[11] S. Soheily Khah, P.-F. Marteau, and N. Bechet, “Intrusion detection in network systems through hybrid supervised and unsupervised machine learning process: A case study on the ISCX dataset,” in Proc. 2018 1st Int. Conf. Data Intell. Security (ICDIS), South Padre Island, TX, USA, Apr. 8–10, 2018, pp. 219–226.
[12] F. Folino, G. Folino, M. Guarascio, F. Pisani, and L. Pontieri, “On learning effective ensembles of deep neural networks for intrusion detection,” Inf. Fusion, vol. 72, pp. 48–69, 2021.
[13] B. A. Tama and S. Lim, “Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation,” Comput. Sci. Rev., vol. 39, p. 100357, 2021.
[14] K. Kim, M. E. Aminanto, and H. C. Tanuwidjaja, Network Intrusion Detection Using Deep Learning: A Feature Learning Approach. Berlin, Germany: Springer, 2018.
[15] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, M. Gabbouj, and D. J. Inman, “A review of vibration-based damage detection in civil structures: From traditional methods to machine learning and deep learning applications,” Mech. Syst. Signal Process., vol. 147, p. 107077, 2021.
[16] K. P. M. Kumar, M. Saravanan, M. Thenmozhi, and K. Vijayakumar, “Intrusion detection system based on GA-fuzzy classifier for detecting malicious attacks,” Concurr. Comput. Pract. Exp., vol. 33, p. 5242, 2021.
[17] M. H. Khan, “HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system,” Processes, vol. 9, p. 834, 2021.
[18] H. Zhang, L. Huang, C. Q. Wu, and Z. Li, “An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset,” Comput. Netw., vol. 177, p. 107315, 2020.
[19] M. K. Siddiqui and S. Naahid, “Analysis of KDD CUP 99 dataset using clustering based data mining,” Int. J. Database Theory Appl., vol. 6, pp. 23–34, 2013.
[20] A. Binbusayyis and T. Vaiyapuri, “Identifying and benchmarking key features for cyber intrusion detection: An ensemble approach,” IEEE Access, vol. 7, pp. 106495–106513, 2019.
[21] T. T. Bhavani, M. K. Rao, and A. M. Reddy, “Network intrusion detection system using random forest and decision tree machine learning techniques,” in Proc. 13th Int. Conf. Distributed Comput. Artif. Intell., Sevilla, Spain, Jun. 1–3, 2016, pp. 637–643, Springer, Berlin/Heidelberg, Germany, 2019.
[22] G. Karatas, O. Demir, and O. K. Sahingoz, “Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset,” IEEE Access, vol. 8, pp. 32150–32162, 2020.
[23] H. Xu, K. Przystupa, C. Fang, A. Marciniak, O. Kochan, and M. Beshley, “A combination strategy of feature selection based on an integrated optimization algorithm and weighted k-nearest neighbor to improve the performance of network intrusion detection,” Electronics, vol. 9, p. 1206, 2020.
[24] B. S. Bhati and C. S. Rai, “Analysis of support vector machine-based intrusion detection techniques,” Arab. J. Sci. Eng., vol. 45, pp. 2371–2383, 2019.
[25] I. S. Thaseen, J. S. Banu, K. Lavanya, M. R. Ghalib, and K. Abhishek, “An integrated intrusion detection system using correlation-based attribute selection and artificial neural network,” Trans. Emerg. Telecommun. Technol., vol. 32, p. 4014, 2021.
[26] S. Waskle, L. Parashar, and U. Singh, “Intrusion detection system using PCA with random forest approach,” in Proc. 2020 Int. Conf. Electronics Sustainable Commun. Syst. (ICESC), Coimbatore, India, Jul. 2–4, 2020, pp. 803–808.
[27] R. R. N. AlOgaili, O. A. Raheem, M. H. G. Abdkhaleq, Z. A. A. Alyasseri, S. A. A. A. Alsaidi, A. H. Alsaeedi, Y. R. Muhsen, and S. Manickam, "AntDroidNet Cybersecurity Model: A Hybrid Integration of Ant Colony Optimization and Deep Neural Networks for Android Malware Detection," Mesopotamian Journal of Cybersecurity, vol. 5, no. 1, pp. 104–120, 2025. DOI: https://doi.org/10.58496/MJCS/2025/008.
[28] M. M. Mijwil, M. Aljanabi, M. Abotaleb, B. S. Shukur, A. S. A. A. Sailawi, I. Bala, K. K. Hiran, R. Doshi, and K. Dhoska, "Exploring the Impact of Blockchain Revolution on the Healthcare Ecosystem: A Critical Review," Mesopotamian Journal of Cybersecurity, vol. 5, no. 1, pp. 78–89, 2025. DOI: https://doi.org/10.58496/MJCS/2025/006.
[29] M. A. Habeeb and Y. L. Khaleel, "Enhanced Android Malware Detection through Artificial Neural Networks Technique," Mesopotamian Journal of Cybersecurity, vol. 5, no. 1, pp. 62–77, 2025. DOI: https://doi.org/10.58496/MJCS/2025/005.
[30] S. D. Shamsi, A. S. H. M. Ali, and W. D. Shamsi, "Hybrid Cooperative Spectrum Structured (HCSS) Approach for Adaptive Routing in Cognitive Radio Ad Hoc Networks," Mesopotamian Journal of Cybersecurity, vol. 5, no. 1, pp. 23–38, 2025. DOI: https://doi.org/10.58496/MJCS/2025/003.
[31] J. Sharma, S. Sonia, K. Kumar, P. Jain, R. H. C. Alfilh, and H. Alkattan, "Enhancing Intrusion Detection Systems with Adaptive Neuro-Fuzzy Inference Systems," Mesopotamian Journal of Cybersecurity, vol. 5, no. 1, pp. 1–10, 2025. DOI: https://doi.org/10.58496/MJCS/2025/001.J.
[32] A. S. . Abdulbaqi, A. M. . Salman, and S. B. . Tambe, “Privacy-Preserving Data Mining Techniques in Big Data: Balancing Security and Usability”, SHIFRA, vol. 2023, pp. 1–9, Jan. 2023, doi: 10.70470/SHIFRA/2023/001
[33] A. Khalilian, A. Nourazar, M. Vahidi-Asl, and H. Haghighi, "G3MD: Mining frequent opcode subgraphs for metamorphic malware detection of existing families," Expert Systems with Applications, vol. 112, pp. 15–33, 2018.
[34] R. Lu, "Malware Detection with LSTM using Opcode Language," *arXiv preprint*, 2019.
[35] S. Choudhary and M. D. Vidyarthi, "A Simple Method for Detection of Metamorphic Malware using Dynamic Analysis and Text Mining," Procedia Computer Science, vol. 54, pp. 265–270, 2015.

[36] H. S. Galal, Y. B. Mahdy, and M. A. Atiea, "Behavior-based features model for malware detection," Journal of Computer Virology and Hacking Techniques, vol. 12, pp. 59–67, 2016.
[37] D. Thakur, S. Sah, P. Ailsinghani and V. Ranga, "A Comparative Study of Machine Learning Models for Network Traffic Classification in the IoT Landscape," 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2024, pp. 670-676, doi: 10.1109/ICAAIC60222.2024.10575268.
[38] K. Alakkari, A. A. Subhi, H. Alkattan, A. Kadi, A. Malinin, I. Potoroko, M. Abotaleb, and E.-S. M. El-kenawy, "A Comprehensive Approach to Cyberattack Detection in Edge Computing Environments," Journal of Cybersecurity and Information Management (JCIM), vol. 13, no. 01, pp. 69–75, 2024. DOI: https://doi.org/10.54216/JCIM.130107
[39] K. Omari, "Phishing Detection using Gradient Boosting Classifier," Procedia Computer Science, vol. 230, pp. 120–127, 2023. DOI: 10.1016/j.procs.2023.12.067
[40] U. Yokkampon, S. Chumkamon, A. Mowshowitz, R. Fujisawa, and E. Hayashi, "Anomaly Detection Using Support Vector Machines for Time Series Data," Journal of Robotics, Networking and Artificial Life, vol. 8, no. 1, pp. 41–46, June 2021. DOI: 10.2991/jrnal.k.210521.010.
[41] T. Tsigkritis, G. Groumas, and M. Schneider, "On the Use of k-NN in Anomaly Detection," Journal of Information Security, vol. 9, pp. 70–84, 2018. DOI: 10.4236/jis.2018.91006
[42] H. Zare Moayedi and M. A. Masnadi-Shirazi, "Arima model for network traffic prediction and anomaly detection," 2008 International Symposium on Information Technology, Kuala Lumpur, Malaysia, 2008, pp. 1-6, doi: 10.1109/ITSIM.2008.4631947
[43] Z. T. Nayyef, M. M. Abdulrahman, and N. A. Kurdi, "Optimizing Energy Efficiency in Smart Grids Using Machine Learning Algorithms: A Case Study in Electrical Engineering", SHIFRA, vol. 2024, pp. 46–54, Apr. 2024, doi: 10.70470/SHIFRA/2024/006
[44] G. Ghanshyam and P. Pandey, "A KNN-Based Intrusion Detection System for Enhanced Anomaly Detection in Industrial IoT Networks," International Journal of Innovative Research in Technology and Science, vol. 12, no. 6, pp. 1-7, Nov. 2024.
[45] M. Landauer, F. Skopik, B. Stojanović, et al., "A review of time-series analysis for cyber security analytics: from intrusion detection to attack prediction," International Journal of Information Security, vol. 24, no. 3, 2025. DOI: 10.1007/s10207-024-00921-0.
[46] I. Tareq, B. M. Elbagoury, S. A. El-Regaily, and E.-S. M. El-Horbaty, "Deep Reinforcement Learning Approach for Cyberattack Detection", Int. J. Onl. Eng., vol. 20, no. 05, pp. pp. 15–30, Mar. 2024.
[47] Y. Xin et al., "Machine Learning and Deep Learning Methods for Cybersecurity," in IEEE Access, vol. 6, pp. 35365-35381, 2018, doi: 10.1109/ACCESS.2018.2836950.
[48] V. Tripathi, A. Dubey, K. Sathvik, and N. Subhashini, "A Comparative Study of Machine Learning Algorithms for Anomaly Based Network Intrusion Detection System," in Topical Drifts in Intelligent Computing. ICCTA 2021. Lecture Notes in Networks and Systems, vol. 426, J. K. Mandal, P. A. Hsiung, and R. Sankar Dhar, Eds. Singapore: Springer, 2022. DOI: 10.1007/978-981-19-0745-6_2
[49] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, S. Chen, D. Liu, and J. Li, "Performance Comparison and Current Challenges of Using Machine Learning Techniques in Cybersecurity," Energies, vol. 13, no. 10, p. 2509, 2020. DOI: 10.3390/en13102509
[50] H. N. Chethan, "Network Intrusion Dataset," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset. [Accessed: Augast. 23, 2024].
[51] G. B. Mensah, "The Era of AI: The Impact of Artificial Intelligence (AI) and Machine Learning (ML) on Financial Stability in the Banking Sector", EDRAAK, vol. 2024, pp. 43–48, Jun. 2024, doi: 10.70470/EDRAAK/2024/007
[52] G.Belding, "Top 7 malware sample databases and datasets for research and training," infosecinstitute, 2021. [Online]. Available: https://www.infosecinstitute.com/resources/malware-analysis/top-7-malware-sample-databases-and-datasets-for-research-and-training/. [Accessed: Julay. 3, 2024].
[53] D. Gupta and R. Rani, "Improving malware detection using big data and ensemble learning," Computers & Electrical Engineering, vol. 86, p. 106729, 2020. DOI: 10.1016/j.compeleceng.2020.106729.
[54] Muchamad Bachram Shidiq, W. Gata, S. Kurniawan, D. D. Saputra, and S. Panggabean, "Time Effort Prediction Of Agile Software Development Using Machine Learning Techniques", j. inspir., vol. 13, no. 2, pp. 39–48, Dec. 2023.
[55] Y. A. Ahmed, B. Koçer, S. Huda, B. A. S. Al-Rimy, and M. M. Hassan, "A system call refinement-based enhanced Minimum Redundancy Maximum Relevance method for ransomware early detection," Journal of Network and Computer Applications, vol. 167, p. 102753, 2020.
[56] J. Singh and J. Singh, "Detection of malicious software by analysing the behavioral artifacts using machine learning algorithms," Information and Software Technology, vol. 121, p. 106273, 2020.
[57] M. Norouzi, A. Souri, and M. S. Zamini, "A Data Mining Classification Approach for Behavioral Malware Detection," Journal of Computer Networks and Communications, vol. 2016, pp. 1–9, 2016.
[58] A. Arabo, R. Dijoux, T. Poulain, and G. Chevalier, "Detecting Ransomware Using Process Behavior Analysis," Procedia Computer Science, vol. 168, pp. 289–296, 2020.
[59] M. Belaoued, A. Boukellal, M. A. Koalal, A. Derhab, S. Mazouzi, and F. A. Khan, "Combined dynamic multifeature and rule-based behavior for accurate malware detection," International Journal of Distributed Sensor Networks, vol. 15, p. 155014771988990, 2019.
[60] J. B. Fraley and M. Figueroa, "Polymorphic malware detection using topological feature extraction with data mining," in Proceedings of the SoutheastCon 2016, Norfolk, VA, USA, 2016, pp. 1–7..
[61] I. Bala, I. A. Pindoo, M. M. Mijwil, M. Abotaleb, and W. Yundong, "Ensuring Security and Privacy in Healthcare Systems: A Review Exploring Challenges, Solutions, Future Trends, and the Practical Applications of Artificial Intelligence," Jordan Medical Journal, vol.58, no.2, pp.250-270, July 2024.
[62] A. K. Abed , Tran., "Utilizing Artificial Intelligence in Cybersecurity: A Study of Neural Networks and Support Vector Machines", *BJN*, vol. 2025, pp. 14–24, Feb. 2025, doi: 10.58496/BJN/2025/002.
[63] L. Hussain, "Fortifying AI Against Cyber Threats Advancing Resilient Systems to Combat Adversarial Attacks", *EDRAAK*, vol. 2024, pp. 26–31, Mar. 2024, doi: 10.70470/EDRAAK/2024/004.