Received: September 29, 2024. Revised: November 21, 2024.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Optimizing Intrusion Detection in IoT through a Combination of Feature Selection and Deep Feedforward Neural Network

Syariful Ikhwan[1,4]* Purwanto Purwanto[2] Adian Fatchur Rochim[3]

[1,2] *Doctoral Program of Information System, School of Postgraduate Studies, Diponegoro University, Indonesia*
[2]*Department of Chemical Engineering, Faculty of Engineering, Diponegoro University, Indonesia*
[3]*Department of Computer Engineering, Faculty of Engineering, Diponegoro University, Indonesia*
[4]*D3 Telecommunication Engineering, Telkom University, Indonesia*
* Corresponding author's Email: syarifuli@telkomuniversity.ac.id

Abstract: Intrusion detection is a crucial aspect of maintaining the security of computer network devices. With the advancement of Internet of Things (IoT) technology, the need for intrusion detection has become even more critical due to the many IoT devices that remain inadequately protected due to resource constraints. Although various intrusion detection methods have been developed, many of them have not been optimized to address the resource limitations of IoT devices, such as limited computational capacity and low power consumption. Furthermore, existing feature selection methods often overlook the potential of combining various selection techniques to enhance accuracy and efficiency. This research introduces a novel intrusion detection framework that integrates multiple feature selection algorithms—Pearson Correlation, Spearman Correlation, and Mutual Information—during the pre-processing phase, aiming to reduce data complexity and improve classification accuracy. Unlike prior approaches that apply single-method feature selection, our method combines these techniques to capture both linear and non-linear dependencies, ensuring that only the most relevant features are retained. When evaluated on the UNSW-NB15 dataset, the proposed Deep Feedforward Neural Network (DFNN) classifier achieved an accuracy of 89.23%, outperforming other models in terms of accuracy and training efficiency. These improvements make the model better suited for real-world IoT environments with computational limitations.

Keywords: IoT, IDS, Neural network, Feature selection, Network security.

1. Introduction

The Internet of Things (IoT) has connected various devices and is widely used across multiple sectors. However, the growing number of IoT users and services presents significant security challenges. Security vulnerabilities in IoT systems can jeopardize applications and services, necessitating further research to ensure the confidentiality, integrity, and availability of data are maintained [1]. One of the solutions to address threats to computer networks and information systems is an intrusion detection system. Intrusion detection enables quicker identification and response to potential threats, allowing for early actions that can safeguard an institution's valuable assets. Intrusion detection systems are designed to detect suspicious behavior or patterns in network traffic and system activities. This detection is crucial for identifying potential threats to the existing systems. The most important factors in combating attacks are early detection and the segregation of network traffic [2]. The development of intrusion detection systems is then carried out with a focus on achieving accurate classification and minimizing false alarms [3].

Researchers have developed various methods to enhance the effectiveness of network intrusion detection systems. One such method is the use of artificial intelligence, which allows the system to recognize intrusions more accurately. The application of Deep Neural Networks (DNN) in intrusion detection systems has shown promising results [4][5]. DNNs can learn from patterns in large

datasets, which enables them to improve accuracy and reduce the number of false alarms. DNNs are highly effective in dynamic networks that frequently undergo changes due to their ability to continuously learn and adapt.

The capability of the developed model in detecting intrusions in artificial intelligence largely depends on the pre-processing stage. At this stage, input features are used as data inputs for the model to learn patterns associated with attacks. Feature selection becomes crucial in enhancing the efficiency and effectiveness of the intrusion detection model [6]. The purpose of feature selection is to identify and select the most relevant and informative features to be used as inputs when training the detection model. This helps reduce data dimensionality, thereby lowering model complexity and speeding up the training process. In the context of intrusion detection, speed and accuracy are of utmost importance [7]. With feature selection, the effective features will provide faster and more accurate insights into suspicious activities.

This research aims to reduce data dimensionality by applying a feature selection method that combines Pearson Correlation, Spearman Correlation, and Mutual Information. Meanwhile, the intelligent technique for classifying attacks in this study uses a Deep Feedforward Neural Networks (DFNN) classifier. The DFNN is a type of DNN in which information flows forward from the input layer to the output layer without any feedback loops. The proposed method in this research contributes to three key areas:

- Propose an intrusion detection method for IoT networks based on DFNN, with a focus on accuracy and lightweight, fast computational processes.
- Introduce a combination of Pearson Correlation, Spearman Correlation, and Mutual Information algorithms to select features closely related to the labels and reduce the complexity of unnecessary features.
- Evaluate the proposed method using a public dataset, which has been widely referenced in intrusion detection research.

The remainder of this article is structured as follows. Section 2 presents research related to feature selection methods and the use of DFNN as a classifier. Section 3 describes the proposed method for intrusion detection using DFNN and feature selection in this study. Section 4 presents the analysis and outcomes of the methods implemented in the tests. Section 5 provides conclusions based on the test results.

2. Literature review

Research on intrusion detection to address network security challenges has advanced rapidly. One approach that continues to be refined is the application of DFNN to distinguish between attack and non-attack scenarios [2]. Efforts to enhance DFNN performance focus on improving the quality of datasets used as model input. Feature selection in the dataset is conducted through various methods to retain informative features and discard less useful ones. Broadly, there are three main approaches for selecting features to be used as classifier input, enabling the model to learn more effectively and accurately: the filter-based approach, the wrapper-based approach, and the embedded approach.

Studies employing filter-based methods select the optimal features by measuring the statistical relationship between features and class labels, without involving any specific model in the selection process. Among the relevant studies, [8] proposes the FS-DL method, which combines feature selection and deep learning to enhance network intrusion detection. Feature selection in this study utilizes standard deviation and association rule mining techniques. This method is designed to identify and select key features that contribute to improving accuracy, reducing redundancy, and lowering computational load. FS-DL minimizes the number of features used and adopts a simple neural network structure. Experimental results from the NSL-KDD and UNSW-NB15 datasets indicate enhanced detection performance with a reduced number of features. FS-DL has been successfully implemented in a Software-Defined Networking (SDN) environment, showcasing its adaptability to various settings. However, despite FS-DL's strong performance in binary classification, its accuracy in multi-class classification remains low. The study conducted by [7] addresses feature selection in Intrusion Detection Systems (IDS) based on DNN with the aim of enhancing accurate classification and reducing the number of false alarms. The datasets utilized in this research are NSL-KDD, UNSW_NB-15, and CIC-IDS-2017. The research methodology includes the use of standard deviation, mean, and median to identify the most relevant features. The selected features were then applied to the DNN for intrusion detection and classification. The model's effectiveness was assessed using metrics including accuracy, precision, recall, F-score, and false positive rate. The results indicate that statistical approaches in feature selection can effectively enhance intrusion detection. While the study highlights the advantages of the feature selection process, it also emphasizes the

need for further evaluation of the algorithms used on more diverse datasets. The work by [9] presents an advanced feed-forward neural network (FFDNN) model that incorporates a filter-based feature selection technique to tackle the increasing security challenges in cloud computing. The FFDNN classifier is provided with feature inputs that have undergone selection using the Extra Tree Classifier, with the aim of improving both accuracy and efficiency in intrusion detection systems. The FFDNN model reached its peak accuracy of 99.53% on the NSL-KDD dataset, 94.45% on the UNSW-NB15 dataset, and 99.80% on the CIC-IDS 2017 dataset. A notable drawback of this research is the high computational burden when applied in real-time scenarios due to the complexity of deep learning and extensive data processing involved in feature selection.

The second approach utilized for feature selection is the wrapper method. This technique selects features by directly evaluating the model's performance for each combination of features, thereby choosing features based on their impact on the model's accuracy. The investigation by [10] applied the Wrapper-Based Feature Extraction Unit (WFEU) using Extra Trees classification to enhance the detection capabilities of Feed-Forward Deep Neural Networks (FFDNN). This study aimed to address the limitations of wireless intrusion detection systems. Efficient and accurate intrusion detection in wireless networks is crucial for network security in the era of the Internet of Things (IoT). Testing on the UNSW-NB15 dataset, WFEU produced an optimal feature vector with 22 attributes, achieving an overall accuracy of 87.10% for binary classification and 77.16% for multi-class classification. Meanwhile, on the AWID dataset, the feature vector reduced to 26 attributes yielded an overall accuracy of 99.66% for binary classification and 99.77% for multi-class classification. The presence of numerous features during the training process can result in prolonged processing times and substantial computational demands. Thus, dimension reduction is essential in the preprocessing phase to manage these challenges effectively. The study [11] combines Principal Component Analysis (PCA) and Grey Wolf Optimization (GWO) for feature engineering as input to a Deep Neural Network (DNN) classifier. PCA is used to reduce data dimensions by simplifying the feature space without sacrificing essential information. Conversely, GWO, inspired by the hunting behavior of grey wolves within their social hierarchy, is employed to enhance feature selection. The results of this study demonstrate a 15% increase in classification accuracy and a 32% reduction in training time compared to various machine learning methods, such as K-Nearest Neighbor (KNN), Naive Bayes, Random Forest (RF), SVM, and DNN without PCA-GWO. The exploration conducted by [12] aims to enhance intrusion detection in Wi-Fi networks by proposing Ensemble Binary Detection Models (EBDM). EBDM converts multi-class detection into binary detection models by utilizing a Binary Model (BM) dedicated to each target class. This approach integrates Feature Selection (FS), Stack Auto Encoder (SAE), and Random Sampling (RS) methods to improve detection performance, while also introducing Ensemble FS (EFS) and Ensemble DB to enhance detection accuracy. EFS consists of three Decision Trees (DT) for feature ranking and two for feature selection. Feature selection is performed using Support Vector Machine and Logistic Regression (LR). The integration of Squeeze-and-Excitation blocks with a DNN was also implemented to enhance DNN performance. Experimental results demonstrate that EBDM can more accurately detect three types of wireless attacks compared to previous approaches, particularly on the AWID dataset. For four-class detection, EBDM achieved an accuracy of 99.625% and an F1 Score of 99.250%. The best detection results in Binary Models (BMs) showed accuracy ranging from 98.665% to 99.998% and F1 Scores ranging from 80.462% to 99.961%, with the number of selected features (NSF) ranging from 12 to 63. The study [6] discusses the selection of Dimension Reduction (DR) methods using Linear Discriminant Analysis (LDA), t-Distributed Stochastic Neighbor Embedding (t-SNE), and Principal Component Analysis (PCA), with a DNN as the classifier. The results indicate that the LDA method has a faster training time compared to PCA, while t-SNE has the longest training time. However, t-SNE demonstrates better accuracy than the other methods. Among the three Feature Selection (FS) methods employed—filter, wrapper, and embedded—the wrapper method achieved higher accuracy, although it also incurs higher computational costs.

The third approach employed for feature selection is the embedded method. In this approach, feature selection occurs during the model training process, prioritizing the most relevant features based on their weights or contributions to the classification outcome. The embedded method for selecting the best features to improve system detection performance was implemented in [13]. This study evaluates the combination of several regularization techniques, L1 regularization (Lasso), L2 regularization (Ridge), elastic net, and dropout, in terms of their impact on DNN-based IDS

performance. Experiments were conducted using the CIC-IDS-2017, UNSW_NB-15, and NSL-KDD datasets. Hyperparameter optimization using GridSearchCV was also employed to determine the dropout probability values. The results indicate that dropout is more effective compared to L1, L2, and elastic net regularization, and that combining dropout with other regularization techniques yields better results. Deep learning was utilized by [14] for feature selection to be used as input for the DNN classifier. The proposed deep learning model is a Deep Autoencoder (DAE). Hyperparameter optimization was also performed through a combination of grid search and random search to improve model performance. The performance of the proposed model was assessed using the NSL-KDD and CSE-CIC-ID2018 datasets. The results obtained show better performance compared to other models. However, the significant time and computational resources required still pose a challenge in this research. In [15], the feature selection method utilized is the Greedy Recursive Feature Elimination with Cross-Validation (GRFECV) algorithm. This method is an enhancement of the popular Recursive Feature Elimination with Cross-Validation (RFECV) algorithm, incorporating a 'greedy' approach to select an optimal subset of features. GRFECV is designed to address the limitations of RFECV by more comprehensively considering the importance of each feature through cross-validation, enabling more efficient and effective feature selection to improve detection accuracy. The testing results indicate that GRFECV outperforms RFECV and other algorithms. However, the method has notable drawbacks, including substantial computational requirements when processing large datasets with many features. Additionally, the algorithm is highly dependent on the model used, as features deemed important in one model may be less significant in another.

Among the three approaches examined, the filter-based method exhibits the lowest power consumption, as it does not require model involvement during the feature selection process [16]. Subsequent studies have implemented the filter-based approach utilizing only a single statistical method. The Pearson correlation method was employed in the study [17] to select features with high correlation to the target feature, aiming to enhance the performance of machine learning models. In this research, around 40 features were chosen from a total of 77 features in the CICIDS 2017 dataset using Pearson correlation. The results demonstrated an improvement in both the efficiency and overall performance of the system. IoT networks are vulnerable to cyber attacks due to the large number of connected devices with limited computational power and storage capacity. The study in [18] highlights the need for effective intrusion detection systems for Internet of Things (IoT) networks. The proposed method involves filter-based feature selection using Pearson correlation to eliminate less relevant features and the use of Generative Adversarial Networks (GANs) to address class imbalance issues in the UNSW-NB15 dataset. The classifier used in this study is a DNN. The results indicate that the proposed model successfully improved intrusion detection accuracy from 84% to 91% with the assistance of synthetic data generated by GANs. A drawback of using a single statistical method, such as Pearson correlation, lies in its limited ability to detect only linear relationships, leaving non-linear relationships unidentifiable.

Based on the literature discussed in this section, there is an opportunity to enhance the capabilities of filter-based methods, which have demonstrated high accuracy alongside low power consumption. The integration of multiple filtering techniques has the potential to improve intrusion detection accuracy while maintaining energy efficiency. Previous studies have not integrated multiple statistical methods concurrently, even though this approach could harness the strengths of each method, especially for capturing both linear and non-linear relationships simultaneously. Therefore, this research aims to combine Pearson Correlation, Spearman Correlation, and Mutual Information to develop an effective and lightweight Intrusion Detection System (IDS) suitable for practical application in an IoT environment.

3. Experimental methodology

The experimental methodology consists of several stages. The first stage involves optimizing the dataset by removing zero-valued data, then standardizing the values and encoding the data that is still in numeric form. The second stage is selecting the features to be included in the classifier. The third stage involves training the model using the dataset to obtain the best intrusion detection model. Finally, the fourth stage is to evaluate the model's performance metrics by testing it on the dataset. The schematic of the proposed approach is as shown in Figure 1.

3.1 Proposed feature selection technique for intrusion detection and classification

Statistical feature selection methods, such as Pearson correlation, Spearman correlation, and Mutual Information, are extensively applied in machine learning due to their efficiency in

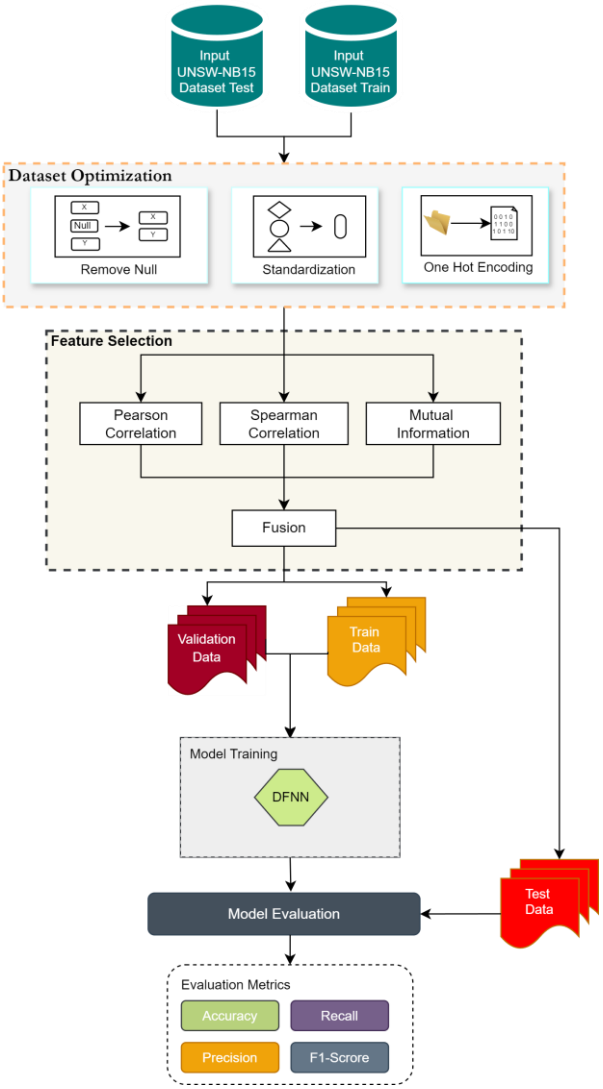

Figure 1. Schematic of the proposed approach

identifying relevant features. These techniques have consistently demonstrated strong performance in isolating features that contribute meaningfully to classification tasks, garnering positive evaluations in various applications. By quantifying relationships between variables, these methods assist in refining feature sets, enhancing model interpretability, and optimizing classification accuracy.

Pearson's correlation coefficient can be used to evaluate the similarity between two features and to determine how it affects the parameter for accelerating convergence [19]. Pearson's correlation algorithm is a statistical method utilized to assess the intensity and orientation of a linear relationship between two quantitative variables. Pearson's correlation value ranges from -1 to +1, and its correlation coefficient is denoted by r. A value of -1 indicates a perfect negative correlation, meaning there is a perfectly linear negative relationship between two variables. A value of +1 indicates a perfect positive correlation, meaning there is a perfectly linear positive relationship between two variables. A value of 0 indicates the absence of any linear relationship between the two variables, meaning that changes in one variable do not result in predictable changes in the other. The value of r is formulated as follows Eq. (1).

$$r = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2 \sum(y_i - \overline{y})^2}} \quad (1)$$

where x_i and y_i represent the individual values of the two measured variables, $\overline{x}$ is the average of the x variable, and $\overline{y}$ is the average of the y variable.

Spearman correlation, commonly referred to as Spearman's rank correlation coefficient or Spearman's rho, is a method for evaluating the relationship between two variables when the data is expressed in ranked form [20, 21]. This method assesses the monotonic relationship between the variables, indicating that as one variable rises, the other variable tends to consistently increase or decrease as well. The formula used to determine the value of the Spearman correlation coefficient is Eq. (2).

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (2)$$

where ρ is the Spearman correlation coefficient, d_i is the difference in the ranks of each observation pair, and n is the number of entries. A Spearman's rank correlation coefficient value of +1 indicates a perfect positive monotonic relationship, meaning that as one variable increases, the other variable also increases. Conversely, a value of -1 signifies a perfect negative monotonic relationship, where an increase in one variable corresponds to a decrease in the other. A value of 0 signifies that there is no monotonic relationship between the two variables involved.

The Mutual Information (MI) algorithm is a statistical method used to evaluate the extent to which two random variables share information. The value of mutual information indicates the level of dependence between the two variables; a greater value signifies a stronger dependence [22]. Unlike the Pearson correlation, which is limited to linear relationships, MI is capable of capturing both linear and non-linear relationships between variables. The formula used to determine the relationship between variables in MI is given in Eq. (3) or Eq. (4).

$$I(X;Y) = H(X) - H(X \mid Y) \quad (3)$$

or

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \quad (4)$$

where $H(X)$ is the entropy of the X variable, $H(Y)$ is the entropy of the variable Y, $H(X,Y)$ is the combined entropy of X and Y, and $H(X \mid Y)$ represent the conditional entropy of X given Y.

Based on the studies previously discussed, this research integrates the Pearson correlation, Spearman correlation, and Mutual Information methods by selecting the best features from each approach. This integration aims to leverage the strengths of each method while compensating for their individual weaknesses, thereby developing a more robust feature selection technique. Algorithm 1 outlines the steps from the initial dataset processing, feature selection based on the proposed method, classification, and ultimately the generation of the confusion matrix.

3.2 Preprocessing dataset

The dataset utilized in this research is publicly accessible and provided in CSV format. The dataset was read and stored in a variable before undergoing a cleaning process to remove duplicate entries and missing values. Encoding was then performed to convert categorical values into a format that can be processed by machine learning algorithms. In this research, One-Hot Encoding was used to convert categorical features into a numerical format. Along with converting certain features to numeric values, the labels were also encoded using One-Hot Encoding.

Data normalization is essential to standardize the characteristics of the data within a specific range, such as transforming all values to fall within the 0 to 1 range. This transformation significantly aids in accelerating convergence and enhances the comparability of features. In this research, Standard Scaler was applied to normalize the dataset.

3.3 Feature selection

Feature selection in this study was conducted in two forms, commonly referred to as feature engineering: selecting a subset of features based on specific criteria and reducing the number of features. The aim of feature selection is to ensure that only the most contributive features are included in the training process, without sacrificing essential information. Feature selection was performed using the Pearson, Spearman, and Mutual Information algorithms. The features identified by these three algorithms were then combined into a single set, with duplicate features removed beforehand.

Data splitting was conducted by dividing the dataset into separate parts for training, validation, and testing purposes. This division aims to ensure that the model developed can generalize effectively to unseen data. In this study, the data was split with 75% allocated for training and 25% for validation. Both the training and validation sets were drawn from the training dataset. For testing purposes, a new dataset, different from the training dataset, was used.

The Deep Feedforward Neural Network (DFNN) is constructed from a combination of feedforward neural networks (FNN) without feedback connections. The key components of a DFNN include the input layer, one or more hidden layers, and output layer. Each layer comprises multiple neurons, fully connected to the neurons in the subsequent layer [23]. The input layer serves as the entry point for initial data into the network. The hidden layers, situated between the input and output layers, may contain numerous neurons, making the network "deep". Each neuron in the hidden layers processes the output from the previous layer, applies an activation function, and then passes the result to the next layer in a process known as feedforward. The final layer, or output layer, produces the network's prediction based on the processed input data. The detailed architecture and configuration used in this study are presented in Table 2.

Algorithm 1. The proposed method PCSCMI

Input:
- Dataset $D = \{d_1, d_2, \ldots, d_n\}$, where n is the number of data instances.

Output:
- Confusion Matrix (Accuracy, Precision, Recall, F1 Score)

Step 1: Dataset Optimization
1. Remove duplicate data:
   $D' = f_{\text{unique}}(D)$
2. Handle missing values:
   $D'' = f_{\text{null}}(D')$
3. Standardize numerical features:
   - Let D''_{num} be the subset of numerical features in D''. Apply standardization:
     $D_{\text{std}} = \dfrac{D''_{\text{num}} - \mu}{\sigma}$
     where μ is the mean and σ is the standard deviation of each numerical feature.
4. One-Hot Encoding for categorical features

- Let D_{cat} represent the categorical features subset. Apply One-Hot Encoding:
  $D_{\text{encoded}} = \text{OHE}(D_{\text{cat}})$

Step 2: Feature Selection

1. Calculate Pearson correlation using Eq. (1)
   - Add features to set C_p if Pearson correlation coefficient > 0.3
2. Calculate Spearman correlation using Eq. (2)
   - Add features to set C_s if Spearman correlation coefficient > 0.3
3. Calculate Mutual Information using Eq. (3)
   - Add features to set C_m if Mutual Information value > 0.3
4. Combine feature sets
   - Merge sets C_p, C_s, and C_m into feature sets:
     $C_{psm} = C_p \cup C_s \cup C_m$
5. Remove redundant features:
   - Eliminate redundant features from C_{psm} based on high correlation.
6. Combine One-Hot Encoded categorical features:
   - Combine set C_{psm} with categorical features to form the final feature set
     $D_{\text{final}} = C_{\text{psm}} \cup D_{\text{encoded}}$

Step 3: Classification using DFNN

1. Split the dataset:
   - Devide D_{final} into 75% training data D_{train} and 25% validation data D_{val}
2. Define DFNN model architecture
   - Construct a DFNN with three hidden layers (ReLU activation) and an output layer (sigmoid activation):
     $h_i = \text{ReLU}(W_i \cdot D_{\text{final}} + b_i)$
     for i = 1, 2, 3 and:
     $y = \sigma(W_{\text{out}} \cdot h_3 + b_{\text{out}})$
3. Compile the model
   - Configure the model with binary cross-entropy loss function and Adam optimizer

Step 4: Train the DFNN Model

1. Train the model:
   - Train the model on D_{train} for a specified number of epochs.

Step 5: Evaluate the DFNN Model

1. Test the model:
   - Evaluate the model on D_{val} to assess its performance

Step 6: Calculate Confusion Matrix and Evaluation Metrics

1. Predict on testing data:
   - Use the model to make predictions on the testing dataset
2. Calculate the confusion matrix:
   - Compute the confusion matrix based on the prediction results
3. Calculate evaluation metrics:
   - Compute accuracy, recall, precision, and F1-Score from the confusion matrix

Return:

- Confusion Matrix with metrics

Table 1. Features used in the training process.

No.	Feature	No.	Feature
1	ct_dst_sport_ltm	14	proto
2	ct_dst_src_ltm	15	rate
3	ct_src_dport_ltm	16	sbytes
4	ct_state_ttl	17	service
5	dbytes	18	sload
6	dload	19	sloss
7	dloss	20	smean
8	dmean	21	spkts
9	dpkts	22	state
10	dtcpb	23	stcpb
11	dttl	24	sttl
12	dur	25	swin
13	dwin		

Table 2. DFNN architecture and configuration.

Criteria	Values
Model	Sequential
Number of hidden layers	3
Size of input	UNSW_NB15: 25
Number of neurons in hidden layers	50,100,50
Activation function used in the hidden layer	ReLU
Activation function utilized in the output layer	Sigmoid
Learning rates	0.01
Optimizer	Adam
Batch-size	128
Epochs	20

3.4 Dataset overview

UNSW NB-15 stands for University of New South Wales (UNSW) Network Behavior - 15. The UNSW_NB-15 dataset was developed to address the limitations of its predecessors, KDD Cup 1999 and NSL_KDD. This dataset offers more realistic and relevant resources for research in network intrusion detection, aiming to reflect more diverse and contemporary network conditions. Created by the University of New South Wales, the dataset consists of 2 million instances and 49 features that describe various aspects of network traffic. It includes nine labels: Normal, Generic, Analysis, Exploits, DoS,

Fuzzers, Reconnaissance, Shellcode, Backdoor, and Worms [24]. The training dataset comprises 44 features. The dataset utilized in this study is presented in a compact format, specifically named UNSW_NB15_training-set.csv for the training data and UNSW_NB15_testing-set.csv for the testing data. These files contain the necessary information for evaluating the performance of the proposed model [25].

3.5 Evaluation metrics

The metrics used to evaluate intrusion detection models include accuracy, recall, precision, and F1 score. Among these, recall and accuracy are the most utilized evaluation metrics in various studies [26]. Evaluation metrics are derived from the data in the confusion matrix produced during the model evaluation phase. The confusion matrix categorizes data into four categories: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). TP refers to instances where the model correctly classifies an attack as an attack. TN refers to instances where the model correctly classifies normal traffic as normal. FP occurs when the model incorrectly classifies normal traffic as an attack, while FN occurs when the model incorrectly classifies an attack as normal traffic.

Accuracy is characterized as the fraction of instances that were predicted correctly in relation to the overall number of instances within the test dataset. A higher accuracy value indicates that the machine learning model is performing better. Accuracy serves as a useful metric when the training dataset contains balanced class distributions [27]. In intrusion detection, accuracy provides an overall view of how effectively the model distinguishes between normal and suspicious traffic. Thus, accuracy can serve as an initial indicator of model performance. However, in the context of imbalanced data, using accuracy alone can lead to bias or incorrect assessments of the model's effectiveness. Therefore, additional indicators are necessary to develop a reliable intrusion detection model. Accuracy is defined according to Eq. (5).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Precision is defined as the proportion of accurately predicted attacks relative to the total instances identified as attacks. A higher precision value indicates that the model's performance is better, as it means that a larger proportion of predicted attacks are indeed true attacks. This metric is essential in evaluating the model's reliability in accurately identifying attack instances while minimizing false positives. The formulation for precision is given in Eq. (6).

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

A high precision value is particularly advantageous in minimizing false alarms, as it reflects the model's capability to correctly identify true intrusions while reducing the occurrence of false positives. This means the model is effective in distinguishing between legitimate security threats and benign activities, which is critical for maintaining the reliability of intrusion detection systems and ensuring that resources are not wasted on investigating non-existent threats.

True Positive Rate (TPR), also known as Recall, is characterized as the proportion of accurately predicted attacks compared to the total number of actual attacks. A higher Recall value indicates that the machine learning model is more effective, as it means the model correctly identifies a larger proportion of actual attacks. Recall is calculated as shown in Eq. (7).

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

The F1-Score, often called the F1-measure, is the harmonic mean of both precision and recall. A higher F1-Score signifies that the machine learning model performs better, as it reflects a balance between precision and recall, combining both metrics into a single measure of model performance. The formula for the F1-Score is provided in Eq. (8).

$$F1 - Score = 2 \times \frac{precision \times recall}{precision + recall} \tag{8}$$

4. Result and discussion

The proposed model was implemented using Python on Google Colab. For training and evaluating the proposed model, the UNSW-NB15 dataset was utilized. The dataset comprises 257,673 rows, divided into 175,341 for training and 82,332 for testing. The feature selection process resulted in 22 selected features out of 44 evaluated features. Three categorical features were chosen after applying one-hot encoding. Thus, 25 features were included in the next phase, which is the classification process. The details of the features involved in this process can be found in Table 1. The training process yielded

evaluation metrics, including accuracy and loss, as shown in Table 3. The classifier achieved an accuracy of 93.92%, indicating that the feature selection method employed was effective in enhancing the model's performance. Fig. 2 illustrates the accuracy graphs for both training and validation phases. The training accuracy graph shows a gradual increase in training accuracy over the initial epochs, which is typical as the model learns from the data. After a few epochs, the training accuracy appears to stabilize with minimal fluctuations, indicating that the model has achieved relatively high performance. The accuracy then plateaus with no significant improvement, reaching a peak of approximately 0.941 in the final epoch, suggesting that the model has learned sufficiently. The validation accuracy graph also shows an initial increase but does not follow the smooth trajectory of the training accuracy, exhibiting greater fluctuations.

This suggests that the model might be experiencing some variation in the validation data. Such fluctuations may indicate a slight degree of overfitting, where the model has learned too much from the training data and struggles to generalize well to new data. Despite this, the validation accuracy generally tracks the training accuracy, indicating that the model has successfully learned relevant patterns in the validation data. In the final epoch, the validation accuracy approaches 0.940, closely aligning with the training accuracy. Overall, the graph illustrates that both validation and training accuracies remain consistent, indicating stability in the model's performance across different phases of evaluation.

Fig. 3 displays the training and validation loss curves. The training loss is notably high at the first epoch, with a value of 0.140, but decreases significantly as the number of epochs increases. After the initial epochs, the rate of decrease in training loss slows down and stabilizes with minor fluctuations between epochs. The loss then remains relatively constant, reaching 0.115 by the final epoch. This decline indicates that the model improves over time and exhibits minimal prediction errors during training. In contrast to the training loss, the validation loss starts at a slightly lower value of 0.130 and decreases progressively as epochs advance. The fluctuations in validation loss are more pronounced compared to training loss, suggesting that the model may encounter variations in the validation data. The validation loss converges to a value of 0.120 by the end of the training, which is close to the training loss value.

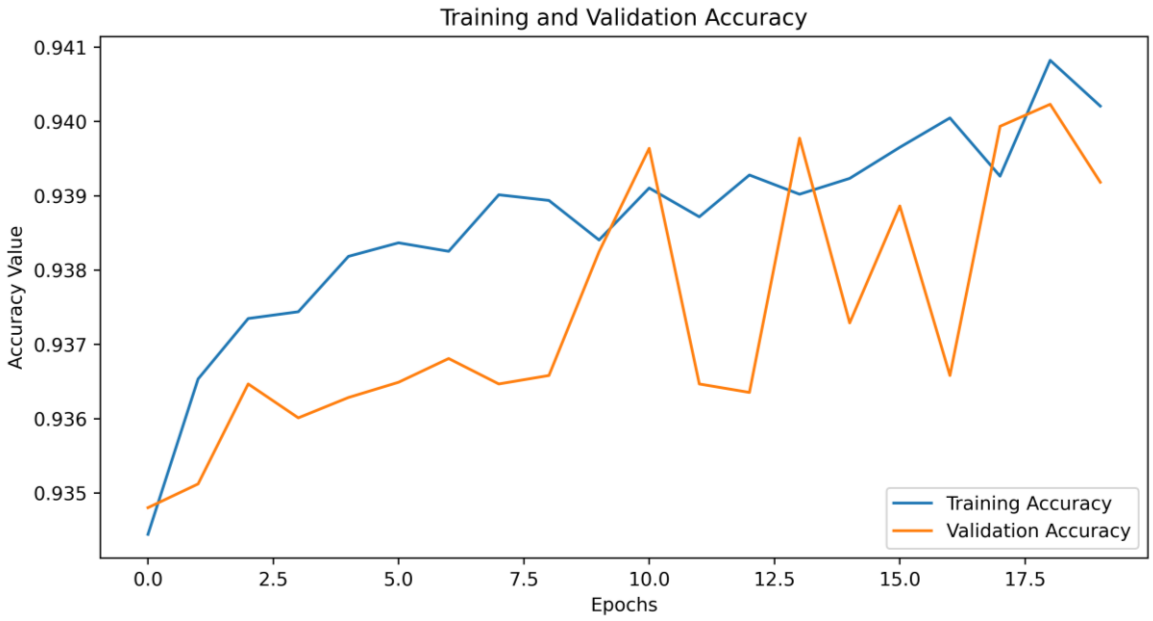

Figure 2. Training and validation accuracy

Table 3. Results of Training and Validation Evaluation

Classification	Feature Selection	Accuracy (%)	Loss	Precision	Recall	F1 Score	Training Time(s)
DFNN	PCSCMI	93.92	0.12	95.32	95.77	95.55	143

Figure. 3 Loss Training and validation

Table 4. Test Evaluation Results

Method	Feature Selection	Accuracy (%)	Loss
DFNN	PCSCMI	89.23	0.22

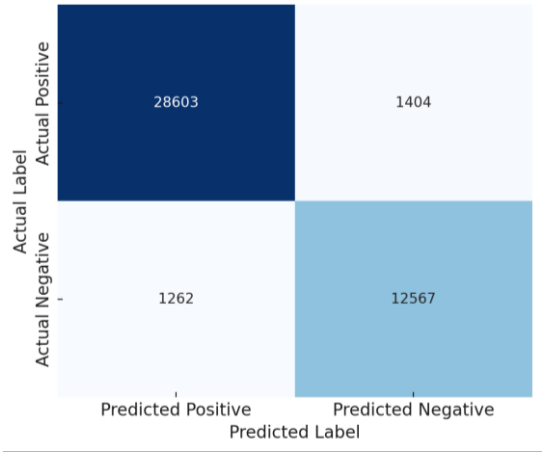

Figure. 4 Confusion matrix

The confusion matrix displayed in Figure. 4 illustrates the model's performance in classifying the test data into four categories. The model successfully classified 28,603 samples as TP, indicating these samples were correctly predicted as positive and truly belong to the positive class. A total of 12,567 samples were classified as TN, representing correctly classified negative samples. Additionally, there were 1,404 FP cases, where negative samples were incorrectly classified as positive, and 1,262 FN cases, where positive samples were incorrectly classified as negative.

Table 4 presents the results of the experiments conducted on the dataset. The table indicates that the accuracy achieved using the proposed feature selection method is 89.23%. This indicates that the method is effective in selecting relevant features, contributing to a significant improvement in classification performance. Table 3 indicates that the execution time of the model is 143 seconds. As explained by [7], there is a direct relationship between energy consumption and execution time longer execution times correspond to higher energy consumption. Thus, the relatively short execution time of the proposed approach suggests that it requires less energy. In conclusion, the proposed method is not only efficient in terms of time but also more energy-efficient compared to other methods that may have longer execution times. Our proposed method was also tested using various combinations to observe the differences in outcomes for each configuration. The results indicate that the combination of all three methods generally outperforms the others, as shown in Figure. 5.

Fig. 6 provides an overview of studies conducted between 2020 and 2024, comparing the proposed method (PCSCMI) with prior studies [10], [18], [8], and [7]], which use Extra Tree, Pearson Correlation, Standard Deviation, and Statistical Selection techniques. The references correspond to those in the

bibliography for easy identification. [10] reported an accuracy of 87.1% when using feature selection with the Extra Tree (ET) algorithm. The ET algorithm successfully selected 22 attributes as optimal features from the UNSW-NB15 dataset. When all features in the dataset (42 features) were used, an accuracy of 87.48% was achieved. Thus, there was a 0.38% decrease in accuracy associated with the feature optimization performed by ET. The use of ET for feature selection proved to be an efficient approach in this study, as it was able to maintain high accuracy with reduced complexity.

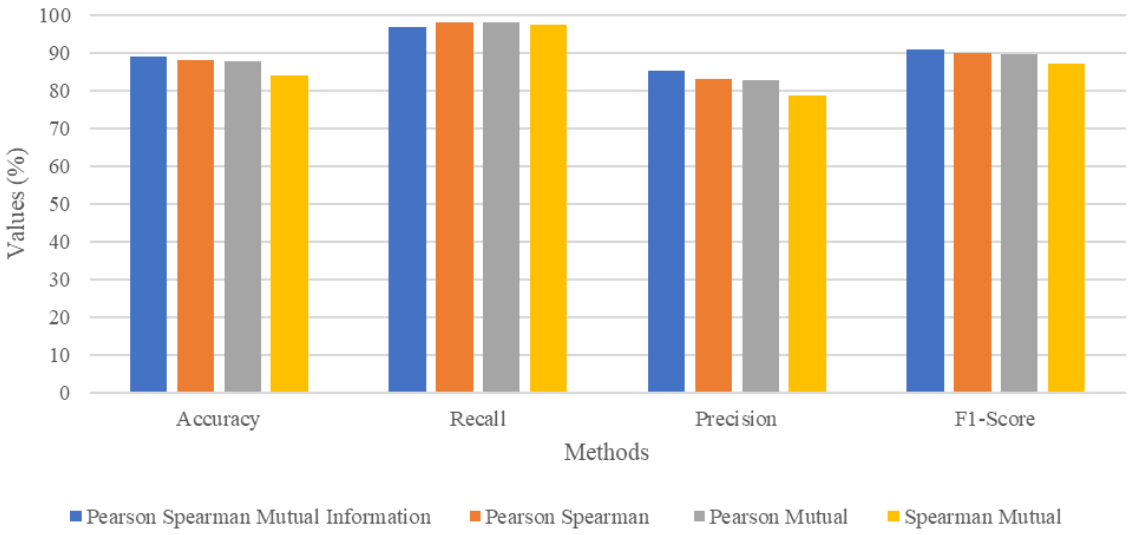

Figure. 5 Combined comparison of statistical methods with proposed methods

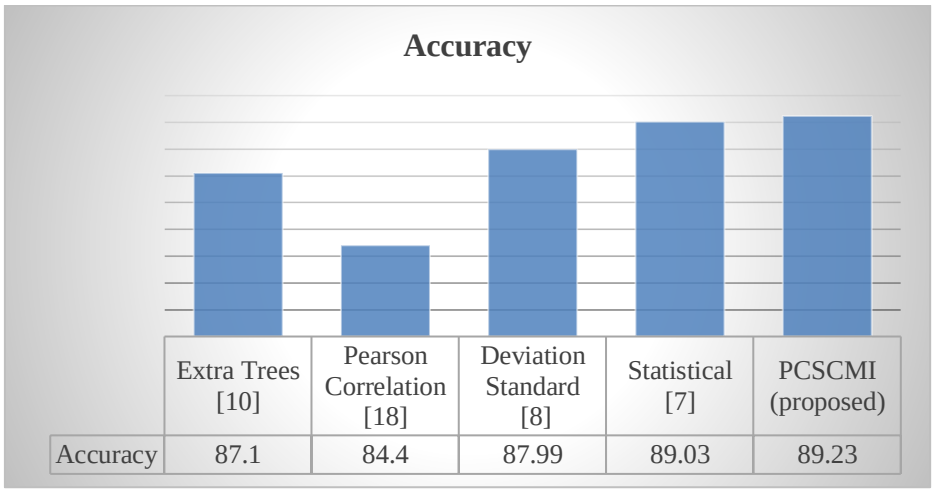

Figure. 6 Comparison of research results with previous research

However, the study also noted that the computational power required for training, validation, and testing the model was significant. Research by [18] applied Pearson correlation to eliminate irrelevant features from the dataset. Testing conducted on the original dataset before performing data balancing yielded an accuracy of 84%. This accuracy is considered sufficient for detecting the presence of attacks or normal traffic, though further improvement to the Pearson correlation method could still be explored.

On the other hand, [8] employed standard deviation and Association Rule Mining methods for feature selection to reduce feature complexity. They used six features in their classifier and achieved an accuracy of 87.99%. Compared to other methods, this study utilized the fewest features while maintaining high accuracy. However, one drawback of this method is the potential loss of important data. By removing features with low standard deviation, the approach may discard features that, while not individually significant, could be crucial collectively in a broader context. Furthermore, some removed

features might hold valuable information about more complex or rare attacks, potentially leading to reduced sensitivity of the model to specific types of attacks, especially those less represented in the dataset.

In a subsequent study, [7] achieved an improved accuracy of 89.03% by using a combination of statistical methods such as standard deviation, mean, and median for feature selection. This combination was used to assess feature relevance and importance, with features exhibiting high deviation and significant mean-median differences deemed more important for learning. However, statistical methods are highly dependent on sample size and data distribution. In cases of small sample sizes or non-normal distributions, the results of feature selection may be biased or unrepresentative. For instance, features that seem important in a small sample may not hold the same significance in a larger population. Another limitation of these statistical methods is their tendency to overlook non-linear interactions between features. These techniques assume that features can be evaluated independently based on statistical properties alone, whereas non-linear interactions can often provide significant predictive value that simple statistical analysis might miss.

Our study, PCSCMI, achieved the highest accuracy of 89.23%, indicating significant potential improvements in the proposed model. This research integrates the strengths of Pearson correlation, Spearman correlation, and Mutual Information methods. Pearson correlation is employed to identify linear relationships between features and the specified labels. Features that exhibit a high correlation with the labels are selected for inclusion in training, as features with high correlation are considered to have a stronger relationship in predicting the desired outcome. The Spearman rho method is used to detect monotonic relationships between features and the target label. Unlike Pearson correlation, which only detects linear relationships, Spearman rho can identify non-linear relationships. One of the advantages of Spearman correlation is its robustness against outliers. In the context of intrusion detection, where data can sometimes be extreme, Spearman correlation proves more reliable in evaluating such data. This capability of Spearman rho compensates for the limitations of Pearson correlation. Finally, while Mutual Information can be used for linear data, it excels in measuring non-linear relationships between features and the target label. Additionally, MI helps avoid redundant features or those that do not provide new information. The advantages offered by the combination of these three methods contribute to the superior results of this study compared to other previously mentioned methods.

5. Conclusion

The strategy proposed in this study intends to boost the performance of intrusion detection systems in Internet of Things (IoT) networks by merging feature selection methods with a Deep Feedforward Neural Network (DFNN). Effective feature selection is essential for reducing data complexity and enhancing computational efficiency in IoT environments. The method combines Pearson Correlation, Spearman Correlation, and Mutual Information to select the most relevant features from large datasets. Only the important features are used in model training, which not only improves intrusion detection accuracy but also reduces complexity, allowing the model to operate faster and more efficiently.

Experimental results show that the proposed method significantly enhances intrusion detection accuracy, achieving a rate of 89.23% on the UNSW-NB15 dataset. Furthermore, removing irrelevant features reduces training time and increases processing efficiency, making this approach more suitable for IoT devices with limited computational power. Although the results are promising, the study acknowledges some limitations. One key limitation is the use of a single dataset for testing, which may restrict the generalizability of the results to other IoT environments. Further work is needed to test the model on diverse datasets and more complex attack scenarios to ensure the approach's broad applicability. Overall, this study makes a significant contribution to optimizing network intrusion detection, both in terms of accuracy improvement and computational efficiency. In the future, this research could be expanded by exploring DFNN parameter optimization and addressing challenges related to real-time implementation. Additionally, further evaluation across multiple standard datasets, such as NSL-KDD, will be conducted to enhance the generalizability and robustness of the proposed method in diverse data environments.

Conflicts of Interest

The authors have no conflicts of interest to disclose

Author Contributions

The author's Contributions are as follows: “Conceptualization, SI, PP and AFR; methodology, SI; software, SI; validation, SI, PP, and AFR; formal

analysis, SI; writing—original draft preparation, SI; writing—review and editing, PP and AFR; visualization, SI; supervision, PP and AFR".

References

[1] H. Mohamed, A. Hamza, and H. Hefny, "An Efficient Intrusion Detection Approach Using Ensemble Deep Learning models for IoT", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, pp. 350–363, 2023, doi: 10.22266/ijies2023.0228.31.

[2] A. E. Cil, K. Yildiz, and A. Buldu, "Detection of DDoS attacks with feed forward based deep neural network model", *Expert Syst Appl*, Vol. 169, 2021, doi: 10.1016/j.eswa.2020.114520.

[3] A. Thakkar and R. Lohiya, "Role of swarm and evolutionary algorithms for intrusion detection system: A survey", *Swarm Evol Comput*, Vol. 53, 2020, doi: 10.1016/j.swevo.2019.100631.

[4] S. P. Thirimanne, L. Jayawardana, L. Yasakethu, P. Liyanaarachchi, and C. Hewage, "Deep Neural Network Based Real-Time Intrusion Detection System", *SN Comput Sci*, Vol. 3, No. 2, 2022, doi: 10.1007/s42979-022-01031-1.

[5] S. Ikhwan, P. Purwanto, and A. F. Rochim, "Comparison Analysis of Intrusion Detection using Deep Learning in IoT Networks", In: *Proc. of 2023 11th International Conference on Information and Communication Technology*, pp. 339–344, 2023, doi: 10.1109/ICoICT58202.2023.10262603.

[6] F. Zare and P. Mahmoudi-Nasr, "Feature Engineering Methods in Intrusion Detection System: A Performance Evaluation", *International Journal of Engineering, Transactions B: Applications*, Vol. 36, No. 7, pp. 1343–1353, 2023, doi: 10.5829/ije.2023.36.07a.15.

[7] A. Thakkar and R. Lohiya, "Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System", *Information Fusion*, Vol. 90, pp. 353–363, 2023, doi: 10.1016/j.inffus.2022.09.026.

[8] L. Zhang, K. Liu, X. Xie, W. Bai, B. Wu, and P. Dong, "A data-driven network intrusion detection system using feature selection and deep learning", *Journal of Information Security and Applications*, Vol. 78, 2023, doi: 10.1016/j.jisa.2023.103606.

[9] H. S. Sharma and K. J. Singh, "A feed forward deep neural network model using feature selection for cloud intrusion detection system", *Concurr Comput*, 2023, doi: 10.1002/cpe.8001.

[10] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper based feature extraction for wireless intrusion detection system", *Comput Secur*, Vol. 92, 2020, doi: 10.1016/j.cose.2020.101752.

[11] R. M. Swarna Priya *et al.*, "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture", *Comput Commun*, Vol. 160, pp. 139–149, 2020, doi: 10.1016/j.comcom.2020.05.048.

[12] C. H. Tseng and Y. T. Chang, "EBDM: Ensemble binary detection models for multi-class wireless intrusion detection based on deep neural network", *Comput Secur*, Vol. 133, 2023, doi: 10.1016/j.cose.2023.103419.

[13] A. Thakkar and R. Lohiya, "Analyzing fusion of regularization techniques in the deep learning-based intrusion detection system", *International Journal of Intelligent Systems*, Vol. 36, No. 12, pp. 7340–7388, 2021, doi: 10.1002/int.22590.

[14] Y. N. Kunang, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, "Attack classification of an intrusion detection system using deep learning and hyperparameter optimization", *Journal of Information Security and Applications*, Vol. 58, 2021, doi: 10.1016/j.jisa.2021.102804.

[15] Y. Lu, S. Chai, Y. Suo, F. Yao, and C. Zhang, "Intrusion detection for Industrial Internet of Things based on deep learning", *Neurocomputing*, Vol. 564, 2024, doi: 10.1016/j.neucom.2023.126886.

[16] K. Albulayhi, Q. A. Al-Haija, S. A. Alsuhibany, A. A. Jillepalli, M. Ashrafuzzaman, and F. T. Sheldon, "IoT Intrusion Detection Using Machine Learning with a Novel High Performing Feature Selection Method", *Applied Sciences (Switzerland)*, Vol. 12, No. 10, 2022, doi: 10.3390/app12105015.

[17] A. Henry *et al.*, "Composition of Hybrid Deep Learning Model and Feature Optimization for Intrusion Detection System", *Sensors*, Vol. 23, No. 2, 2023, doi: 10.3390/s23020890.

[18] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Anomaly based network intrusion detection for IoT attacks using deep learning technique", *Computers and Electrical Engineering*, Vol. 107, 2023, doi: 10.1016/j.compeleceng.2023.108626.

[19] H. Zhu, X. You, and S. Liu, "Multiple Ant Colony Optimization Based on Pearson Correlation Coefficient", *IEEE Access*, Vol. 7, pp. 61628–61638, 2019, doi: 10.1109/ACCESS.2019.2915673.

[20] T.-N. Dao and H. Lee, "Stacked Autoencoder-Based Probabilistic Feature Extraction for On-Device Network Intrusion Detection", *IEEE Internet Things J*, Vol. 9, No. 16, pp. 14438–14451, 2022, doi: 10.1109/JIOT.2021.3078292.

[21] G. Kou, Y. Lu, Y. Peng, and Y. Shi, "Evaluation of classification algorithms using MCDM and rank correlation", *Int J Inf Technol Decis Mak*, Vol. 11, No. 1, pp. 197–225, 2012, doi: 10.1142/S0219622012500095.

[22] S. Gavel, A. S. Raghuvanshi, and S. Tiwari, "Maximum correlation based mutual information scheme for intrusion detection in the data networks", *Expert Syst Appl*, Vol. 189, 2022, doi: 10.1016/j.eswa.2021.116089.

[23] T. K. Gupta and K. Raza, "Optimizing Deep Feedforward Neural Network Architecture: A Tabu Search Based Approach", *Neural Process Lett*, Vol. 51, No. 3, pp. 2855–2870, 2020, doi: 10.1007/s11063-020-10234-7.

[24] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)", In: *Proc. of 2015 Military Communications and Information Systems Conference*, 2015, doi: 10.1109/MilCIS.2015.7348942.

[25] R. Ahmad, I. Alsmadi, W. Alhamdani, and L. Tawalbeh, "A comprehensive deep learning benchmark for IoT IDS", *Comput Secur*, Vol. 114, 2022, doi: 10.1016/j.cose.2021.102588.

[26] Z. Yang *et al.*, "A systematic literature review of methods and datasets for anomaly-based network intrusion detection", *Computers & Security*, Vol. 116, 2022, doi: 10.1016/j.cose.2022.102675.

[27] O. Yousuf and R. N. Mir, "DDoS attack detection in Internet of Things using recurrent neural network," *Computers and Electrical Engineering*, Vol. 101, 2022, doi: 10.1016/j.compeleceng.2022.108034.