ITC 4/51
Information Technology
and Control
Vol. 51 / No. 4 / 2022
pp. 771-785
DOI 10.5755/j01.itc.51.4.31818

Decision Tree with Pearson Correlation-based Recursive Feature Elimination Model for Attack Detection in IoT Environment

Received 2022/07/12 | Accepted after revision 2022/08/31

http://dx.doi.org/10.5755/j01.itc.51.4.31818

HOW TO CITE: Padmashree, A., Krishnamoorthi, M. (2022). Decision Tree with Pearson Correlation-based Recursive Feature Elimination Model for Attack Detection in IoT Environment. *Information Technology and Control*, 51(4), 771-785. http://dx.doi.org/10.5755/j01.itc.51.4.31818

Decision Tree with Pearson Correlation-based Recursive Feature Elimination Model for Attack Detection in IoT Environment

Padmashree, A.
Department of Computer Science and Business Systems, Bannari Amman Institute of Technology,
Sathyamangalam, 638401, India

Krishnamoorthi, M.
Department of Information Technology, Dr. N. G. P. Institute of Technology,
Coimbatore, 641048, India

Corresponding author: apadmashree22@outlook.com

The industrial revolution in recent years made massive uses of Internet of Things (IoT) applications like smart cities' growth. This leads to automation in real-time applications to make human life easier. These IoT-enabled applications, technologies, and communications enhance the quality of life, quality of service, people's well-being, and operational efficiency. The efficiency of these smart devices may harm the end-users, misuse their sensitive information increase cyber-attacks and threats. This smart city expansion is difficult due to cyber attacks. Consequently, it is needed to develop an efficient system model that can protect IoT devices from attacks and threats. To enhance product safety and security, the IoT-enabled applications should be monitored in real-time. This paper proposed an efficient feature selection with a feature fusion technique for the detection of intruders in IoT. The input IoT data is subjected to preprocessing to enhance the data. From the preprocessed data, the higher-order statistical features are selected using the proposed Decision tree-based Pearson Correlation Recursive Feature Elimination (DT-PCRFE) model. This method efficiently eliminates the redundant and uncorrelated features which will increase resource utilization and reduces the time complexity of the system. Then, the request from IoT devices is converted into word embedding using the feature fusion model to enhance the system robustness. Finally, a Deep Neural network (DNN) has been used to detect malicious attacks with the selected features. This proposed model experiments with the BoT-IoT dataset and the result shows the proposed model efficiency which outperforms other existing models with the accuracy of 99.2%.

KEYWORDS: Attack Detection, Internet of Things (IoT), Deep learning, Decision Tree, Recursive Feature Elimination, Deep neural network, BoT-IoT.

1. Introduction

The advanced technologies evolution is recently focused on research that will automate everything using the computer networks connected to the devices. This revolution of the digital industry enhances the human life quality and sends trillion of information through these technologies. The IoT sensors can create a large volume of data that can be processed and transferred in IoT environments such as healthcare, retail, transport, and automotive industries. More industries have researched how the IoT can increases goods and services, business ethics, and organizational changes using Machine and Deep Learning models. The ML and DL approaches increase the reliability, efficiency, and production of the companies with the help of sensors, applications, and programs.

The IoT machine-to-machine communication and person-to-person communication are made through Network packets and protocols. This can have various bugs and flaws that are abused by attackers day by day. The network attackers use this process to make susceptible information and corrupt the devices and resources [20]. If the attackers are stopped by the IoT cybersecurity then it is estimated to lose the companies cost around $90 trillion by the year 2030 [29]. The most normal risk in IoT is malware that is abused by zero-day attacks. The attackers produce the threats to the computer activities using various approaches such as Denial of Service (DoS), Progressive Determined risk (PR), and Decentralized DoS (DDoS). The approaches such as security protocols, access control mechanisms, biometric discovery models, and cryptography are not sufficient to provide protected infrastructure.

With the advancement in network security, attack detection systems are significant that can detect and address all network attacks through advanced algorithms. Some of the researchers reported that 70% of IoT devices are focused on a variety of network threats that make the most of 15 different vulnerabilities, such as encryption and password security. The domain in which IoT is widely used in smart homes, smart transportation, smart cities, smart agriculture, supply chain system, hospital, smart grid system and earthquake detection, and so on [6],[25]. For the malicious agents, IoT application is vulnerable that making the devices of IoT a source of attacks on diverse domains, and making the devices engaged. The wireless medium is used by the IoT devices IoT to transmit data that makes them quickly esteem for an attack. The usual communication threat of the local network is restricted to limited nodes, but IoT attack covers the maximum area and has disturbing effects on IoT [18]. Protected IoT communicationsareneededthe guard against cyber attacks. To the vulnerability of the IoT devices, the security measures become vulnerable.

1.1. Objectives

- To review the recent papers on attack detection models in IoT, and also to define the clear problem definition on the same aspect.
- To find efficient feature selection models to choose the most important and relevant features that can ensure detection accuracy.
- To find an optimal hyperparameter and train the classification model with that parameters to reduce the false prediction rate.

1.2. Motivation

The driving nature of IoT is forced for home automation, modern healthcare, smart cities, and improved manufacturing. The government machinery, businesses, and communities are pushed to form a connected knowledge-based networking system. The IoT policy and elegant advanced approaches are complex in smart homes, healthcare, smart cities, smart transportation, and smart grids. Anomaly detection in IoT is an upcoming research concern. Threats in IoT are increasing research interest with the use of IoT environments in all fields. As the result of multiple protocols addition, thousands of threats are recognized to come out regularly. These attacks are minor variations of previous known cyber attacks. It shows that even with sophisticated approaches like cryptography are hard to identify the tiny modification of these threats within a time. The success rate of using ML and DL approaches in various big data sectors has identified that assistance to cyber security. With the motivation, this approach use ML and DL-based methods for feature selection and classification system to detect attacks in the IoT environment.

1.3. Contribution of the Work

The major contribution of this paper is as follows.

- Preprocessing: the input data is preprocessed to enhance the quality of the dataset using four methods such as data cleaning, log processing, normalization, and one hot encoding approach.
- Feature Selection: An enhanced data processing approach using Decision Tree (DT) with Pearson Correlation-based Recursive Feature Elimination (DT-RFE) is proposed to find the features that also reduce the feature dimension. This proposed model removes the redundant data and also discards the uncorrelated data from the BoT-IoT dataset.
- Feature fusion: the Multimodal feature fusion enhances the system robustness by deciding whether the feature gets weight value or not. This will also increase the system's interpretability. The weights assigned to the selected features have been used for the neural network in the classification phase.
- Classification: Optimized DNN model has been used to learn the different features selected from DTPCRFE to predict the various types of attacks.

The remaining section of this paper is stated as follows: Related work is discussed in Section 2. Section 3 described the dataset with its feature description. Section 4 introduced the proposed system model and methods. Section 5 discussed the experimental and comparative result analysis and Section 6 concludes the proposed model with its future scope.

2. Related Works

This section discusses the recent works of literature on attack detection models in the IoT environment. Kan et al. [10] have explained that in network security, it is significant to detect attacks in IoT networks. In their paper, they proposed an attack detection method for an IoT system based on the Adaptive Particle Swarm Optimization Convolutional Neural Network (APSO-CNN). Their proposed algorithm optimizes the one-dimensional CNN parameters. For the fitness of the composing method, the cross-entropy loss of their obtained trained CNN value is considered. Parthasarathi et al. [23] have explained about decision tree structure and the use of the tree for key management is group communication. From the simulated outcome the effectiveness and reliability of the algorithm are found and the IoT attack detection is exposed. Pecori et al. [24] have explained the involvement of IoT in the daily lives of humans. They demonstrated the detection of traffic in the network and classification of the detected network. In their paper, they introduced a large dataset to detect the traffic in the network. With the help of the deep network model, they examined binary classification and multinominal classification.

Nimbalkarand Kshirsagar [22] have explored various attacks on IoT due to the occurrence of vulnerabilities in devices. They stated that detection of attack is a tedious process for the machine learning (ML) method due to the occurrence of features of traffic in IoT systems. Their paper presented a feature selection for intrusion recognition systems for the exposure of DoS and DDoS attacks. Using the insertion operation and union operation the sunsets of features in the proposed system are obtained. The valuation and verification of their proposed method are performed based on IoT-BoT, and KDD Cup 1999 datasets with a JRip classifier. Atul et al. [5] have exposed that digital transmission is offered an efficient communication stage to share and relocate information. Some of the system challenges they mentioned are security barriers, abnormality, and failure in service. Their paper analyzed and provided a communication pattern using Energy-Aware Smart Home (EASH) framework. The irregularity sources of the announcement standard are differentiated using the method of machine learning. The performance, accuracy, and effectiveness are measured with the help of the composing method

Rahman et al. [27] have proposed an IDS approach namely the Scalable Machine Learning for IoT-Enabled Smart Cities. Their paper addressed the restriction of centralized IDS by proposing semi-distributed and distributed methods. They interconnected efficient feature extraction and feature selection. To allocate the tasks, they developed parallel machine-learning techniques. Their results obtained provide an attack detection accuracy and building time performance. Guet al. [7] have explored that the security and privacy issues in IoT stimulate more and more concentration. They described that IoT attacks are causing incredible defeat to the IoT networks and threatening the safety of humans. They proposed a reinforcement learning-based threat detection model that detects the pattern of attack and its transformation. In their paper, they also explored the IoT traffic features and use entropy-based metrics to predict the attacks in IoT networks.

Krishna and Thangavelu [12] have examined the DoS attack in the IoT system. They demonstrated the security issues and attack that takes place on IoT devices. To detect the attack, they proposed two algorithms a hybrid meta-heuristic lion and a Firefly optimization algorithm (ML-F). They used NSL-KDD and IoT datasets for performing the analysis. Their proposed algorithm attains a maximum performance and classifies the attacks respectively. Lian et al. [31] developed a Decision tree with a Recursive feature elimination-based to choose the features. They used a stacking fusion model to fuse various ML algorithms for the detection of attacks using the NSL-KDD dataset and secured more than 98% of accuracy. Yang et al. [32] developed an IDS system using a knowledge graph and statistical feature selection model. They used CNN and BiLSTM to identify the malicious attacks. The obtained accuracy was 90.01 % using the NSL-KDD dataset.

Sagu et al. [26] developed a hybrid NN model for the detection of attacks in IoT. It combines CNN and DBN to detect attacks. Further, this model is enhanced with the Seagull Adopted Elephant Herding optimization (SAEHO) model to tune the weights for better detection. Anwer et al. [4] evaluated the ML-based approaches for malicious traffic attack detection. It used three approaches Support vector machine (SVM), Gradient boosted decision trees (GBDT), and Random forest (RF) for attack detection in IoT. Among the approaches, Random forest secured improved accuracy of 85.34% using the NSL-KDD dataset. Inayat et al., [8] reviewed the learning-based ML and DL methods for attack detection in IoT. They also discussed the recent research publications and also future research scope of attack detection.

Yadav et al. [30] developed Auto Encoder with DNN based attack detection model to detect the network attacks in 5G connections. The AE model reduces the detection time and improves accuracy, precision, and recall. This model secured 99.76% of accuracy for attack detection.Garagei et al. [1] used both machine and deep learning models such as Decision tree, SVM, KNN, EL, PCA, CNN, AE, RNN, and GAN using various datasets to obtain improved accuracy. Ioannou et al. [9] detect forward and blackhole network attacks using SVM. For experiment analysis, the IoT test bed dataset has been used. The existing research works still lack accuracy and robustness. Feature selection models are to be concentrated which will increase the detection rate of attacks. Hence, in this paper, the Feature re-selection model is proposed and the DL model has been used for classification.

3. Materials and Methods

The BoT-IoT dataset contains normal IoT network traffic with various attacks and it represents the real IoT ecosystem. It is created by the cyber center of new south Wales University in 2018 [11]. The malicious traffic was created with intelligent systems including remotely operated garage doors, smart fridges, intelligent thermostats, and motion-controlled lights. It contains 73000000 instances with 42 features. Each instance is classified as an attack or normal. The instances are categorized into four attacks such as DoS, DDoS, reconnaissance, and intelligence stealing. Some of the superfluous features are removed and the remaining features and attacks are listed in Tables 1-2.

Table 1
BoT-IoT superfluous feature set

Feature Number	Feature	Description	Data type
f1	_pkSequence ID	Row_identifier	Int
f2	_Start time	Record_Start_Time	Float
f3	_Flags	Flow state flag	Categorical
f4	_Protocol	Protocol textual representation	Categorical
f5	_Source port	Port number of the source	Categorical
f6	_Destination port	Port number of the destination	Categorical
f7	_Packets	Total number of packets in a transaction	Int
f8	_State	State of the transaction	Categorical
f9	_Dur	The total duration of the record	Float
f10	_Mean	Aggregated record avg duration	Float
f11	_Dpkts	Count of destination to source	Int
f12	_Sbytes	Byte count from source to destination	Int
f13	_Dbytes	Byte count from destination to source	Int
f14	_TBpSIP	Total No. of bytes per source IP	Int
f15	_TPPProto	Total No. of packets per protocol	Int
f16	_ARPProto PSrcIP	Avg rate per protocol per source IP	Float
f17	_ARPProto PDport	Avg rate per protocol per destination port	Float
f18	_PktsPStateP ProtocolPSrcIP	No. of packets grouped by the state of flows and protocols per source IP	Integer
f19	_Srate	A packet of the source to destination per second	Float

Table 2
Attacks with instances in the BoT-IoT dataset

BoT-IoT	Category	Total number of instances
Normal		9658
Attacks	DoS	254659
	DdoS	12123565
	Reconnaissance	1593444
	Information theft	1654

4. Proposed Methodology Design

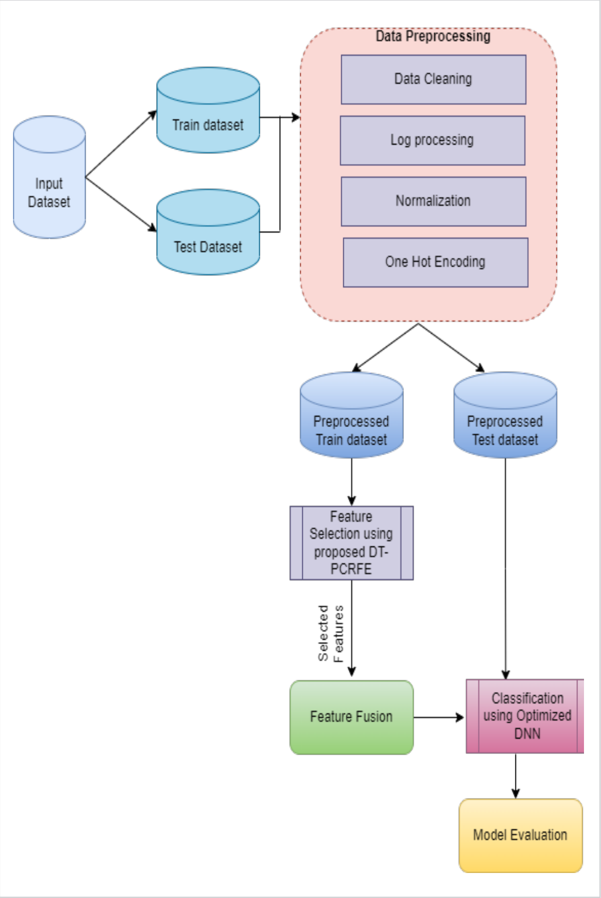
This section presents the overall architecture of the proposed attack detection model for IoT and discusses preprocessing feature selection, and detection methodologies.

4.1. System Architecture

The overall architecture of the proposed model is shown in Figure 1. This Attack detection system consists of four phases such as (i) preprocessing (ii) feature selection (iii) feature fusion and (iv) classification of attack detection. Data preprocessing will improve the dataset quality using the approaches such as data cleaning, log processing, normalization, and one hot encoding. The feature selection phase extracts acts the important and relevant features using the proposed DT-PCRFE approach. Feature fusion enhances the system's robustness by deciding whether the feature gets weight value or not. This will also increase the system's interpretability. This phase also covers the features as vectors for further processing. Next, the DNN classifier is used to detect the malicious requests or attacks from the IoT device requests using the extracted features

Figure 1

Overview of Proposed Attack Detection model for IoT



4.2. Data Preprocessing

The initial input data are difficult to process due to the large volume of network traffic to detect the attacks with all the features which have different forms either numeric or non-numeric data. to solve the constraint issues of non-numerical features and enhance the quality of the datasets, this paper used four approaches for preprocessing which include data cleaning, log processing, Normalization, and one-hot encoding [15].

Data cleaning removes the redundant data by detecting duplicate values. For both training and testing datasets, the data cleaning is performed. The symbolic features are converted into numerical features since ML and DL approaches are working on real number vectors [13]. The feature label is categorized into normal and abnormal network traffic. It is considered an attack and it is classified as DoS, DDoS, Reconnaissance, and Information Theft. Log processing efficiently reduces the difference between the features of the data. The features that have the larger values are treated as an outlier that will affect the system performance. Hence, the log function denoted in Equation (1) is executed to reduce the dimensionality that transforms the feature value to the same granularity value.

$$D_i' = \log(f_i), \qquad (1)$$

where, f_i= feature i. next, the feature normalization will convert feature value with the suitable range that will reduce the data imbalance and larger value preference issues. First, from the python scikit learn, the mapping process is applied to the dataset to map the symbols into a unique numeric value. Normalization is the key step to representing the values of the data within the same range for optimal feature selection. In this work, min-max normalization is used to fill the gap between the values in the range 0 to 1 and the Equation(2) denoted by the normalization process.

$$D'_{ij} = \frac{X_{ij}-\min(f_i)}{Max(f_i)-Min(f_i)}, \quad (2)$$

where, i – feature, j – a record of the dataset and $Max(f_i)$, $Min(f_i)$ are the maximum and minimum values of the features respectively. Hence, all the continuous value features are mapped into the range of values between [0,1] providing the importance of each feature. Finally, one hot encoding is to convert the categorical data to unique values that assign the current category value as bit 1 and the other as 0. This will improve the DL model with better input vectors [16]. For example, the DoS label features are converted to the form of [1,0,0,0,0].

These transformed datasets are then given as input to the feature selection process using the proposed Decision tree with the Pearson Correlation RFE model.

```
Algorithm 1: Preprocessing
Input: Data D with features $f_1, f_2 \ldots f_n$
Output: transformed features $D' = f'_1, f'_2 \ldots f'_n$
Step 1: for i=1 to n
Step 2: if ($f_i$ = = symbol) then
Step 3: Apply python Scikit learn to map the symbols into numeric
Step 4: Data cleaning, Log processing using Equation (1), Normalization using Equation (2) and One-hot encoding of categorical features.
Step 5:else
Step 6:Data cleaning, Log processing using Equation (1), Normalization using Equation (2) and One-hot encoding of categorical features.
Step 7:End if
Step 8: End for
```

4.3. Feature Extraction Using Proposed Decision Tree with PCRFE (DT-PCRFE)

Once the data preprocessing is over, it is difficult to give input directly to the learner due to the high dimensionality. Therefore, it is important to select relevant features to train the ML and DL models. The good feature selection approach selects the most relevant features which will improve the detection accuracy of the classifier. In this paper, a novel feature selection model called DT-PCRFE has been proposed. The PCRFE is an iteratively building model and selects the relevant as well as irrelevant features based on the coefficients. The recursive process is continued for all the features and selected features are grouped as a vector for further processing and the remaining features are eliminated. If the relationship between the feature and its response variable is nonlinear then tree-based methods are used which do not requires a much-debugging process. In this work, the decision tree model has been used for this purpose.

Decision Tree (DT) used the information entropy index for its feature selection. The tree computes the information entropy of the data and split it layer by layer. At last, each instance is separately divided. Entropy is a measure to determine the ambiguity of the random feature. For example, D is the random variable having a limited number of values and the probability distribution of D is denoted in Equation (3)

$$P(D = f_i) = p_i, \quad (3)$$

where ith feature f_i is corresponding to the probability p_i as one by one. The random variable D, entropy is computed as in Equation (4)

$$H(D) = -\sum_{i=1}^{n} p_i \log p_{i.} \quad (4)$$

While the entropy is greater, then it is not much difficult to search the variable ambiguity D is greater. That is, the probability of this value is less than 1 and the logarithm of the probability is less than 0. The minus sign in the formula will frustrate the value which is negative that is produced by the log function. While the p_i difference corresponds to f_i is greater than H is also greater. The joint probability distribution of two random variables D and G is denoted as in Equation (5)

$$P\left(D = f_i, G = g_j\right) = p_{ij,} \quad (5)$$

where f and g are the ith and jth features. The conditional entropy H (G|D) is the ambiguity of the random variable G under the condition D and it is computed as in Equation (6)

$$H(G|D) = \sum_{i=1}^{n} p_i \, H(G|D = f_i). \quad (6)$$

The information gain (IG) of A(feature) of X (training dataset) is denoted as IG(X|A) and it is computed as in Equation (7)

$$IG(X|A) = H(X) - H(X|A). \quad (7)$$

The IG represents the degree of inaccuracy of G information minimized after the feature D information is educated. Using IG (X|A) as a feature for dataset partition may cause the issue of selecting the features with more values. Hence, the IG is used here to correct the stated issue as in Equation (8)

$$IG_R = \frac{IG(X|A)}{H(X)}. \tag{8}$$

Suppose, the decision tree (DT) number of leaf nodes is |DT| and the leaf node is denoted as t, the number of samples in the node is N_t and the no. of sample points of k is N_k. The leaf node entropy is H_t and penalty term $\rho \geq 0$ is an optional parameter. The decision tree DT loss function L is denoted as in Equation (9)

$$L_\rho(DT) = \sum_{t=1}^{|DT|} N_t H_t(DT) + \rho(DT), \tag{9}$$

where the entropy calculation is in Equation (10)

$$H_t(DT) = -\sum_k \frac{N_{tk}}{N_t} log \frac{N_{tk}}{N_t} \tag{10}$$

The loss function is responsible to find the difference quantity between the actual and predicted value. The learning goal is to minimizes the loss function. So, in Equation (9), the first part is rewritten as,

$$R(T) = \sum_{t=1}^{|DT|} N_t H_t(DT) = -\sum_{t=1}^{|DT|} \sum_{k=1}^{K} N_{tk} log \frac{N_{tk}}{N_t}. \tag{11}$$

The loss function is simplified as,

$$R_\rho(DT) = R(DT) + \rho(DT), \tag{12}$$

where R(T)- training data prediction error and |DT| is the complexity of the model. The penalty term ρ balances the complexity and prediction error of the model. Next, PCRFE is executed on various feature combinations. With the calculation of sum of its result coefficients, the score value of important feature is allotted and the best feature is added into the subset. Pearson Correlation (PC) is the association among the data within the range between -1 and 1 where +1 indicates the positive correlation , 0 indicates no correlation and -1 indicates negative correlation of the data. while Compares to the existing ML based feature selection models [17], at each step it remove features various in PCRFE, irrelevant features are removed at once. The features Correlation Coefficient is calculated using the Equation (13).

$$PCorr_{fi,gi} = \frac{\sum_{i=1}^{n}(f_i - \bar{D})(g_i - \bar{G})}{\sqrt{\sum_{i=1}^{n}(f_i - \bar{D})^2}\sqrt{\sum_{i=1}^{n}(g_i - \bar{G})^2}}, \tag{13}$$

where f_i, g_i – features for correlation that is in consideration. This result is close to the interval -1 and 1. The value closer to -1 or 1 relates the strong relationship of two features and 0 relates the weak relationship of two features. Then, threshold value has been used to rank the correlated features. The features with least amount of rank will be removed. The feature removal-computation is denoted as Equation (14).

$$PCRFE(f_i) = \sum_{i=1}^{n}\left(g_{i,j} - \sum_{j=1}^{d} PCorr_{f_i,g_j} \times f_i\right)^2. \tag{14}$$

Next, this paper introduced a feature fusion in feature extraction phase. The multimodal feature fusion approach efficiently combines the features selected from the feature selection process and assigns weights to the features which will useful for the neural network in classification process. The step by step procedure is stated in Algorithm 2. The feature pair are selected as input for processing. Decision is trained which compute the ranking criterion. For all the feature pair in the dataset, the correlation is computed and based on the threshold the correlated features are selected and added to the subset R. The non-correlated features are removed from the set. For selected features, weight set is created by assigning weight parameters to each feature. If the feature belongs to the selected feature then the weight is multiplied with optional parameter called. Or else the weight does not change. This weight parameter can be used for classification process.

Algorithm 2: (DT-PCRFE Feature selection and Feature Fusion)

```
Input: preprocessed data set D, dataset size N and number of features n, feature pair S = {$f_i$, $g_i$}.
Output:Selected feature subset $R = \{f_i, i = 1, 2, \dots r\}$ with weights $W = \{w_i, i = 1, 2, \dots r\}$
Step 1: Initialize $W = \varnothing$ and feature order set $R = \varnothing$
Step 2: Fori =1 to N
Step 3: Decision tree is trained and calculates the ranking criterion
Step 4: For each $< f_i, g_i > \in S$ do
Step 5: correlation coefficient of the feature is computed using the Equation (13)
Step 6: remove the features using the PCFSR Equation (14)
Step 7: if ($PCFSR(f_i) \geq threshold$) then
Step 8: Add the features into the subset R
Step 9: Else
Step 10:remove the features
Step 11: End if
Step 12: End for
Step 13: for each $f_i \in S$ do
Step 14: Weight calculation as $\begin{cases} w_i \quad if\ f_i \notin R \\ \rho \times w_i\ if\ f_i \in R \end{cases}$
Step 15:  End for
Step 16: Output the feature set R with Weight W
```

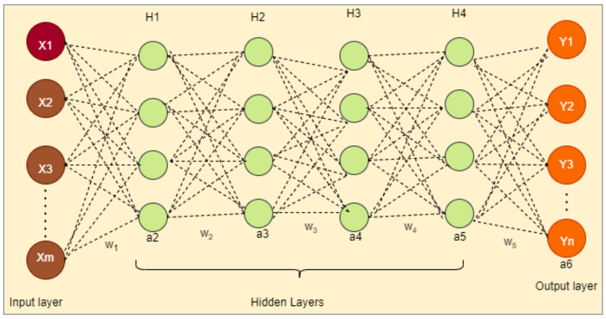
Using the proposed feature selection model, the dataset original feature set is reduced and the most important features are selected. Among the 42 features, the redundant and superfluous features are removed in preprocessing phase. With that reduced feature set of 19 features stated in Table 1, the most important and relevant features such as R={f3, f4, f7, f8, f12, f13, f16, f17 and f18} as a total of nine features has been selected for further processing. These selected feature subset is produced as input to the classifier for attack detection.

4.4. Classification – Optimized DNN

This section discussed about the classification model called Deep neural network with optimized hyper parameter settings for the detection of attacks in IoT. Initially, the hyper parameters such as learning rate, epoch size, momentum, batch size, and dropout regularization and so on are selected to improve the model performance. Random Search approach has been used to select the hyper parameters for DNN. At each instance, the model is trained with the parameters selected by the random search. It can improve the model performance after fixed number of iterations execution. Deep Neural network is a kind of Artificial Neural network (ANN) with more hidden layers. Each DNN have input layer, multiple hidden layers and one output layer. Each hidden layer consists of more neurons. Based on the received inputs, each neuron is fired or retained.

Figure 2

Architecture of Optimized DNN



The DNN architecture is shown in Figure 2. The proposed model consists of one input layer, four hidden layers and one output layer. X denotes the input feature, w indicates the weight of the link between neurons from Layer i to Layer i+1, Y is the output and a is an activation function which is used to fire the neuron based on the forward propagation computation. Each layer use different activation function for better computation. The hidden layer implements ReLu activation function and the output layer uses softmax activation function defined in Equation (15) and Equation (16)

$$ReLU(a_i) = \max(0, a_i) \tag{15}$$

$$Softmax(a_i) = \frac{e^{a_i}}{\sum_{j=1}^{n} e^{a_i}}, \tag{16}$$

where, a_i - obtained output from neuron i in the output layer and n – number of classes in the output layer.

The DNN comprised of two stages such as forward propagation and backward propagation. In forward propagation, the inputs are multiplied with the weights and bias which is assigned to each neuron travel towards hidden layer. The final predicted output is Y. Each hidden layer ‘L’ neuron calculates the following

$$a_l = w_l^L . H^{l-1} + b^l \tag{17}$$

$$H_l = a(H_l), \tag{18}$$

where a – activation function, H - hidden layer, w – weight and b – bias. The DNN has been trained by backpropagation which employs gradient descent (GD)method for its weight updation. This will reduce the error between actual and predicted results. The gradient calculation computes the changes in the weight with respect to its expected output. The error between predicted and actual output stated in the output layer is computed and backpropagated to the preceding hidden layers. Based on gradient values, the weight and bias are updated. The GD method is optimized using Adam optimizer which is the combination of gradient descent with momentum and RMS (Root Mean Square) prop approach. In Momentum approach, the velocity and the gradient is calculated and RMSP use weighted average method on second gradient moment (dw2). The Adam optimizer employs both past squared gradient (U), past momentum (V) computed using Equation (19) and (20). The bias is added to U and V using Equation (21) and (22) and the weights are updated using Equation (23)

$$U = \beta_1 V + (1 - \beta_1)dw \quad (19)$$

$$V = \beta_2 V + (1 - \beta_2)dw2 \quad (20)$$

$$U = \frac{U}{1-\beta_1^i} \quad (21)$$

$$V = \frac{V}{1-\beta_2^i} \quad (22)$$

$$w' = w - \alpha \frac{U}{\sqrt{V}+\epsilon}, \quad (23)$$

where α - learning rate and β - average parameter [0,1]. The DNN is trained and tested using the BoT-IoT dataset with the hyper parameters selected from the optimization approach.

5. Experimental Results and Discussions

The proposed efficient feature selection based attack detection model is implemented using Scikit learn library of python. This section discusses about the experimented results using the evaluation metrics such as Accuracy, Recall, Precision, Recall and AUC.

5.1. Evaluation Metrics

The Proposed Model attack detection results is evaluated in terms of four categories includes True_Positive (TP), True_Negative (TN), False_Positive (FP) and False_Negative (FN) which is denoted as a confusion matrix [27] as shown in Table 3.

Table 3
Confusion matrix

Actual class	Predicted class	
	Normal	Attack
Normal	True_negative (TN)	False_positive (FP)
Attack	False_negative (FN)	True_positive (TP)

True_Positive (TP): The network correctly detects the number of instances belongs to desired class.

True_Negative (TN): The network incorrectly detects the number of instances that are not belongs to the desired class.

False_Positive (FP): The network not detects the number of instances that are belongs to the desired class.

False_Negative (FN): The network correctly detects the number of instances are not belongs to the desired class.

$$Accuracy(\%) = \frac{TN+TP}{TN+FN+TP+FP} \quad (24)$$

$$Precision = \frac{TP}{TP+FP} \quad (25)$$

$$recall\ (or) Attack\ detection\ rate(ADR) = \frac{TP}{(TP+FN)} \quad (26)$$

$$F - Measure = \frac{2*Precison*Recall}{Precision+Recall} \quad (27)$$

$$FalsePostiveRate\ (FPR) = \frac{FP}{FP+TN} \quad (28)$$

$$AUC = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) \quad (29)$$

LogLoss: It measures the accuracy of the method with the probabilistic value as output. The log value of 0 is the perfect and it will increase as per the likelihood of the differed real label.

Training time: The a1mount of time needed to build the classification model.

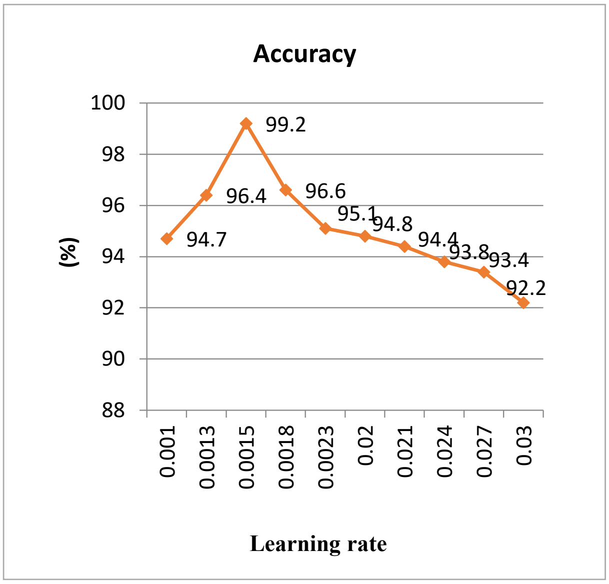
5.2. Hyper Parameter Settings of Proposed Attack Detection Model for IoT

The proposed efficient feature selection with DL based attack detection model is evaluated with ReLU and Soft max activation functions with the learning rate from 0 to 1 and number of epoch.The learning rate is important and highest range leads loss and lowest range leads overfitting issues.

Hence, the proposed model is evaluated in terms of various range of learning rate and best learning rate is fixed based on the detection result. Figure 3 shows the execution of proposed model with different learning rate and improved accuracy of 99.2% is secured with the learning rate as 0.0015. While increasing the learning rate, the accuracy get decreased. Hence, the optimal learning rate for our model is 0.0015. in terms of number of epochs, at the maximum of 100 epochs, the model gets training loss. After 60 epochs, the loss is not changed. Therefore, the epoch is set as 60 for our proposed model.

Figure 3

Accuracy changes based on various Learning rate



5.3. Result Analysis

The confusion matrix for the proposed model is show in Table 4. Based on these, the metrics are evaluated and the results of the proposed attack detection model are shown in Table 5 using BoT-IoT datasets. The secured results show the effectiveness of proposed model which is efficient in detecting the attacks in IoT environment.

Table 4

Confusion matrix evaluation

Actual class	Predicted class	
	Normal	Attack
Normal	99.2	0.8
Attack	1.1	98.9

Table 5

Evaluation of proposed Attack Detection model

Datasets	BoT-IoT	
TP rate	0.989	0.0989
FP rate	0.01	0.001
Precision	0.984	0.991
Recall	0.991	0.987
F-Measure	0.991	0.985
ROC	0.993	0.992
Class	Attack	Normal

Table 6 demonstrates the proposed model evaluation results in terms of accuracy, false prediction rate, log loss and training time. The proposed model secured the accuracy of 99.2% and false positive rate of 99.5%. It also obtained reduced log loss and training time

Table 6

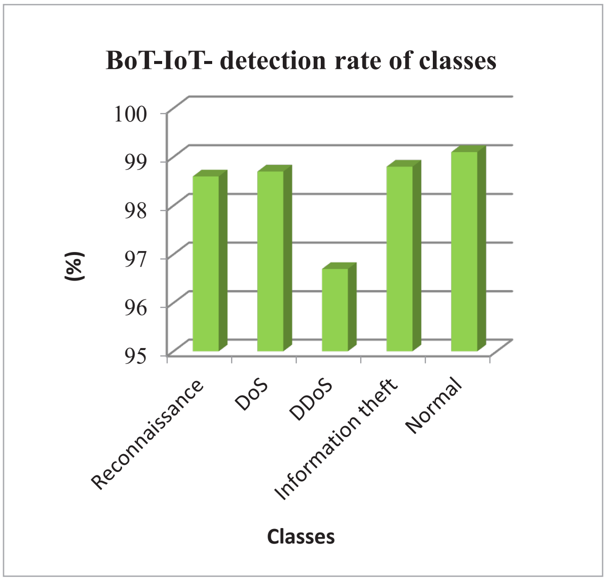
Proposed model evaluation in terms of accuracy and log loss

Datasets	Accuracy	FPR	Log loss	Training time (ms)
BoT-IoT	99.2	99.5	0.091	149

For the considered dataset, Figure 4 illustrates the detection rate of the records that are belongs to the attack classes such as Reconnaissance (98.6%), DoS (98.7%), DDoS (96.7%), Information theft (98.8%) and Normal (99.1%) which proves that the proposed model is efficient on detecting the attacks and normal classes.

Figure 4

Detection rate of BoT-IoT dataset classes



5.4. Comparative Analysis of Proposed Attack Detection Model with Conventional Systems

The performance of the proposed model DTP-CRFE-ODNN is compared with the various feature selection model such as Wrapper based Neuro Tree [29], Knowledge graph [31], Improved Principal Component Analysis [27] and Modified kNN [21]. The analyzed results are shown in Table 7.

The proposed efficient feature selection model secured the accuracy of 99.2% which is higher than the traditional feature selection models. The measures are computed in both training and testing phase. There is a slight variance in testing and training phase results. Compared to traditional approaches, the proposed model secured improved accuracy, false prediction rate, F-score, recall and precision values.

Likewise, the proposed feature selection with classification model is compared with other classifiers such as CNN-BiLSTM [31], CNN-DBN [26] and Auto Encoder with DNN [8] in order to prove the efficiency of the complete system. The results are compared in terms of detection rate and AUC and results are shown in Figures 5-6. The proposed model secure the detection rate and ROC as 99.1% and 0.99 respectively which is optimum, best compared to other detection systems.

The existing approaches such as CNN-BiLSTM secured 98.3% and 0.976 as detection rate and ROC, CNN-DBN obtained the detection rate as 98.6% and

Table 7

Comparative analysis of proposed vs existing algorithms

Feature selection model	BoT-IoT					
	Phase	Accuracy (%)	FPR (%)	F-Score (%)	Recall (%)	Precision (%)
Wrapper -Neurotree	Training	93.31	1.81	95.62	94.53	92.83
	Testing	93.12	1.72	95.18	94.34	92.78
Knowledge Graph	Training	96.41	1.63	96.72	96.46	94.81
	Testing	96.71	1.62	96.61	96.51	94.53
Improved PCA	Training	96.42	1.91	97.13	96.72	95.91
	Testing	96.81	1.98	97.25	97.67	95.83
Modified kNN	Training	94.21	1.53	96.21	96.31	94.91
	Testing	94.39	1.54	96.36	96.06	94.29
Proposed DT-PCRFE	Training	99.01	1.02	98.82	98.91	98.62
	Testing	99.19	1.01	98.91	98.62	98.46

Figure 5
Detection rate comparison of Proposed vs Traditional algorithms

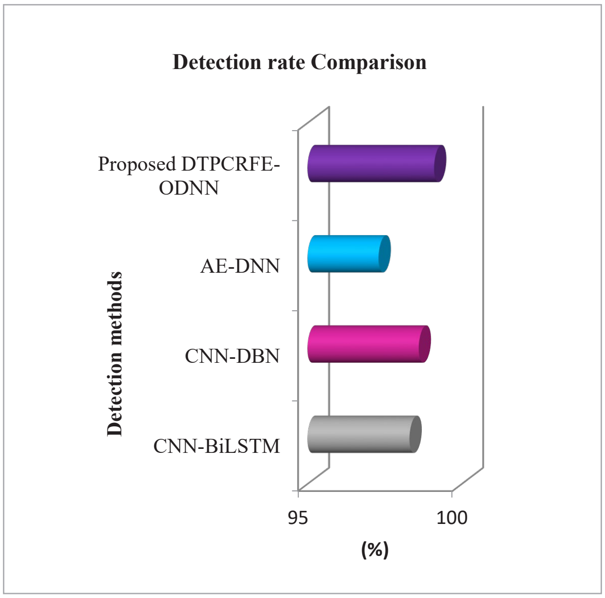
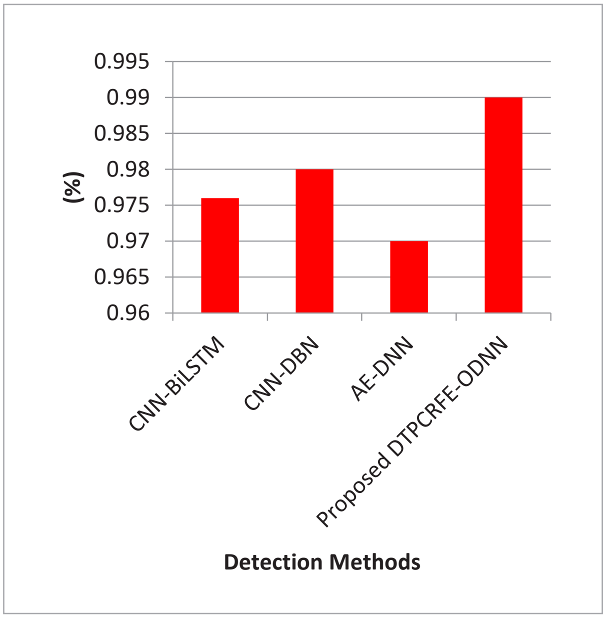


Figure 6
AUC-ROC comparison of proposed vs Existing approaches



ROC as 0.98 and AE-DNN secured the detection rate of 97.3% and ROC as 0.97. Due to the efficient implementation of feature selection and feature fusion model with optimized DNN, the proposed attack detection system performance is superior to other attack detection systems and it is suggested to identify the attacks in the IoT environment.

5. Conclusion

The security threats to IoT-enabled systems suffered from severe security risks due to the inherent characteristics of advanced technologies. These characteristics make the IoT environment efficient, functional, and versatile but it is vulnerable to the threat to use the information for the wrong purpose. This paper introduced an efficient feature selection and detection system for IoT environments using ML and DL approaches. Initially, the input data is preprocessed using four approaches such as data cleaning; Log processing, Normalization, and One-hot encoding to make the data balanced for further processing. Second, the efficient feature selection model using Decision Tree with Pearson Correlation based Recursive Feature Elimination has been proposed which selects the most relevant and correlated features. The feature fusion approach will assign a weight to the selected features which will use for neural network training at the initial stage. The proposed model selects nine relevant features from the BoT-IoT dataset. Next, an optimized Deep Neural network (DNN) has been used with the selected number of features for the detection of attacks in the BoT-IoT dataset. The proposed model is evaluated in terms of the evaluation metrics and compared with the conventional feature selection and classification system to prove the proposed system performance. Using the BoT-IoT dataset, the proposed model secured 99.2% of accuracy on the detection of attacks and normal transmission in IoT environment which shows the efficiency and effectiveness of the proposed model. In future, the proposed feature selection model is enhanced with hybrid classification system optimized by swarm intelligence approaches and edge computing will be introduced to enhance energy and resource utilization of the attack detection system.

References

1. Al-Garadi, M. A., Mohamed, A., Al-Ali, A. K., Du, X. Ali, I., Guizani, M. A Survey of Machine and Deep Learning Methods for Internet Of Things (IoT) security,IEEE Commun. Surv.2022, 22, 1646-1685. https://doi.org/10.1109/COMST.2020.2988293
2. Ali, M. H., Jaber, M. M., Abd, S. K., Rehman, A., Awan, M. J., Damasevicius, R., Bahaj, S. A. Threat Analysis and Distributed Denial of Service (DDoS) Attack Recognition in the Internet of Things (IoT), Electronics, 2022, 11(3). https://doi.org/10.3390/electronics11030494
3. Alzaqebah, A., Aljarah, I., Al-Kadi, O. Damasevicius, R. A Modified Grey Wolf Optimization Algorithm for an Intrusion Detection System. Mathematics, 2022, 10(6). https://doi.org/10.3390/math10060999
4. Anwer, M., Khan, S. M., Farooq, M., Waseemullah, U. Attack Detection in IoT Using Machine Learning. Engineering, Technology & Applied Science Research, 2021, 11(3), 7273-7278. https://doi.org/10.48084/etasr.4202
5. Atul, D. J, Kamalraj, R., Ramesh, G., Sankaran, K. S., Sharma, S., Khasim, S. A Machine Learning-based IoT for Providing an Intrusion Detection System for Security. Microprocessors and Microsystems, 2022, 82, 103741. https://doi.org/10.1016/j.micpro.2020.103741
6. Brun, O., Yin, Y., Gelenbe, E., Kadioglu, Y. M., Augusto-Gonzalez, J., Ramos, M. Deep Learning with Dense Random Neural Networks for Detecting Attacks Against IoT-connected Home Environments. In International ISCIS Security Workshop, Springer, Cham, 2018, 79-89. https://doi.org/10.1007/978-3-319-95189-8_8
7. Gu, T., Abhishek, A., Fu, H., Zhang, H., Basu, D., Mohapatra, P. Towards Learning-automation IoT Attack Detection Through Reinforcement Learning. In 2020 IEEE 21st International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), IEEE, 2020, 88-97. https://doi.org/10.1109/WoWMoM49955.2020.00029
8. Inayat, U., Zia, M. F, Mahmood, S., Khalid, H. M., Benbouzid, M. Learning-Based Methods for Cyber Attacks Detection in IoT Systems: A Survey on Methods, Analysis and Future Prospects. Electronics, 2022, 11, 1502. https://doi.org/10.3390/electronics11091502
9. Ioannou, C., Vassiliou, V. Experimentation with Local Intrusion Detection in IoT networks Using Supervised Learning. In Proceedings of the 16th International Conference on Distributed Computing in Sensor Systems (DCOSS), Marina del Rey, CA, USA, 2020, 423-428. https://doi.org/10.1109/DCOSS49796.2020.00073
10. Kan, X., Fan, Y., Fang, Z., Cao, L., Xiong, N. N., Yang, D., Li, X. A Novel IoT Network Intrusion Detection Approach Based on Adaptive Particle Swarm Optimization Convolutional Neural Network. Information Sciences, 2021, 568, 147-162. https://doi.org/10.1016/j.ins.2021.03.060
11. Koroniotis, N., Moustafa, N., Sitnikova, E., Turnbull, B. Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset, arXiv preprint arXiv:1811.00701
12. Krishna, E. S., Thangavelu, A. Attack Detection in IoT Devices Using Hybrid Metaheuristic Lion Optimization Algorithm and Firefly Optimization Algorithm. International Journal of System Assurance Engineering and Management, 2021, 1-14. https://doi.org/10.1007/s13198-021-01150-7
13. Kumar, K., Kumar, A., Kumar, N., Mohammed, M. A., Al-Waisy, A. S., Jaber, M. M., Shah, R., Al-Andoli, M. N. Dimensions of Internet of Things: Technological Taxonomy Architecture Applications and Open Challenges-A Systematic Review. Wireless Communications and Mobile Computing, 2022. https://doi.org/10.1155/2022/9148373
14. Lakhan, A., Mastoi, Q. U. A., Elhoseny, M., Memon, M. S., Mohammed, M. A. Deep Neural Network-based Application Partitioning and Scheduling for Hospitals and Medical Enterprises Using IoT Assisted Mobile Fog Cloud. Enterprise Information Systems, 2022, 16(7), 1883122. https://doi.org/10.1080/17517575.2021.1883122
15. Lakhan, A., Mohammed, M. A., Elhoseny, M., Alshehri, M. D., Abdulkareem, K. H. Blockchain Multi-objective Optimization Approach-enabled Secure and Cost-efficient Scheduling for the Internet of Medical Things (IoMT) in Fog-Cloud System. Soft Computing, 2022, 1-14. https://doi.org/10.1007/s00500-022-07167-9
16. Lakhan, A., Mohammed, M. A., Rashid, A. N., Kadry, S., Abdulkareem, K. H., Nedoma, J., Martinek, R., Razzak, I. Restricted Boltzmann Machine Assisted Secure Serverless Edge System for Internet of Medical Things. IEEE Journal of Biomedical and Health Informatics, 2022. https://doi.org/10.1109/JBHI.2022.3178660
17. Lakhan, A., Mohammed, M. A., Nedoma, J., Martinek, R., Tiwari, P., Vidyarthi, A., Alkhayyat, A., Wang, W. Federated-Learning Based Privacy Preservation and Fraud-Enabled Blockchain IoMT System for Healthcare. IEEE Journal of Biomedical and Health Informatics, 2022. https://doi.org/10.1109/JBHI.2022.3165945

18. Li, W., Tug, S., Meng, W., Wang, Y. Designing Collaborative Blockchained Signature-based Intrusion Detection in IoT Environments. Future Generation Computer Systems, 2019, 96, 481-489. https://doi.org/10.1016/j.future.2019.02.064

19. Lian, W., Nie, G., Jia, B., Shi, D., Fan, Q., Liang, Y. An Intrusion Detection Method Based on Decision Tree-Recursive Feature Elimination in Ensemble Learning. Hindawi Mathematical Problems in Engineering, 2020, Article ID 2835023. https://doi.org/10.1155/2020/2835023

20. Muna, A. H., Moustafa, N., Sitnikova, E. Identification of Malicious Activities in Industrial Internet of Things Based on Deep Learning Models. Journal of Information Security and Applications, 2018, 41, 1-11. https://doi.org/10.1016/j.jisa.2018.05.002

21. Naeem, H., Guo, B., Naeem, M. R., Ullah, F., Aldabbas, H., Javed, M. S. Identification of Malicious Code Variants Based on Image Visualization. Computers & Electrical Engineering, 2019, 76, 225-237. https://doi.org/10.1016/j.compeleceng.2019.03.015

22. Nimbalkar, P., Kshirsagar, D. Feature Selection for Intrusion Detection System in Internet-of-Things (IoT). ICT Express, 2021, 7(2), 177-181. https://doi.org/10.1016/j.icte.2021.04.012

23. Pecori, R., Tayebi, A., Vannucci, A., Veltri, L. IoT Attack Detection with Deep Learning Analysis. In 2020 International Joint Conference on Neural Networks (IJCNN), 2020, 1-8. https://doi.org/10.1109/IJCNN48605.2020.9207171

24. Pirretti, M., Zhu, S., Vijaykrishnan, N., McDaniel, P., Kandemir, M., Brooks, R. The Sleep Deprivation Attack in Sensor Networks: Analysis and Methods of Defense. International Journal of Distributed Sensor Networks, 2006, 2(3), 267-287. https://doi.org/10.1080/15501320600642718

25. Rahman, M. A., Asyhari, A. T., Leong, L. S., Satrya, G. B., Tao, M. H., Zolkipli, M. F. Scalable Machine Learning-based Intrusion Detection System for IoT-enabled Smart Cities. Sustainable Cities and Society, 2020, 61, 102324. https://doi.org/10.1016/j.scs.2020.102324

26. Sagu, A., Gill, N. S., Gulia, P. Hybrid Deep Neural Network Model for Detection of Security Attacks in IoT Enabled Environment. International Journal of Advanced Computer Science and Applications (IJACSA), 2022, 13(1). https://doi.org/10.14569/IJACSA.2022.0130115

27. Saranya, S. S., Sabiyath, F. N. IoT Information Status Using Data Fusion And Feature Extraction Method. Computers, Materials & Continua, 2022, 70(1), 1857-1874. https://doi.org/10.32604/cmc.2022.019621

28. Sitnikova, E., Foo, E., Vaughn, R. B. The Power of Handson Exercises in SCADA Cybersecurity Education. Information Assurance and Security Education and Training, Springer, Berlin, Heidelberg, 2013, 83- 94. https://doi.org/10.1007/978-3-642-39377-8_9

29. Stein, G., Chen, B., Wu, A. S., Hua, K. A. Decision Tree Classifier for Network Intrusion Detection with GA-based Feature Selection. Proceedings of the 43rd Annual Southeast Regional Conference on - ACM-SE 43, Kennesaw, Georgia, 2005, 136-141. https://doi.org/10.1145/1167253.1167288

30. Yadav, N., Pande, S., Khamparia, A., Gupta, D. Intrusion Detection System on IoT with 5G Network Using Deep Learning. Hindawi Wireless Communications and Mobile Computing, 2022, Article ID 9304689. https://doi.org/10.1155/2022/9304689

31. Yang, X., Peng, G., Zhang, D., Lv, Y. An Enhanced Intrusion Detection System for IoT Networks Based on Deep Learning and Knowledge Graph. Hindawi Security and Communication Networks, 2022, Article ID 4748528. https://doi.org/10.1155/2022/4748528

32. Zhou, Y., Qin, R., Xu, H., Sadiq, S., Yu, Y. A Data Quality Control Method for Seafloor Observatories: The Application of Observed Time Series Data in the East China Sea. Sensors, 2018, 18, 2628. https://doi.org/10.3390/s18082628