CRAMER'S CORRELATED DEEP CONVOLUTIONAL AND COX REGRESSIVE BOOTSTRAP CLASSIFIER FOR DOS ATTACK DETECTION

P. Nagarajan*[1] & Dr. S.Veni[2]

*[1]Research Scholar, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore -641024, India

[2]Professor, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore -641024, India

Abstract:

Wireless sensor network (WSN) occupies a certain role in diverse applications that needs data collection and transmission. With the random node deployment, they are susceptible to attacks. Most frequent attacks on WSNs which goal every layers of protocol stack is DoS attack. This attack directly affects network performance as well as prone network security. In this paper, a novel method called Cramer's correlated deep convolutional and Cox regressive Bootstrap Classifier (CCDC-CRBC) is proposed for DoS attack detection in WSN with enhanced accuracy, minimum time. CCDC-CRBC method comprises three different steps to determine the different types of DoS attacks in the network. First, a Z-score normalization-based preprocessing model is employed to eliminate the noise in the input dataset. Then the normalized outputs are fed to the process of feature extraction for selecting pertinent features. The process of feature extraction is performed by using Cramer's correlated deep convolutional learning. The deep convolution network comprises several layers to accurately learn the given input and extract more pertinent features. With the extracted features, the classification of different kinds of DoS attacks is made using Czekanowski Cox Regressive Bootstrap Aggregative Classifier. Simulation evaluation of the CCDC-CRBC method is performed by different metrics namely attack detection accuracy, attack detection time, precision, recall, f-measure. The outcomes demonstrate CCDC-CRBC method efficiently improves performance of DoS attack detection in WSN.

Keywords: Denial of Service, WSN, Z-score Normalization, Cramer's Correlation, Deep Convolutional Learning, Bootstrap Aggregative Classifier.

DOI: 10.24297/j.cims.2023.6.22

1. Introduction

WSNs are broadly used in numerous regions for real-time event recognition. The nodes and base station communicate in a wireless manner, which makes the nodes prone to different types of attacks. DoS attacks in WSNs are commonly varied when compared with other wired and wireless networks. Therefore, the detection of such kinds of attacks is essential in WSNs. For attaining improved recognition rate, capability to plan enhanced detection scheme is wanted behind using classification methods.

A method depended on Principal Component Analysis and Deep Convolution Neural Network called as (PCA and DCNN) was designed in [1] to determine the DoS attack in WSNs. First, PCA was employed to find the important features and eliminate the redundant features. Then, the DCNN was applied to classify the attack data. But, the attack detection was not reduced. To detect the DoS attacks, A novel method called Boruta Feature Selection with Grid Search Random Forest (BFSGSRF) algorithm was designed in [2]. But, the efficient classifier was unable to be used to improve the performance of classification with minimum training time.

A deep learning algorithm with a decision tree classifier was introduced in [3] to discover the normal and abnormal characteristics of the system. But, multiple attack detection was not carried out. A novel framework based on machine learning methods as described in [4] to find and diminish the DoS attacks. But, the accuracy of the classifiers was not efficient in attack detection.

To identify and give smart defense against DoS attacks in wireless network, Machine Learning-based attack detection approach was introduced in [5]. However, time consumption in attack recognition was not focused. Stealthy Data Transmission with Deep learning (SDTDL) model was employed in [6] to discover the jamming attacks in wireless networks. However, precision and recall metrics remained undressed.

Trust-based attack detection was used in [7] to detect the DoS attacks in WSN to carry out consistent data transmission. But, the deep learning mode was not used to find all kinds of attacks in the network. In [8], Intelligent Slime Mold (ISM) has been developed to identify the jamming node in WSN. However, the running time was not reduced.

For distinguish as well as segregate DoS attacks, A lightweight DoS detection method named Deep Learning-based Defense Mechanism (DLDM) has been introduced [9]. Despite the false alarm rate being reduced, the recall value was not improved. To determine the attacks in the network, Novel classifier ensemble model named stack of ensemble (SoE) was described in [10] . But, the classification time was not measured. To solve the above-mentioned issues, a novel attack recognition method depend on classification process is introduced.

1.1 Proposal Contributions

- To enhance performance of DoS attack recognition in WSN, CCDC-CRBC method is proposed by using three different processes such as data preprocessing, feature extraction, and classification.
- The proposed CCDC-CRBC method uses the Z-score normalization-based preprocessing model to get the better quality data for accurate attack detection.

- Cramer's correlated deep convolutional learning is introduced to extract the more relevant features with the help of multiple layers. The correlation between two features is measured in the hidden layers to identify the relevant and irrelevant features. By using feature extraction, the time needed for DoS attack detection is reduced.
- Czekanowski Cox Regressive Bootstrap Aggregative Classifier employed to accurately classify the given data for finding various kinds of Denial of Service (DoS) attacks namely Blackhole, Grayhole, Flooding, as well as Scheduling attacks in WSN.

Experiments are carried out to measure the performance of the CCDC-CRBC method and existing works. The experimental result reveals that the CCDC-CRBC method is highly efficient for DoS attack detection in WSN.

This manuscript is organized as follows. Section 2 explains literature survey of attack detection in WSN. Section 3 shows processes of CCDC-CRBC method. In Section 4 gives the experimental assessment. Section 5 describes outcomes as well as discussions of proposed and existing methods. Section 6 concludes the results of the proposed method.

2. Literature Review

To find false event data ,A new algorithm was designed in [11]. However, performance of attack recognition was not analyzed with enormous sensor nodes. To find the intrusion using Sparse Auto Encoder (SAE) and DNN, two-stage hybrid methodology was constructed [12]. But, the robust classification results failed to be obtained to increase the precision.

Naive Bayes Classifier was designed in [13] to detect the attack in WSN. For protecting the attack, Enhanced Code-Based Round Trip Time (EC-BRTT) technique was employed; the robust classification results were not obtained. A network model that depended on ANN was introduced in [14] to determine the distributed DoS attacks in the network. However, the attack detection time was not considerably reduced.

In [15], the fuzzy logic mechanism was developed for DDoS attack identification in a faster way. But, the attack detection performance was not improved in terms of precision. For finding service attack in WSN, Two types of machine learning techniques namely neural network and Support Vector Machine was employed [16]. But, the accuracy of attack detection was not improved.

Systematic survey of DDoS detection and mitigation approaches was examined in [17]. But, the reviewed method was not efficient due to the lack of generating strong classification results. For identifying clone node present in WSN, A hybrid clone node detection (HCND) mechanism was designed [18]. But, failed to improve the attack recognition rate.

Statistical model called exponential moving average (EMA) was developed in [19] to find the replica node attack. However, failed to measure the precision of attack recognition. False positive attacks , false negative attack detection methods were designed in [20]. But, attack detection accuracy was not measured.

3. Methodology

A DoS attack is type of cyber warfare that cruel users safeguard authentic applications. It attempts for using victim system's resources also initiates network congestion through producing an immense volume of traffic in the targeted region. Therefore, the detection of DoS attacks is needed to ensure better communication in WSNs. In this work, Cramer's correlated deep convolutional and Cox regressive Bootstrap Classifier (CCDC-CRBC) method is introduced for identifying various kinds of DoS attacks in WSN. The Block diagram of proposed CCDC-CRBC method is shown in figure 1.

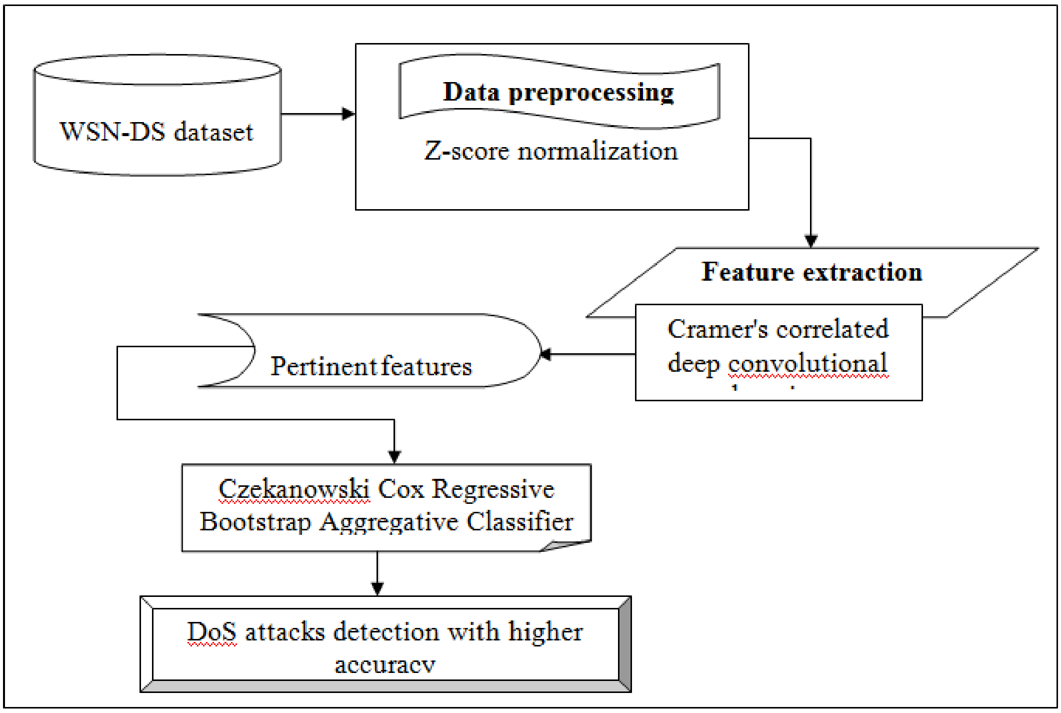

WSN-DS dataset
Data preprocessing
Z-score normalization
Feature extraction
Cramer's correlated deep convolutional
Pertinent features
Czekanowski Cox Regressive Bootstrap Aggregative Classifier
DoS attacks detection with higher accuracy

Figure 1 Block diagram Cramer's correlated deep convolutional and Cox regressive Bootstrap Classifier for DoS attack detection in WSN

From above figure, DoS attack recognition is performed by using the CCDC-CRBC method with the input of the WSN-DS dataset. From the input dataset, several data (samples) are collected as input. With this, Z-score normalization-based preprocessing model is employed to eliminate the noise presented in the data and provides normalized results. Then, Cramer's correlated deep convolutional learning is used on the normalized features to extract the relevant features for reducing the time consumed in the attack detection process. Lastly, the classification of data is carried out to find the different types of DoS attacks by using Czekanowski Cox Regressive Bootstrap Aggregative Classifier.

3.1 Data collection

The input data is considered to attain the attack detection that is collected via WSN-DS dataset. Categories of attack types are black hole, gray hole, flooding, and scheduling attacks. The features of WSN-DS dataset are 19 to find the DoS attack in the WSN by introducing the proposed method.

3.2 Z-score normalization-based preprocessing model

This section discussed pre-processing using the development of Z-score normalization. The data collected from the WSN-DS dataset comprises a different form of features and are not consistent for further process. There are some missing data or values and noise are included in the network. The processing of such data introduces the error in the final attack detection outcome. Therefore, data preprocessing using the normalization method is required for obtaining better-quality data for further use. To this end, Z-score normalization-based preprocessing model is applied to prepare the data understandably for increasing the accuracy of attack detection. The process of Z-score normalization-based preprocessing model is shown in figure 2.

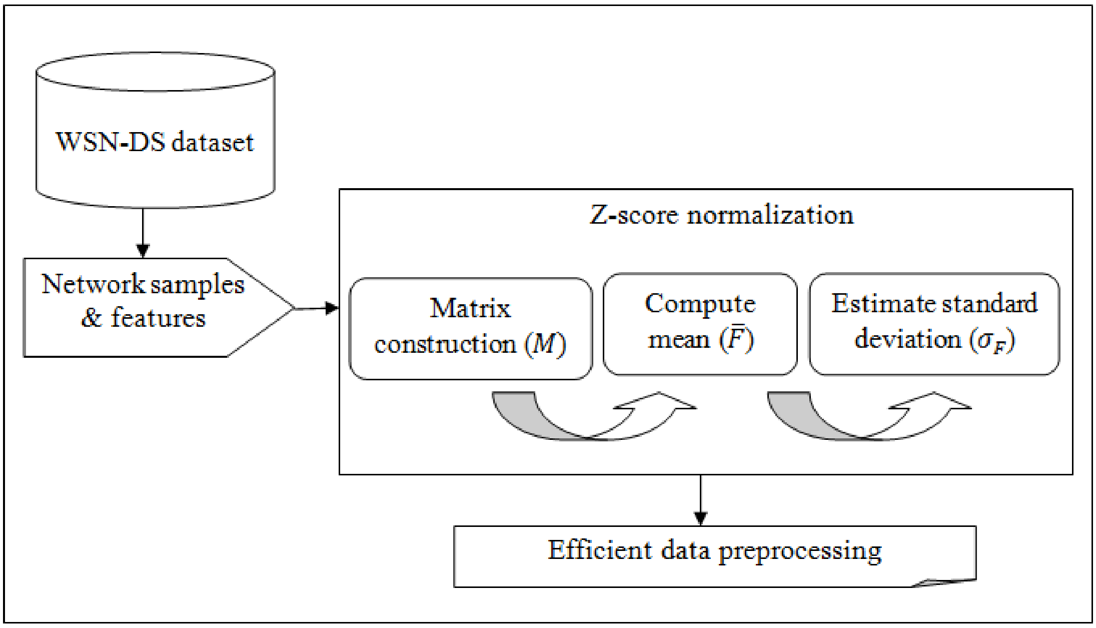

WSN-DS dataset
Network samples & features
Z-score normalization
Matrix construction (M)
Compute mean ($\bar{F}$)
Estimate standard deviation (σ_F)
Efficient data preprocessing

Figure 2 Block diagram of Z-score normalization based preprocessing

Figure 2 illustrates the block diagram of Z-score normalization based preprocessing. By applying preprocessing model, the process of noise data elimination is carried out. Let us consider, the WSN-DS dataset 'DS' with 'n' number of samples $S = \{S_1, S_2, \dots, S_n\}$ 'm' number of features '$F = \{F_1, F_2, \dots, F_m\}$'. The collected data (samples) and features are arranged in the matrix format (i.e. rows and columns) as given below.

$$M = \begin{bmatrix} S_{11} & S_{12} & \dots & S_{1m} \\ S_{21} & S_{22} & \dots & S_{2m} \\ S_{31} & S_{32} & \dots & S_{3m} \\ S_{n1} & S_{n2} & \dots & S_{nm} \end{bmatrix} \quad (1)$$

From equation (1), the collected data is denoted as a matrix M where each row denotes the samples ('n') and each column denotes a feature ('m'). Then, the mean value of the features in the samples is computed as follows.

$$\bar{F} = \frac{F_1+F_2+F_3+\cdots F_m}{m} \quad (2)$$

From equation (2), the mean of features '$\bar{F}$' in the dataset is determined for each sample where 'm' is the total number of features. After that, the standard deviation of the features is mathematically calculated as follows.

$$\sigma_F = \sqrt{\frac{\sum_{i=1}^{m}(F_i-\bar{F})^2}{m}} \quad (3)$$

From equation (3), 'σ_F' standard deviation of features is determined based on the feature 'F_i', a mean value of the features '$\bar{F}$' and a total number of features 'm'. Followed by, the Z-score function for each feature normalization is computed as given below.

$$Z - score = \frac{F_i-\bar{F}}{\sigma_F} \quad (4)$$

From equation (4), Z-score is computed to rescale the features in the dataset into [0, 1]. This, in turn, the noise data are eliminated and instabilities are fixed to reduce the computational complexity. Therefore, all the data in the dataset is normalized for the further training process. With the normalized data, the cluster head node and cluster member node are determined by computing the residual energy of the sensor node as follows.

$$E_{res}[SN] = E_{ini}\,[SN] - E_{cons}[SN] \quad (5)$$

Where 'E_{res}' is residual energy of sensor node, '$E_{ini}\,[SN]$' indicates initial energy of sensor node , '$E_{cons}[SN]$' refers to energy utilized by sensor node. By computing residual energy of sensor nodes, nodes are ranked during ascending order.

$$E_{res}[SN]_i = E_{res}[SN]_1 < E_{res}[SN]_2 < E_{res}[SN]_3 < \cdots < E_{res}[SN]_n \quad (6)$$

From the above equation, a node with a higher rank (i.e. higher) is selected as cluster head node (CH) , other nodes are considered cluster member nodes (CM). The pseudo-code of Z-score normalization-based Preprocessing is described below.

```
Input: Dataset ‘$DS$’, Features ‘$F = \{F_1, F_2, \dots, F_m\}$’, Network Samples ‘$S = \{S_1, S_2, \dots, S_n\}$’
Output: Efficient data preprocessing
Initialize ‘$m$’ number of features and ‘$n$’ number of samples
Begin
    1. For each Network Samples ‘$S$’ with Features ‘$F$’
    2. Construct matrix in equation (1)
    3. Find the mean of features given in equation (2)
    4. Calculate the standard deviation given in equation (3)
    5. Estimate Z-score as in equation (4)
    6. Compute residual energy of sensor node in equation (5)
    7. Rank the nodes based on ‘$E_{res}[SN]$’ in equation (6)
    8. Identify cluster head ‘$CH$’ and cluster member ‘$CM$’ nodes
    9. End for
End
```

Algorithm 1 Z-score normalization-based Preprocessing

The above Algorithm depicts process of Z-score normalization based preprocessing. The number of samples with their features (data) is collected from the WSN-DS dataset. Then, the Z-score normalization is applied to compute the mean and standard deviation values to eliminate noisy data and provide better quality data for DoS attack detection in WSN. Lastly, with the normalized data, CH and CM nodes are identified depend on computing residual energy of sensor nodes.

3.3 Cramer's correlated deep convolutional learning-based feature extraction

With the normalized features, feature extraction is performed using Cramer's correlated deep convolutional learning. Due to the increasing sizes of data, a huge amount of features or attributes are generated. All the features are generally not pertinent for attack detection. This causes inaccurate results and consumes more time for further processing. Therefore, feature extraction is needed to extract the pertinent features that would be more informative for the detection process. Therefore, Cramer's correlated deep convolutional learning is applied to significantly learn the given features for extracting pertinent features with minimum time and maximum accuracy. A Block diagram of Cramer's correlated deep convolutional neural network is given in figure 3.

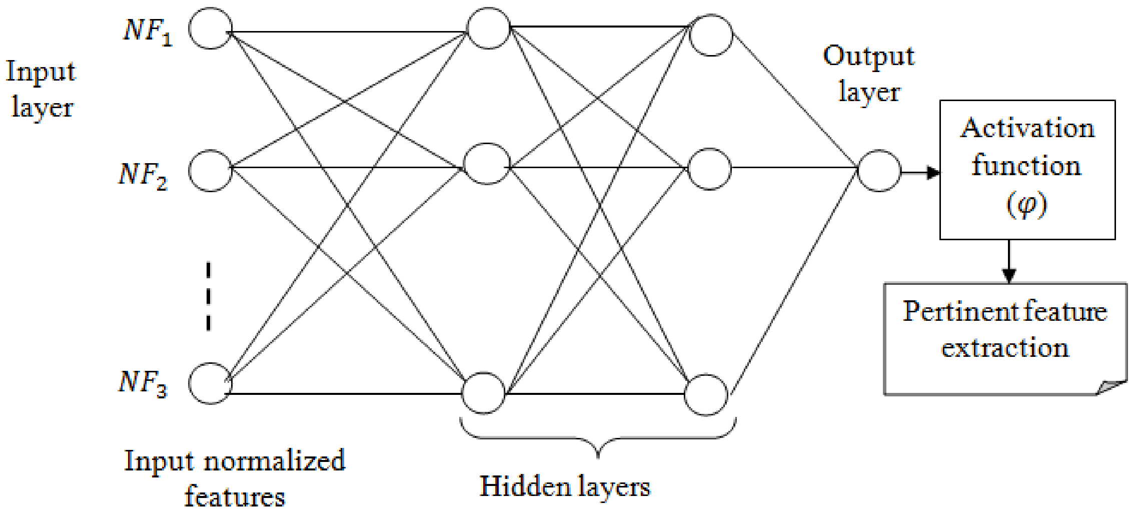

NF_1
Input layer
NF_2
NF_3
Output layer
Activation function (φ)
Pertinent feature extraction
Input normalized features
Hidden layers

Figure 3 Cramer's correlated deep convolutional learning-based feature extraction

As given in the above figure, with the results of normalized features from the preprocessing step, pertinent feature extraction is carried out using the proposed deep convolutional network. The deep convolutional network includes several layers such as input layer, hidden layer, output layer to deeply examine features. Here, normalized features are given as input in input layer. Hidden layer carry out Cramer's correlation between features. Lastly, the output of pertinent feature extraction is obtained at the output layer.

Consider the normalized features in the dataset are provided as input to the input layer at the time 't' through the variable weights and bias. Then, the activity of neurons in this layer is formulated as follows.

$$\alpha(t) = \sum_{i=1}^{m} \delta_1 * NF_i(t) + b \qquad (7)$$

Where '$\alpha(t)$' is a neuron activity at the input layer, '$NF_i(t)$' is a normalized feature, 'δ_1' is a weight in the input layer and 'b'refers a bias. The input in '$\alpha(t)$' is transformed to the hidden layers. In a deep convolutional network, hidden layers include a layer that performs a convolution operation. In that layer, Cramer's correlation is applied to compute the correlation between two features in the dataset and it is given by,

$$C_r = \left[\frac{\sum\sum|NF_i - AF_i|^2}{(m-1)+(n-1)}\right] \qquad (8)$$

Where C_r refers a Cramer's Correlation coefficient, NF_i refers to a normalized feature (i.e. training feature), AF_i refer to an attack feature, m and n are sample sizes. The results of correlation values varied from 0 to +1 where ‘0’ indicates no correlation between two features and ‘1’ indicates two features are highly correlated. Thus, output of hidden layer is provided as below.

$$\beta(t) = \sum_{i=1}^{m} NF_i\ (t) * \delta_1 + (\delta_2 * \beta\ (t-1)) \qquad (9)$$

Where $\beta(t)$ indicates output of hidden layer, ‘δ_2’ denotes weights of the hidden layers, ‘$\beta\ (t-1)$’ denotes output of earlier hidden layer. ‘$*$’ indicates convolutional operator. The output of hidden layer transformed to output layer.

$$\gamma(t) = \varphi\ [\delta_3 * \beta(t)] \qquad (10)$$

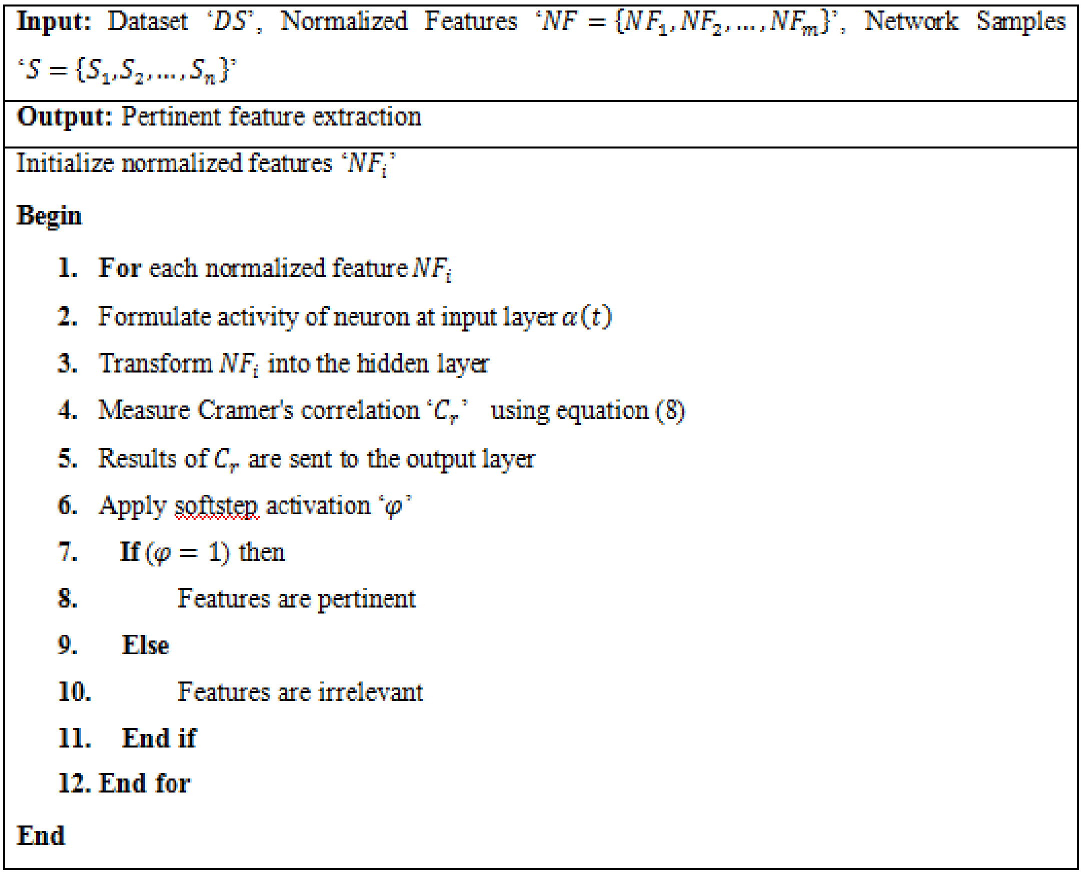
Where ‘$\gamma(t)$’ refers to an activity of neurons at the output layer, ‘δ_3’ refers to weight of hidden and output layer, ‘φ’ denotes activation function. In the output layer, the softstep activation function is applied to analyze the results of Cramer's correlation for extracting pertinent features.

$$\varphi = \frac{1}{1+e^{-C_r}} \qquad (11)$$

From the above equation, soft step activation function gives the outcomes as 0 or 1. If the activation function gives the output as ‘1’ then the input normalized features and attack features are highly correlated and the feature is said to be relevant. Otherwise, features are said to be irrelevant.

$$\varphi = \begin{cases} 1, & pertinent\ features \\ 0, & irrelvant\ features \end{cases} \qquad (12)$$

Based on activation function results, the pertinent features are extracted at the output layer. The pseudo-code representation of Cramer's correlated deep convolutional neural network feature extraction is given below.



```
Input: Dataset 'DS', Normalized Features '$NF = \{NF_1, NF_2, \ldots, NF_m\}$', Network Samples '$S = \{S_1, S_2, \ldots, S_n\}$'
Output: Pertinent feature extraction
Initialize normalized features '$NF_i$'
Begin
    For each normalized feature $NF_i$
    Formulate activity of neuron at input layer $\alpha(t)$
    Transform $NF_i$ into the hidden layer
    Measure Cramer's correlation '$C_r$' using equation (8)
    Results of $C_r$ are sent to the output layer
    Apply softstep activation '$\varphi$'
        If ($\varphi = 1$) then
            Features are pertinent
        Else
            Features are irrelevant
        End if
    End for
End
```

Algorithm 2 Cramer's correlated deep convolutional learning-based feature extraction

As given in the above algorithm, Cramer's correlated deep convolutional learning is applied to extract the pertinent features in a significant manner with the help of multiple layers. First, normalized features are taken as input in the input layer. Then, the hidden layer uses Cramer's correlation for finding the relationship between the normalized feature and the attack feature. The results of correlation values are analyzed using softstep activation function at the output layer. With this, pertinent and irrelevant features are identified, and extract the pertinent features for DoS attack detection. The designed feature extraction not only extracts the features, but also removes the irrelevant features for further process. This, in turn, attack detection time is reduced and thus the accuracy is improved.

3.4 Czekanowski Cox Regressive Bootstrap Aggregative Classifier

After extracting relevant features, the classification of data samples or data packets is carried out for identifying various sorts of DoS attacks in WSN by Czekanowski Cox Regressive Bootstrap Aggregative Classifier (CCRBEC). CCRBEC is a machine learning ensemble technique to classify the given samples (data) with extracted features in an accurate and time-efficient manner. On the contrary to other classification methods, CCRBEC gives robust classification results by constructing strong classifiers.

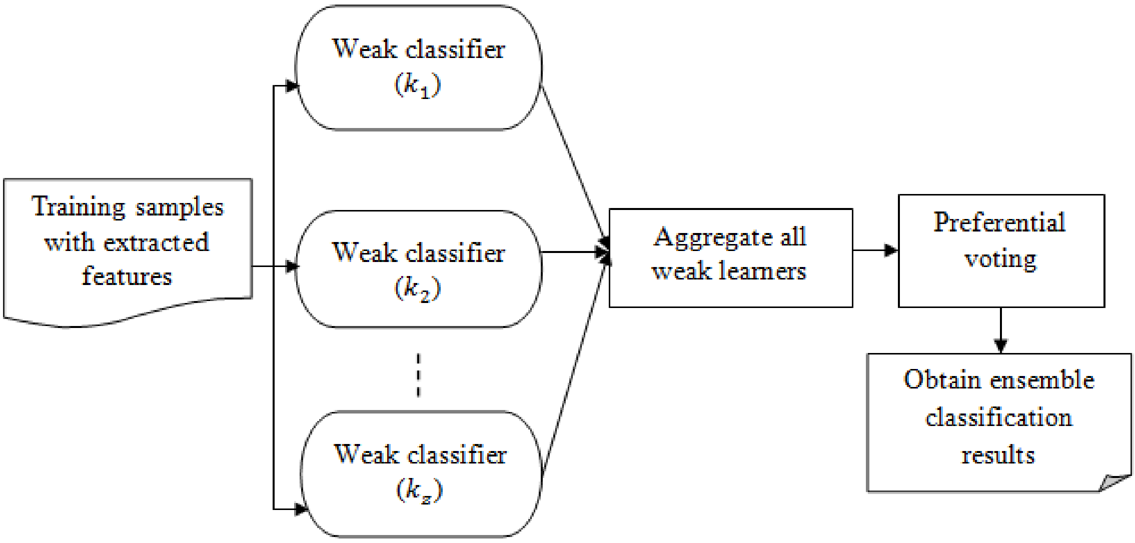

Training samples with extracted features
Weak classifier (k_1)
Weak classifier (k_2)
Weak classifier (k_z)
Aggregate all weak learners
Preferential voting
Obtain ensemble classification results

Figure 4 Block Diagram of Czekanowski Cox Regressive Bootstrap Aggregative Classifier

As given in the above figure, Czekanowski Cox regressive bootstrap aggregative classifier is an ensemble classifier that considers training data samples (bootstrap samples) '$S = \{S_1, S_2, \dots, S_n\}$' as input for attack detection. The input is subjected to the number of weak learners $k_1, k_2, k_3, \dots k_z$ in CCRBEC to classify the samples. The designed classifier uses the Czekanowski Cox regressive tree as weak learners to examine and classify the data into various classes. The Czekanowski Cox regressive tree includes the diverse nodes root node, branch nodes, and leaf node. The root node is a top node, branch node describes the value of the results and the leaf node gives an output of classification. Czekanowski-Cox Regression is applied to classify the data samples. Cox Regression is a statistical method to find the association between variables training data and testing data by using Czekanowski similarity as follows.

$$\Psi = 2 * \left(\frac{S_{tr} \cap S_{ts}}{\sum S_{tr} + \sum S_{ts} - S_{tr} \cap S_{ts}}\right) \qquad (13)$$

Where 'Ψ' refers a czekanowski similarity, 'S_{tr}' is a training sample with extracted features, 'S_{ts}'is a testing sample (i.e., DoS attack detection). The intersection symbol '∩' indicates mutual dependence among two samples. Depend on similarity measure, a weak classifier classifies all the data samples. The results of a weak classifier include some training errors. To reduce the error and thereby increase the accuracy of classification, all the results of weak classifiers are summed to get a strong classifier outcome.

$$y = \sum k_1 + k_2 + k_3 + \cdots., k_z \qquad (14)$$

Where 'y' denotes a bagging strong classification result, and 'k_z' is a weak classification outcome. Then, the error rate is computed between the predicted and exact classification results. It is given by,

$$\varepsilon = E_x(y) - P_r(y) \qquad (15)$$

Where 'ε' is an error rate, '$P_r(y)$' is a predicted output of classification using the ensemble classifier and '$E_x(y)$' is the exact output of the classification. Subsequently, preferential voting is used for ranking the classification results by considering the error rate. Here, the weak classifier result with lesser error is ranked first than the other classifier results. With the ranking process, the classifier with higher error is avoided to reduce the attack detection time. Lastly, the classifier with majority votes (i.e., lesser error) is selected to carry the data classification.

$$y = \underset{z}{arg\,max}\ \ v\,(k_i) \qquad (16)$$

Where $arg\,max$ refers to an argument of the maximum function to find the majority vote (v) of the weak classifier (i.e. k_i) whose conclusion is determined to the z^{th} classification results. With the obtained strong classifier results, DoS attack detection is performed. In our work, various sorts of DoS attacks are determined through data packet broadcast among CH to a base station through the CM nodes. Every CH gathers data packets from their cluster member node for further transmission to base station(BS). If CH blocks all data packets and not forwarding to sink node when CM sends data packets to CH, then blockhole attack is determined. CH drops data packets constantly or randomly, a Grayhole attack is detected. If the cluster head broadcasts a large volume of data packets than the threshold for taking a huge amount of energy from cluster members, then the flooding attack is identified. Lastly, a scheduling attack is detected when the cluster head assigns all cluster member nodes to the same TDMA schedule for sending the data packets. The pseudo-code for Czekanowski Cox regressive bootstrap aggregative classifier is described as follows.

```
Input: : Dataset ‘$DS$’, Samples ‘$S = \{S_1, S_2, \dots, S_n\}$’ with Extracted features $EF_i$
Output: Accurate DoS attack detection
Begin
    For each bootstrap data sample ‘$S_i$’
        Construct ‘$k$’number of weak learners
        Measure the Czekanowski similarity ‘$\Psi$’ using equation (13)
        Classify the samples
    End for
    Aggregate all weak learner results using equation (14)
    For each weak classifier results ‘$k_i$’
        Compute the error rate using equation (15)
        Rank the weak classifier results based on the error rate
        Select the weak classifier results with minimum error
        Determine majority votes of $k_i$ using equation (16)
        Get strong classification results
    End for
End
```

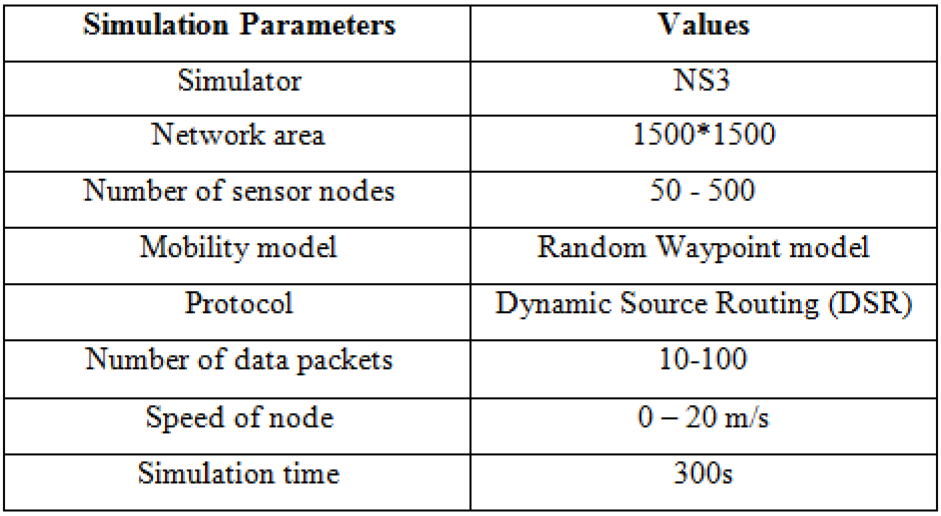
Algorithm 3 Czekanowski Cox Regressive Bootstrap Aggregative Classifier

As given in the above algorithm, Czekanowski Cox regressive bootstrap aggregative classifier used for classifying the data samples for DoS attack detection. With the input of pertinent features, classification is carried out to classify the data samples into different types of DoS attacks. First, the Czekanowski Cox regression tree is employed to measure the similarity between training and testing data samples. Depend on similarity value, input samples are categorized. Then the weak classification results are combined as well as error rate is determined. Voting scheme is used for finding weak classifier with minimum error. With the voting process, strong classification results are obtained. From that, all the data are classified to find the different types of DoS attacks in WSN.

4. Experimental Setup

Experimental evaluation of the proposed of CCDC-CRBC method and existing PCA and DCNN [1] and BFSGSRF [2] are implemented in NS3. The outcomes of DoS attack recognition by proposed and existing methods are analyzed by using the WSN-DS dataset (https://www.kaggle.com/datasets/bassamkasasbeh1/wsnds). The dataset comprises 374661 records (i.e. samples) that denote four kinds of attacks and normal behavior (no- attacks). The dataset comprises 19 features or attributes. By performing preprocessing and feature extraction processes, more relevant features are only extracted. Then, the samples are classified with extracted features to find the various types of attacks in the network. The results of the CCDC-CRBC method are compared with PCA and DCNN [1] and existing BFSGSRF [2] to ensure the effectiveness of the method.

Table 1 Simulation Parameters



Simulation Parameters	Values
Simulator	NS3
Network area	1500*1500
Number of sensor nodes	50 - 500
Mobility model	Random Waypoint model
Protocol	Dynamic Source Routing (DSR)
Number of data packets	10-100
Speed of node	0 – 20 m/s
Simulation time	300s

The results of CCDC-CRBC method are analyzed under the metrics namely attack detection accuracy, attack detection time, precision, recall, f-measure. The simulation result of the CCDC-CRBC method is compared with conventional methods such as PCA and DCNN [1] and existing BFSGSRF [2].

5. Discussions

Outcome of the proposed CCDC-CRBC method is discussed. Performance of CCDC-CRBC method compared with existing PCA and DCNN [1] and existing BFSGSRF [2].

5.1 Attack detection accuracy(ADA)

It is defined as ratio of several samples (data) classified normal or DoS attacks to total number of samples taken for the experimental evaluation. The formula for calculating the attack detection accuracy is given below,

$$A_{DA} = \left(\frac{NS_{CC}}{S_t}\right) * 100 \qquad (17)$$

Where A_{DA} denotes an attack detection accuracy, 'NS_{CC}' indicates number of samples correctly classified and 'S_t' denotes total number of samples. It computed in terms of percentage (%).

Table 2 Tabulation for Attack Detection Accuracy

Number of samples	Attack Detection Accuracy (%)		
	CCDC-CRBC	PCA and DCNN	BFSGSRF
50	94	88	82
100	92	85	80
150	90	83	79
200	89	81	78
250	88	79	76
300	86	78	75
350	84	76	73
400	82	74	71
450	81	73	68

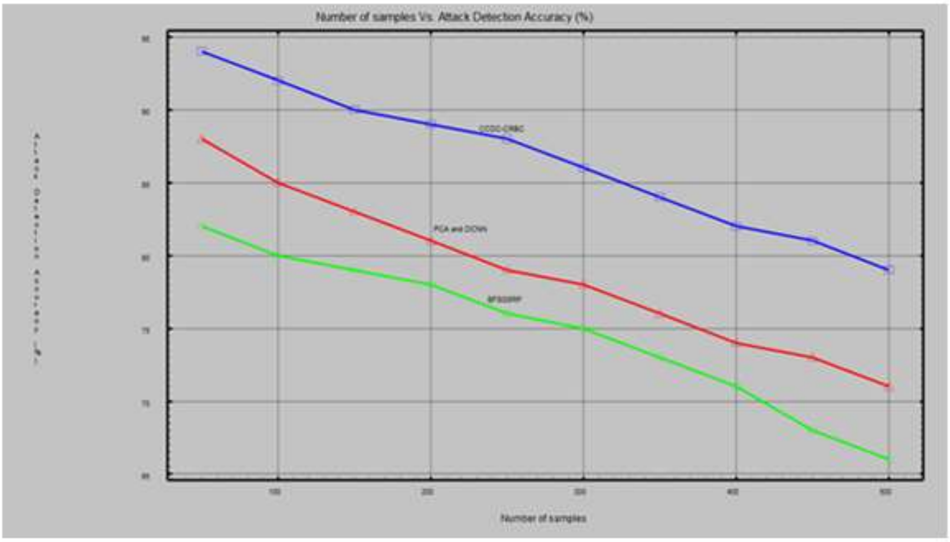


Figure 5 Results of Attack Detection Accuracy using CCDC-CRBC, PCA and DCNN [1] and existing BFSGSRF [2]

Table 2 and figure 5 show the comparative performance analysis of ADA the CCDC-CRBC method and conventional methods based on the number of samples. As shown in the figure, 'x' axis refers to the number of samples, and 'y'axis refers to the attack detection accuracy of three methods. As compared to existing XGBoost-RNN-IDS [1] and existing PCA and DCNN [2] methods, the proposed CCDC-CRBC method achieves higher accuracy during DoS attack recognition. Through increase in number of samples from WSN-DS dataset, results of attack detection accuracy are decreased. However, the CCDC-CRBC method gives comparatively better results. For instance, with the input of '50' samples in the first iteration, number of samples precisely classified normal or DoS attack is '47' using the CCDC-CRBC method whereas '44' samples are correctly classified using XGBoost-RNN-IDS [1] and '41' samples are correctly classified using PCA and DCNN [2] respectively. Then, the overall attack detection accuracy is obtained as 94%, 88%, and 82% for CCDC-CRBC method, existing [1] and [2] respectively.

The higher accuracy is attained by Czekanowski Cox Regressive Bootstrap Aggregative Classifier for attack detection. The designed bagging classifier uses the Czekanowski Cox Regressive tree as a weak classifier. The regression tree computes the similarity between the testing data (DoS attack data) value and training data with extracted features. If both the data are highly similar, then the data is classified as attack data. With this, the accuracy of attack detection improved in the CCDC-CRBC method than the existing [1] and [2]. The average of ten outcome demonstrate attack detection accuracy is improved using the proposed CCDC-CRBC method by 10% and 16% compared to PCA and DCNN [1] and BFSGSRF [2].

5.2 Attack detection time (ADT)

It is to find performance of DoS attack recognition in WSN. It is measured as amount of time needed for identifying different types of DoS attacks. It is mathematically calculated as given below.

$$DoSAD_{time} = \sum_{i=1}^{n} S_i * Time\ [AD] \tag{18}$$

Where '$DoSAD_{time}$' refers to a DoS attack detection time, 'S_i' refers to network samples considered for simulation and '$Time\,[AD]$' refers a time used in detecting the DoS attacks '$Time\ [AD]$'. It is determined in milliseconds (ms).

Table 3 Tabulation for Attack Detection Time

Number of samples	Attack Detection Time (ms)		
	CCDC-CRBC	PCA and DCNN	BFSGSRF
50	15.20	22.14	26.31
100	23.51	35.41	40.56
150	27.31	42.36	47.84
200	35.12	48.62	53.63
250	40.15	56.14	61.64
300	47.63	68.41	73.51
350	50.46	77.62	82.96
400	66.58	88.31	95.15
450	80.45	103.45	110.48
500	96.38	115.52	122.36

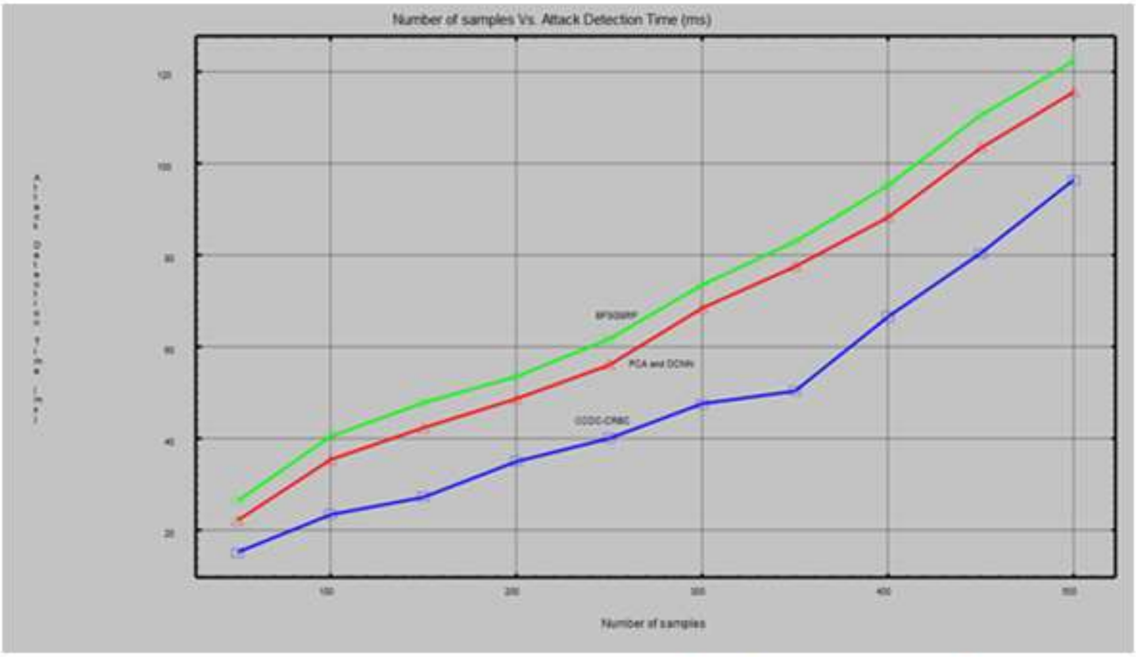


Figure 6 Results of Attack Detection Time using CCDC-CRBC, PCA and DCNN [1] and existing BFSGSRF [2]

From above Table 3 , Figure 6 demonstrate simulation outcomes of attack detection time with number of samples. As given in above figure, the effectiveness of the proposed CCDC-CRBC method for DoS attack detection is verified through the comparison with existing PCA and

DCNN [1] and existing BFSGSRF [2]. In the above figure, an increasing trend is observed. It means with the increase in the number of samples, the time taken for attack detection is also improved. However, the proposed CCDC-CRBC method uses minimal time for attack detection than the other methods. The lesser time is consumed with the application of Cramer's correlated deep convolutional learning. Cramer's correlation is used in the CCDC-CRBC method to determine the pertinent features that give enough information to detect the DoS attacks in WSN. Also, irrelevant features are eliminated in the deep learning model that decreases the time taken for DoS attack detection.

When considering 50 samples to compute attack detection time calculation, the proposed CCDC-CRBC method consumes '$15.2\,ms$' for classifying the data as normal or different types of DoS attacks. In addition, '$22.14\,ms$' and '$26.31\,ms$' of attack detection time is consumed in existing PCA and DCNN [1] and existing BFSGSRF [2] to classify and detect the attacks. Similarly, the remaining results of attack detection time are computed for various ranges of samples (i.e.,$100, 150 \dots 500$). From these results, it is concluded that the proposed CCDC-CRBC method reduces attack detection time by 29% , 35% compared to PCA and DCNN [1] , existing BFSGSRF [2].

5.3 Precision

It is estimated as number of samples correctly classified and detected DoS attacks to total number of samples taken for experimental evaluation. Precision mathematically computed as follows.

$$P = \frac{TP}{TP+FP} \tag{19}$$

In equation (19), 'P' refers a precision, and is it measured depending on true positive 'TP' (i.e., no. of samples accurately classified DoS attacks) , false positive rate 'FP' (i.e., no. of samples mistakenly classified DoS attacks). The results precision using different methods is provided as below.

Table 4 Tabulation for precision

Number of samples	Precision		
	CCDC-CRBC	PCA and DCNN	BFSGSRF
50	0.93	0.86	0.80
100	0.92	0.84	0.78
150	0.92	0.83	0.77
200	0.90	0.81	0.76
250	0.89	0.80	0.74
300	0.87	0.78	0.70
350	0.85	0.75	0.69
400	0.83	0.74	0.68
450	0.81	0.72	0.67
500	0.79	0.70	0.66

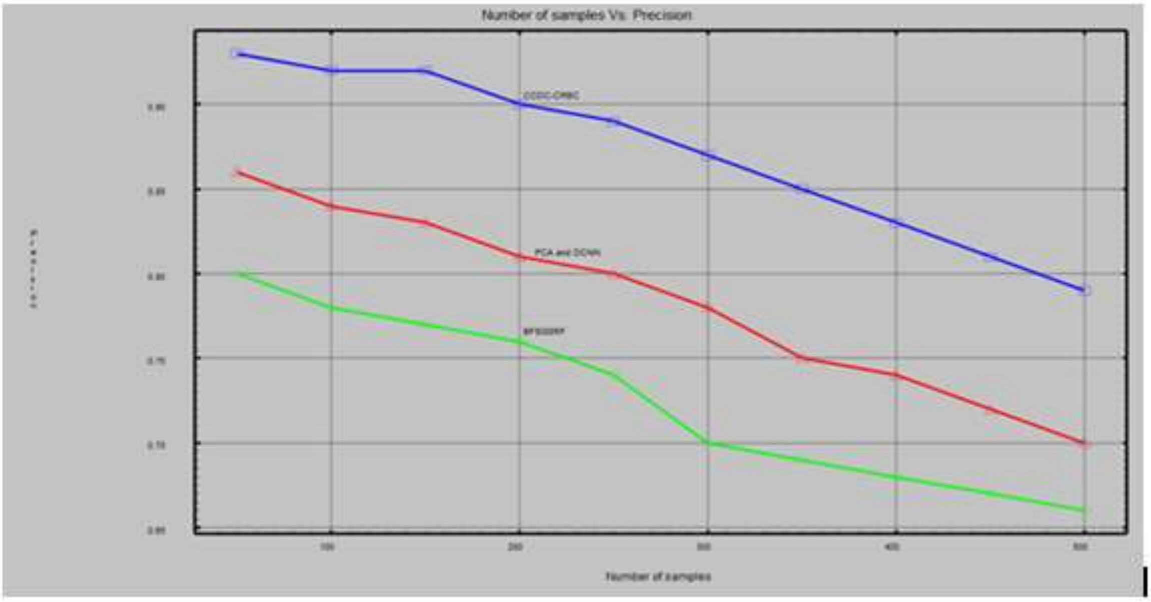


Figure 7 Results of Precision using CCDC-CRBC, PCA and DCNN [1] and existing BFSGSRF [2]

From above Table 4 , figure 7 depict performance outcome of precision using CCDC-CRBC method, PCA and DCNN [1], and existing BFSGSRF [2] methods. A number of network samples is used as input and it is considered in the ranges from 50 to 500. From the above figure, it is clear that the CCDC-CRBC method increases the precision value during DoS attack detection more than the existing [1] and [2] methods. This is proved with the statistical analysis of tabulated values. With the input of 50 network samples taken in DoS attack detection, the actual correct classification of samples is 45. From that, the proposed '42' number of network samples was correctly detected using the CCDC-CRBC method, '39' and '37' number of network samples were correctly detected using existing PCA and DCNN [1] and existing BFSGSRF [2] methods. Thus, the overall precision is measured as '0.93', '0.86' and '0.80' for the CCDC-CRBC method, existing [1] and [2] respectively. This higher value of precision is attained by measuring Czekanowski Cox similarity in the bootstrap aggregating classifier. With this, the training data that is more related to the testing data is classified to detect the diverse types of DoS attacks in WSN. In turn, the precision of the CCDC-CRBC method is improved to the other methods. The simulation results of precision using the proposed CCDC-CRBC method is enhanced by 11% and 20% compared to existing PCA and DCNN [1] and existing BFSGSRF [2] methods.

Recall

It is defined as ratio of the TP results (i.e., correct detection of DoS attack) that are successfully obtained. The recall is calculated as follows,

$$R = \frac{TP}{TP+FN} \tag{20}$$

In equation (20), 'R' is a recall, 'TP' is a true positive and 'FN' is a false negative (no. of samples are wrongly classified as normal). The results of the recall proposed and existing methods are given below.

Table 5 Tabulation for recall

Number of samples	Recall		
	CCDC-CRBC	PCA and DCNN	BFSGSRF
50	0.90	0.83	0.78
100	0.88	0.81	0.75
150	0.87	0.80	0.73
200	0.85	0.78	0.71
250	0.84	0.77	0.69
300	0.83	0.76	0.68
350	0.82	0.75	0.65
400	0.816	0.73	0.63
450	0.803	0.70	0.61
500	0.794	0.69	0.60

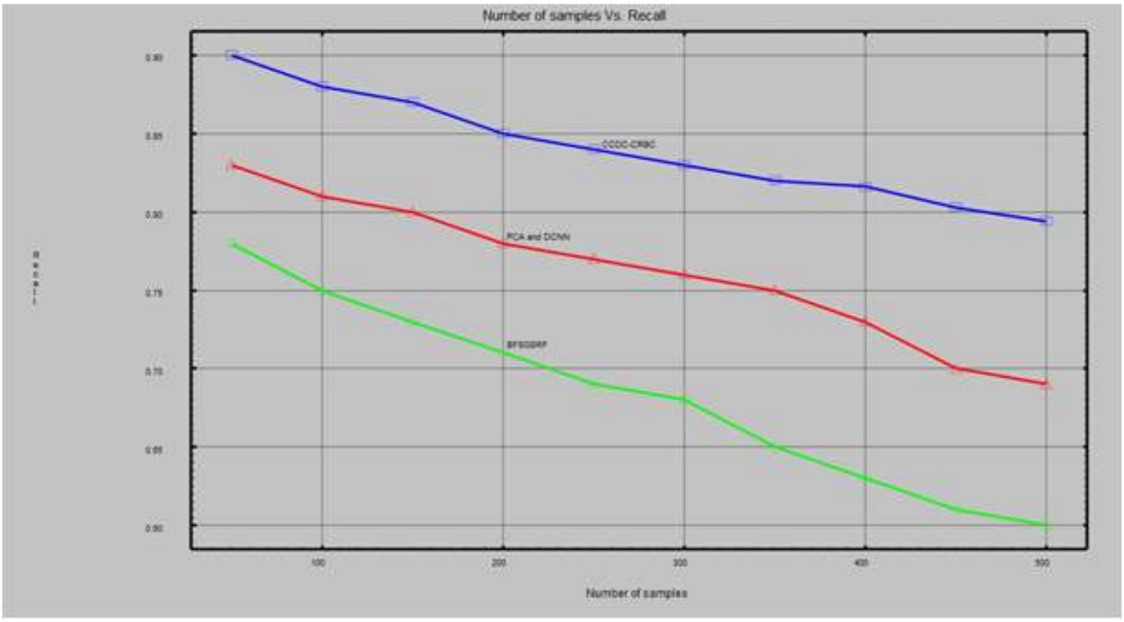


Figure 8 Results of Recall using CCDC-CRBC, PCA and DCNN [1], and existing BFSGSRF [2]

Table 5 and Figure 8 illustrate the experimental results of recall for the proposed CCDC-CRBC method with existing [1] and [2] methods based on the number of samples. As given in the above figure, the horizontal axis takes the network samples, vertical axis provides the recall value obtained for three classification methods. As compared to existing PCA and DCNN [1] and existing BFSGSRF [2], recall of the proposed CCDC-CRBC method is found to be better in the attack detection process. By using 50 samples as input, recall of the proposed CCDC-CRBC method is obtained as ‘0.9’ whereas recall of PCA and DCNN [1] is obtained as 0.83 and BFSGSRF [2] obtains ‘0.78’ respectively. Likewise, the results of a recall are obtained for nine remaining runs. In all the runs, the output of recall is improved in the CCDC-CRBC method than the other methods. This is due to the design using Cramer's correlated deep convolutional learning algorithm. In this algorithm, the normalized features are taken as input. Then, Cramer's correlation is employed in the hidden layer where the correlation between features is computed. Lastly, softstep activation is employed to find the pertinent and irrelevant features for DoS attack detection. The CCDC-CRBC method considers pertinent features for accurate attack detection. Therefore, the results of the recall are improved in the CCDC-CRBC method than the other methods. The experimental outcome of recall using the proposed CCDC-CRBC method is enhanced by 105 and 24% than the existing PCA and DCNN [1] and existing BFSGSRF [2] methods.

5.5 F-measure

It is measured based on average results of precision as well as recall. The F- measure is mathematically determined as follows.

$$F - measure = 2 * \left(\frac{P*R}{P+ R}\right) \tag{21}$$

From equation (21), the f-measure is measured to evaluate the performance of DoS attack detection.

Table 6 Tabulation for f-measure

Number of samples	F-measure		
	CCDC-CRBC	PCA and DCNN	BFSGSRF
50	0.91	0.84	0.79
100	0.89	0.82	0.76
150	0.89	0.81	0.74
200	0.87	0.79	0.73
250	0.86	0.78	0.71
300	0.84	0.76	0.68
350	0.83	0.75	0.66
400	0.82	0.73	0.65
450	0.81	0.71	0.63
500	0.79	0.69	0.62

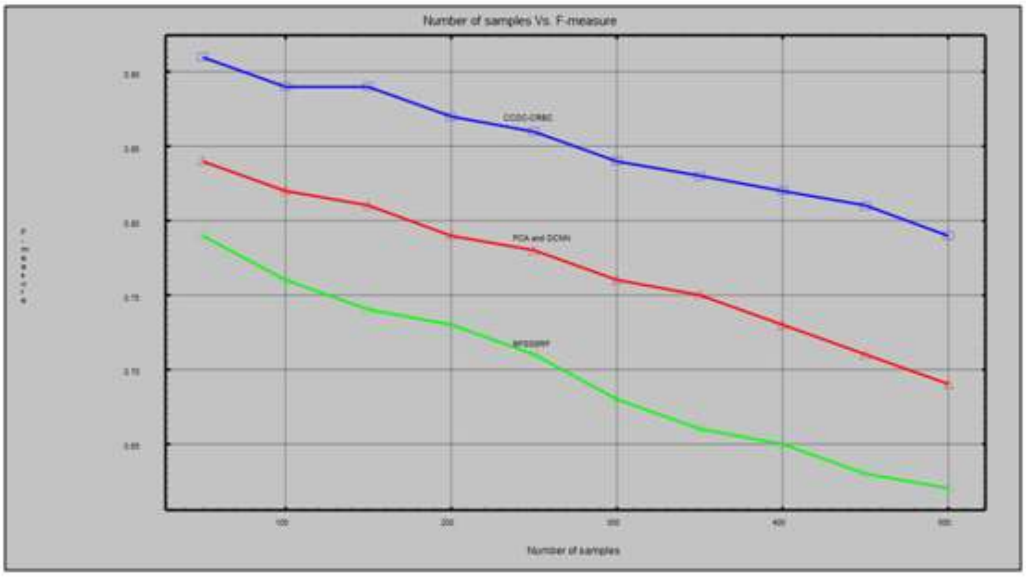


Figure 9 Results of F-measure using CCDC-CRBC, PCA and DCNN [1] and existing BFSGSRF [2]

From above Table 4 and figure 7 illustrate performance outcome of f-measure with number of data samples. In the above figure, the performance of f-measure is improved in the proposed CCDC-CRBC method than the existing PCA and DCNN [1] and existing BFSGSRF [2]. When taking 50 samples in the first iteration, the precision is observed as '0.93', recall is observed as '0.9' and the f-measure is observed as '0.91' for the CCDC-CRBC method. Similarly, '0.84' of f-measure is measured for existing PCA and DCNN [1] and '0.79' for BFSGSRF [2]. This is because of using deep learning-based feature extraction and bootstrap method for the classification process. First, deep learning is applied to get the relevant features. Then, the bootstrap method is employed to categorize the diverse kinds of DoS attacks. An accurate classification of attacks is made. Therefore, the performance of f-measure using the proposed CCDC-CRBC method is enhanced by 10% , 24% compared to [1] and [2].

6. Conclusion

A novel method referred to as CCDC-CRBC is introduced for DoS attack detection using a classification process in WSN with better accuracy. The proposed CCDC-CRBC method initially performs a data preprocessing process with the help of Z-score normalization for getting a better quality of data to get accurate attack detection results. Then, Cramer's correlated deep convolutional learning is applied to carry out the feature extraction where the more informative features for DoS attack detection are obtained. The correlation measure between normalized features and attack features not only extracts the pertinent features but also eliminates the redundant features that are not contributed to attack detection. Lastly, a bagging ensemble classifier called Czekanowski Cox regressive bootstrap aggregative classifier employed for classifying the data with extracted features in an accurate manner. Experimental evaluation is carried out through the WSN-DS dataset by using attack detection accuracy, attack detection time, precision, recall, and f-measure. Experimental outcomes clearly demonstrate CCDC-CRBC method improves ADA, precision, recall, and f-measure with minimum ADT than the conventional method.

References

[1] Chengpeng Yao, Yu Yang, Kun Yin, Jinwei Yang, "Traffic Anomaly Detection in Wireless Sensor Networks Based on Principal Component Analysis and Deep Convolution Neural Network", IEEE Access, Volume 10, 2020, Pages 103136 – 103149

[2] Sridevi Subbiah, Kalaiarasi Sonai Muthu Anbananthen, Saranya Thangaraj, Subarmaniam Kannan, Deisy Chelliah "Intrusion Detection Technique in Wireless Sensor Network using Grid Search Random Forest with Boruta Feature Selection Algorithm", Journal of Communications and Networks, Volume 24, Issue 2, 2022, Pages 264 – 273

[3] [3] Judy Simon, N. Kapileswar, Phani Kumar Polasi, M. Aarthi Elaveini, "Hybrid intrusion detection system for wireless IoT networks using deep learning algorithm", Computers and Electrical Engineering, Elsevier, Volume 102, September 2022, Pages 1-11

[4] Alaeddine Mihoub, Ouissem Ben Fredj, Omar Cheikhrouhou, Abdelouahid Derhab,

Moez Krichen, "Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques", Computers & Electrical Engineering, Elsevier, Volume 98, March 2022, Pages 1-11

[5] B.B. Gupta, Pooja Chaudhary, Xiaojun Chang, Nadia Nedjah, "Smart defense against distributed Denial of service attack in IoT networks using supervised learning classifiers", Computers & Electrical Engineering, Elsevier, Volume 98, 2022, Pages 1-13

[6] E. Jayabalan, R. Pugazendi, "Deep learning model-based detection of jamming attacks in lowpower and lossy wireless networks", Soft computing, Springer, volume 26, 2022, Pages 12893–12914

[7] C. Anand, N. Vasuki, "Trust Based DoS Attack Detection in Wireless Sensor Networks for Reliable Data Transmission", Wireless Personal Communications, Springer, volume 121, 2021, Pages 2911-2926

[8] Amirthasaravanan Arivunambi, Arjun Paramarthalingam, "Intelligent slime mold algorithm for proficient jamming attack detection in wireless sensor network", Global Transitions Proceedings, Elsevier, Volume 3, Issue 2, November 2022, Pages 386-391

[9] M. Premkumar, T.V.P. Sundararajan, "DLDM: Deep learning-based defense mechanism for denial of service attacks in wireless sensor networks", Microprocessors and Microsystems, Elsevier, Volume 79, November 2020, Pages 1-10

[10] Bayu Adhi Tama, Sunghoon Lim, "Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation", Computer Science Review, Elsevier, Volume 39, February 2021, Pages 1-27

[11] Yuichi Sei, Akihiko Ohsuga, "False Event Message Detection Robust to Burst Attacks in Wireless Sensor Networks", IEEE Open Journal of the Communications Society, Volume 3, 2022, Pages 1630-1642

[12] K. Narayana Rao, K. Venkata Rao, Prasad Reddy P.V.G.D, "A hybrid Intrusion Detection System based on Sparse autoencoder and Deep Neural Network", Computer Communications, Elsevier, Volume 180,2021, Pages 77-88

[13] K. Lakshmi Narayanan, R. Santhana Krishnan, E. Golden Julie, Y. Harold Robinson, Vimal Shanmuganathan, "Machine Learning Based Detection and a Novel EC BRTT Algorithm Based Prevention of DoS Attacks in Wireless Sensor Networks", Wireless Personal Communications, Springer, Volume 127, 2021, Pages 479-503

[14] R. Gopi, V. Sathiyamoorthi, S. Selvakumar, Ramasamy Manikandan, Pushpita Chatterjee, N. Z. Jhanjhi, Ashish Kumar Luhach, "Enhanced method of ANN based model for detection of DDoS attacks on multimedia internet of things", Multimedia Tools and Applications, Springer, Volume 81, 2022, Pages 26739–26757

[15] P. J. Beslin Pajila, E. Golden Julie, Y. Harold Robinson, "FBDR-Fuzzy Based DDoS Attack Detection and Recovery Mechanism for Wireless Sensor Networks", Wireless Personal Communications, Springer, volume 122, 2022, Pages 3053-3083

[16] Dongxian Yu, Jiatao Kang, Junlei Dong, “Service Attack Improvement in Wireless Sensor Network Based on Machine Learning”, Microprocessors and Microsystems, Elsevier, Volume 80, February 2021, Pages 1-6

[17] Ismael Amezcua Valdovinos, Jesús Arturo Pérez-Díaz, Kim-Kwang Raymond Choo, Juan Felipe Botero, “Emerging DDoS attack detection and mitigation strategies in software-defined networks: Taxonomy, challenges and future directions” Journal of Network and Computer Applications, Elsevier, Volume 187, 2021, Pages 1-29

[18] P.P. Devi, B. Jaison, “Protection on Wireless Sensor Network from Clone Attack using the SDN-Enabled Hybrid Clone Node Detection Mechanisms”, Computer Communications, Elsevier, Volume 152, 2020, Pages 316-322

[19] S. Anitha, P. Jayanthi, R. Thangarajan, “Detection of Replica Node Attack Based on Exponential Moving Average Model in Wireless Sensor Networks”, Wireless Personal Communications, Springer, volume 115, 2020, Pages 1651–1666

[20] Jung Sub Ahn and Tae Ho Cho, “Modeling and Simulation of Abnormal Behavior Detection Through History Trajectory Monitoring in Wireless Sensor Networks”, IEEE Access, Volume 10, 2022, Pages 119232 – 119243.