SID-RIS: Cascaded Intrusion Detection System for Industrial Internet of Things

P.L.S. Jayalaxmi [a], Rahul Saha [a], Lalit Garg [b], and Gulshan Kumar [a]

[a] *School of Computer Science and Engineering Lovely Professional University, Punjab, India*
[b] *Department of Computer Information System (CIS) Faculty of Information Communication Technology (ICT) University of Malta, Malta*

Abstract

The remote accessibility of Industrial Control System (ICS) with the emergence of smart industrial infrastructure has initiated various vulnerabilities and security breaches in industrial networks and Supervisory Control and Data Acquisition (SCADA). The development of specific security mechanisms can reduce the vulnerabilities of physical and data explosions with less human intervention and control the environmental and financial loss. Traditional Intrusion Detection Systems (IDSs) are very much prone to false-positive rates, high implementation costs, and low-speed models. We propose a novel Smart Intrusion Detection with Risk Identification System (SID-RIS) incorporated with Deep Learning (DL) algorithms. The proposed model is trained and tested on BoT-IoT and KDD+ datasets for the optimal features. The results show that the model is most suitable for classifying the anomaly behavior of the data with high accuracy and low false rates.

Keywords 1

Industrial Control Systems (ICS), Supervisory Control and Data Acquisition, Intrusion Detection, Deep Learning, False-Positive.

1. Importance of security in IIOT

The increased connectivity of smart machines raises thewagers. The first alarm situation to breach industrial security was in 2009, when the speed of the centrifuge nuclear enrichment plant was modified to spin out of control by a Stuxnet malware. This was introduced via flash drive for a stand-alone network, which spread automatically across the networks [1]. A new malware called Trident destabilizes Safety Instrumented System (SIS) and provides a path forhackers to destroy the files by feeding false data [2]. A strong, smart, and safe shield is very much essential to reduce industrial espionage, IP leakage, information theft, which maylead to the production sabotage. Industrial Control Systems (ICSs) have unique vulnerability, as each connected devices represent a potential risk in each layer of the network, whichis particularly susceptible to cyberattack. The major cause of hazardous issues in the industrial sector is the incompatible operation system, outdated Programmable Logic Controllers (PLCs), and Human-Machine Interfaces (HMIs) in an isolated environment with a lack of regular updates on attack patterns, and poor standards [3]. Table 1 provide a detailed list of IoT attack and the proposed counter measures to mitigate the complications of the attack.

WAI-2022: Workshop on Artificial Intelligence, January 27 – 28, 2022, Chennai, India.
EMAIL: rsahaaot@gmail.com (Rahul Saha)
ORCID: 0000-0003-3921-9512 (Rahul Saha)

Table 1: Various IoT attack and the Counter Measures

IoT Layer	Attack type	Measures
Physical	Jamming DoS, Collision, Exhaustion, Man-in-the-Middle attacks	Packet alternate (re)routing System logs modelling Spiking neural network classification CUmulative SUM (CUSUM) algorithm [4],[5]
Datalink	, Phishing, Data Transit	Data encryption algorithm Intelligence Web Application Firewall (IWAF) URL Embedding (UE) [6],[7]
Network	Routing, DDoS, SCADA Modbus attacks	Network filtering and Secure MQTT, ABE algorithm Next Generation firewalls filtering capabilities Mapping by extracting URLs from spam mail [8]
Transport	System flooding	Intrusion detection and prevention system Compressed Transport Protocols Ingress filtering and IDS solutions [9]

1.1. Introduction – IDPS

An Intrusion Detection System (IDS) is a network securitytechnology built to detect vulnerable exploits in the cyberworld. IDS is classified in two forms based on the detection component as Network Intrusion Detection System (NIDS) and Host-based Intrusion Detection System (HIDS). A NIDS observes, monitors, and analyzes the network traffic to identifysuspicious events, whereas HIDS trace abnormal activities andreport to the security server. Anomaly Intrusion Detection(AID) observes the behavior by scanning the ports. Signature Intrusion Detection (SID) matches predefined patternsbased on vulnerability and exploit are used to defend thesituation [5]. SID methods have high rate of accuracy in classifying known attacks and AID methods are popularin identifying zero-day attacks; Both of the methods pro- duce high false-positive rates. IDS detects and reports to the security system but, lacks in preventing the exploitation, neither raise any automatic action to mitigate the risk [6]. Nowadays Intrusion Detection and Prevention System (IDPS) has become the dominant deployment option for the security system [7]. Feature selection, compatibility, and unavailabilityof the labeled dataset are the primary challenges faced by the current IDS models. Immense efforts are required to collect labeled datasets from real-time network traffic and preserve the confidentiality of the internal data. Feature selection plays a vital role in the development of the classification model, to learn good features on a limited amount of labeled datain supervised classification [8]. These features can be applied for other classification models with a small amount of dataset.Deep learning techniques are more popular for feature reduction and classification; these methods are successfully applied in image, audio, text, and numerical dataset for developing application models [9]. In this present work, we propose a Deep Learning (DL)-based NIDS model for classification and identifying the most relevant features and detecting the anomalies. We call our proposed model of intrusion as Smart Intrusion Detection with Risk Identification System (SID-RIS). The model is trained and verified on the KDD+ datasets and UNSW-BoT-IoT dataset. We have presented a comparative analysis with the existing techniques to evaluate the efficiency of our model.

1.2. Organization

The remaining paper is organized in four sections. In Section 2, a few latest and closely related work is discussed. Section 3 presents an overview of the proposed model and the risk factor analysis with the implementation procedures on both datasets. The results and comparative analysis of the model are discussed in Section 4. We conclude our work with future scope in Section 5.

2. Related Work

Focusing on security applications, DL techniques with remarkable quality of self-learning are beneficial to developing intrusion detection models. These models result in low false rates and high accuracy as compared to traditional machine learning techniques. The standard Neural Network (NN) architecture is created with a multi-layer perceptron using a liner stack classifier. Raw data in the form of numbers/images/audio are fed into the neurons as input represented with x1, x2, x3,..., xn. Each input is multiplied by weights (w1,w2,w3,…,wn) and passed to an activation function which maps the input signals into an output signal.

$$z = f(b + \sum_{i=1}^{N} x_i w_i) \quad (1)$$

In Equation 1, x represents the inputs, w represents weights to be added for each input, z is used for output, b represents bias, and f represents the activation function. The model adjusts the weights and repeats the task to improve the accuracy using back-propagation.

Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) are the most popular methods used for detecting malware activities with self-learning techniques [13]. ANN is emphatic in monitoring network traffic and detecting Imminent attacks. ANN, CNN, and Deep Neural Network (DNN) are some of the supervised instance learning techniques trained with feed-forward neural networks. Yazan et al. [10] propose a Spider Monkey Optimization (SMO) algorithm for dimensionality reduction and the Stacked-Deep Polynomial Network (SDPN) for attack classification. The Deep Feature Embedding Learning (DFEL) model has been compared with KNNs, DT, and SVM and results with a 99.14% F1 score. Olakunie Ibitoye et al. [11]. compared a Feedforward neural network model with a self-normalizing neural network model for BoT-IoT dataset and resulted in 9% higher performance accuracy of SNN IDS than FNN IDS. A Generic algorithm- based Deep belief network model was proposed by Zhang et al. [12]. The model structure was integrated with a selection of features with crossover, mutation, and elite retention technique of generic algorithm. once the maximal algebraic value is reached the optimal structure is created. Restricted Boltzmann Machine (RBM) and Back-propagation network (BPN) are used for classification [12]. Roopak et al. [13] pro- pose four different classification deep learning models as MLP (Multilayer Perceptron), 1d-CNN, LSTM, CNN+LSTM with a comparative analysis on machine learning technique. CNN+LSTM, LSTM, 1d-CNN techniques have high accuracy rate than SVM, Naive Bayes, and Random Forest machine learning techniques for the CICIDS2017 dataset. CNN+LSTM remains the best-proposed technique with 97.15% accuracy where LSTM results in 96.24% and MLP results in 13.66% false rates. Thamilarasu G. et al. [14] propose a three-layer framework with network connection phase, anomaly detection phase, and the mitigation phase to identify, analyze, and reduce the risk factor using CNN techniques. Another integrated technique using LSTM and CNN as Hybrid CNN model testing on UNSW dataset is proposed by S.smys et al. [15]. LSTM is used for feature extraction and CNN for intrusion detection. The model gave an excellent performance with a 2.19 sec training time. Another Deep belief network model tested on a real-time dataset proposed by Balakrishnan et al. [16]. This model enhances the security network compared to Domain Generation Algorithm (DGA) with 0.997 highest precision. Chao Liang et al. [17] propose a multi-agent system with the blockchain and deep learning (DNN) algorithm, tested for the NSL-KDD dataset, and resulted in 91.50% accurate on testing. The Transient Search Optimization (TSO) algorithm by Fatani et al. [23] maintains the balancing between exploitation and exploration phases. The model is tested on the most popular IoT datasets including KDDCUP-99, NSL-KDD, BoT-IoT, and CICIDS-2017. It achieves higher accuracy compared to several existing approaches.

3. SID-RIS Model

The traditional architecture of the IDS model is prone to security leaks. The multi-layer recursive structure analyzes the data at various levels and makes the model effective to handle the minute complications. The IoT system is exposed to multiple devices with different processing frameworks

connected to various locations. A single layer model lacks in generating enhanced performance, as it is restricted with the scope of the connected components. The multi-layer model is distributed across the system and executes the processes at each level covering the major to minor values based on the state of the system. A self-trained security model with previous inputs minimizes human interaction. In this section, we focus on the proposed methodology. The data set used in building the detection model, the feature extraction techniques, and the multi-layered Cascade detection and classification algorithm are explained in sequence in this section.

3.1. Dataset

We use NSL-KDD and UNSW-2018-BoT-IoT datasets in the present work. KDD Cup dataset is prepared using the network traffic captured by the 1998 DARPA IDS evaluation program [24]. The BoT-IoT dataset is collected from Cyber Range Lab of UNSW Canberra [25].

KDD + Dataset We have used the NSLKDD+ dataset that has 41 labeled input features with binary and multi-class attack classification. A total of 38 traffic classes with 21 attack classes are available in the test data, from which 16 attacks and 1 normal class are considered for training. The attack records are grouped into four major classes as DoS, Probing, user-to- root (U2R), and root-to-local (R2L) [24]. We have selected KDD+ dataset with a total of 125973 records of which 58630 are attack values and 67343 are normal records.

UNSW-BoT-IoT dataset The BoT-IoT dataset is collected from the Cyber Range Lab of UNSW Canberra. The environment with the combination of normal and attack traffic is configured and collected in various formats. The dataset is created in three categories: i) entire dataset with all features, ii) 5% of data with training and testing files with all the features, and iii) 10-best features with training and testing splits. The dataset has been classified with nine types of cyber- attacks and is represented with 46 labeled feature classes. To test the efficiency of the model we have selected a 5% best- featured dataset which has 10,48,457 attack records and 118 normal values [25]. The dataset supports four attack classes as DDoS, DoS, Normal, reconnaissance, and theft. Table 2 displays various attack classes and number of records in each class for both the datasets. 70% of the data is considered as the training data, 15% for the validation, and 15% for the testing. SID-RIS focuses on multi-class classification to train and trace various cyberattacks.

Table 2: Attack class and no of records

BoT-IoT Dataset		KDD-Dataset	
Class	No. of Records	Class	No. of Records
Normal	118	Normal	67343
DDoS	550955	DoS	45927
DoS	471635	Probe	11656
Reconnaissance	25846	R2L	995
Theft	21	U2R	52
Grand Total	1048575	Grand Total	125973

3.2. Data Normalization

As the first step of normalization, we use the fill missing function to replace all the empty values with standard and constant values. In the second step, all the categorical values (NaN) are converted into numerical identities for easy prediction. We have applied one hot-encoding technique for conversion. This technique processes the categorical variable and converts it into a numerical representation. But at the same time natural ordering between categories with integers may result in poor performance or unexpected results, we have converted the string values to a new binary variable and added for each unique integer value. The BoT-IoT dataset contains three categorical features as

prototype, attack category, and sub category (NaN values) which are encoded into numerical form before producing as input to the network model. Normal values are indicated with 0 and the attack values as 1 or any categorical integer value based on the class.

3.3. Feature Extraction

In this study, we are extending our previous experiment [26] in which we have concluded that the model tested with optimal features resulted in a minimum error, compared to the model trained for the entire dataset. In this study, we are using the best subset evaluated from feature reduction techniques. Then we train the sample with Cascading Feed Forward Back Propagation (CFFBP) classification and detection technique. We have selected the 10 best-featured samples provided by the BoT-IoT dataset, and for the KDD dataset, we have used CFS- Subset-Evaluator, a feature reduction algorithm, that results in six best features from 41 labeled values. The CFS subset evaluation technique generates a subset of attributes with the individual predictive ability of each feature, along with the degree of redundancy between them. The resulted features are highly correlated with the target class and have low inter co- relation with other input values. Further to improve the training efficiency and speed up the detection process, we have used the encoded values as input for detection models generated from the Auto Encoder (AE) technique. AE reduces the given input into the lower-dimensional format and regenerates the output as a new representation. To replicate the input vector against the output layer, and train the AE model, we implement a back-propagation algorithm. For a given input X and reconstruction result as x, the network is trained by minimizing the error L (x, $\hat{\boldsymbol{x}}$) to measure the variation between the original input and the encoded output. We have trained AE with 25 hidden layers using the scaled conjugate gradient training algorithm. The model performance is evaluated using Mean Square Error (MSE) with L2 sparsity regularizes, the model results in with 6.66% MSE.

To prevent over-fitting additional information is given to the model in the process of regularization. L2 regression is also considered as ridge regression with the linear regression in Equation 2 and the loss function with L2 norm of the weights represented in Equation 3.

$$\hat{\boldsymbol{x}} = \boldsymbol{w_1} + \boldsymbol{x_1} + \boldsymbol{w_2 x_2} + \cdots + \boldsymbol{w_n} + \boldsymbol{x_n} + \boldsymbol{b}. \quad (2)$$

$$Loss = Error(x, \hat{x}) + \lambda \sum_{i=1}^{N} \quad w_i^2. \quad (3)$$

In the above expression for an auto-encoder model, $\hat{x}$ with x as input variables, w represents the weight, and b represents the bias. We use a loss function to analyze the difference between the true and predicted values. The regularization parameter is represented by $\lambda > 0$ $ and $\sum$ is used to calculate the total loss and predict the efficiency of the model for each input and added weight. The neurons are "inactive" if their output value is close to 0 and active if it is close to 1; we use the sparsity parameter to make it inactive and avoid over-fitting issues. This checks that the average activation of each hidden neuron is close to it, which is a small value close to zero [27].

3.4. Layering in SID-RIS

We have selected Cascading Feed Forward Back Propagation (CFFBP) method to classify the anomaly and identify the attack and normal packets. As with all other network models, Feed Forward (FF) model consists of a single input layer, multiple hidden layers, and selected output layers. Back Propagation (BP) is used as a learning algorithm to train the network models by updating the weights and calculation the error values to propagate the prior layer. The non-linear transfer function of multiple layers allows one to learn both linear and non-linear relations between input and output vectors [28].

Connecting the input weights from each successive layer is the unique property of the proposed model. Networks with multiple layers have the potential to learn the complex relations between input and output vectors. The model begins with a single input layer and adds multiple connected layers one

by one in the process which receives connections from the original input layer and all previously hidden units. A connection from a neuron and multi-layer network is combined with a direct link and shaped through an activation function in the hidden layer [29]. Perceptions are added one by one in this correlation, it starts with a small number and ends up with a bigger size. Additional connections improve the speed and learning rate. The process is terminated when the net performance is accurate. Such network pattern is called as Cascading Forward Back Propagation Neural Network (CFBPNN). The mathematical expression of CFBPNN is given in Equation 4.

$$y = \sum_{i=1}^{n} \quad f^i w_i^i x_i + f^0 (\sum_{j=1}^{k} \quad w_j^0 f_j^h (\sum_{i=1}^{n} \quad w_{ji}^h x_i)). \tag{4}$$

In Equation 4, y represents the output layer, $\sum_{i=1}^{n}$ is used to calculate the sum of weights and bias of each layer. The special feature of this network is to carry forward the calculated weights and bias by establishing a direct relationship between the input and hidden layers using $f^i w_i^i x_i + f^0$An activation function is used to train the complex patterns and take decisions for passing the values for the next layers. Figure 1, represents the internal structure of the cascade model representing the internal connectivity to the weights of the previous layer to next.

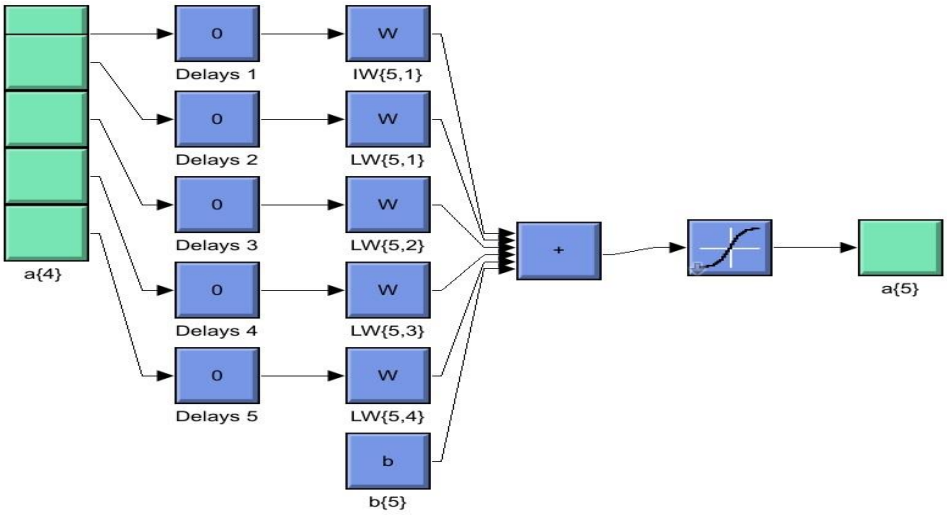


Figure 1: CFBPNN Inner Layer structure

3.5. Experimental Setup

The experiment is conducted on both dataset samples for optimal features adopted from the feature extraction model. The dataset is split into a 70% training set and 15%test set and 15% for validation according to the random data split method. The system is trained using cascading feed-forward network with seven inputs for the KDD+ dataset and 19 input layers for the BoT-IoT dataset. Five hidden layers (10 nodes each) and output layer (1 node) for four attack classes in binary form (0 for normal and 1 for attack) are considered. The model is trained and experimented on an I5 processor (16 GB RAM and 1 TB Octan memory) with a window 10 operating system using MATLAB R2021a environment. Based on the repeated experiments conducted, we have adopted the network model with ideal parameters which produce the highest accuracy and the lowest false rate. We then define the evaluation parameters and finally, discuss the results.

Various parameters used to activate the network are: i) Data division method: Random (dividerand), ii) no. of Epochs: 1000, iii) transfer function: Transit, iv) training method: Levenberg-Marquardt (trainlm), v) adaption learning function: learngdm, and vi) performance indicator: Mean Square Error (MSE). Cascading model is applied and tested for both the data sets with the same parameters given above. The only change in the size of the input layer is based on the number of features available in the dataset.

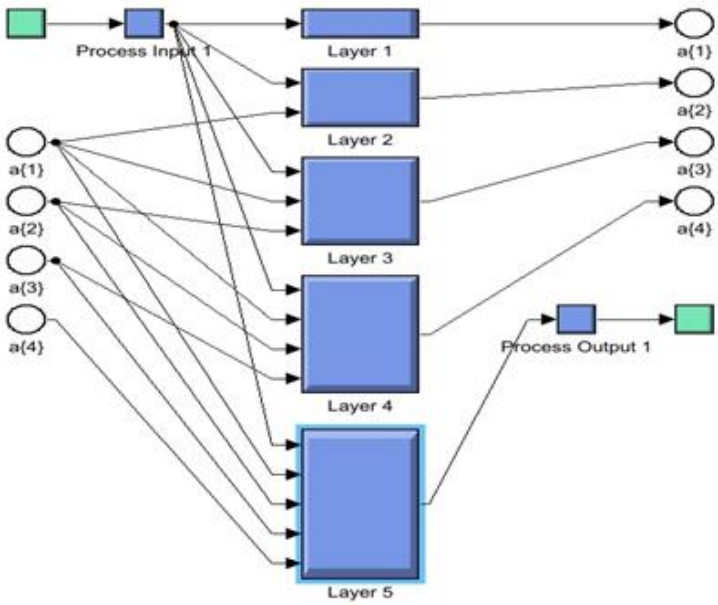


Figure 2: CFBPNN Internal Network structure

Figure 2 displays the internal structure of the CFBPNN model with 19 features assigned to the input layer with 5 hidden neurons. 4 hidden layers with 10 neurons in each and one for the final output layer. Each layer is displayed with the weight and bias added from the previous node. The internal architecture of the CFBP model for the first input layer is indicated as process input 1 with five hidden layers Layer $_{1..5.}$ Training function Transig for each layer calculating the initial input weights and bias received from previous units a1to4 before the hidden layer and *a1*.. *a4* carrying the weights for the next layer represented after the hidden layer structure in Figure 3, and finally, the classification output (four attack classes) represented with process Output single layer.

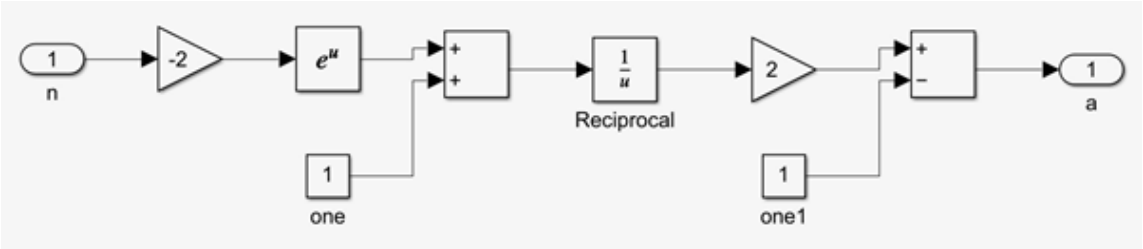


Figure 3: Internal structure of the layer representing Transig function

3.6. SID-RIS Risk Factor analysis

To identify the correlation between input and target variables, we have represented a correlation plot for all the six input variables and one target variable for the subset evaluated from KDD+ dataset. The relation between the variable and the impact is displayed in Figure 4. The diagonal cells represent self-correlation. The last column and the row represent the correlation between the variable in a horizontal and vertical direction. From Figure 4a, we observe that Src-bytes and Dst- Bytes have high impact on the attack types. Diff-RV-rate, srv- error-rate, and logged-in-status variables are having very little impact with zero and negative values for the evaluated subset. Figure 4b represents the Correlation plot (coorplot) for BoT-IoT dataset with 18 input features and one target value with four classes. It is observed that only seven variables have a positive impact on the target variable and the other ten variables have a negative impact on the attack variable. This experiment helps in tracing the most prominent variable having a high impact on the attack variants. This technique can be further enhanced using the script to develop any prevention model.

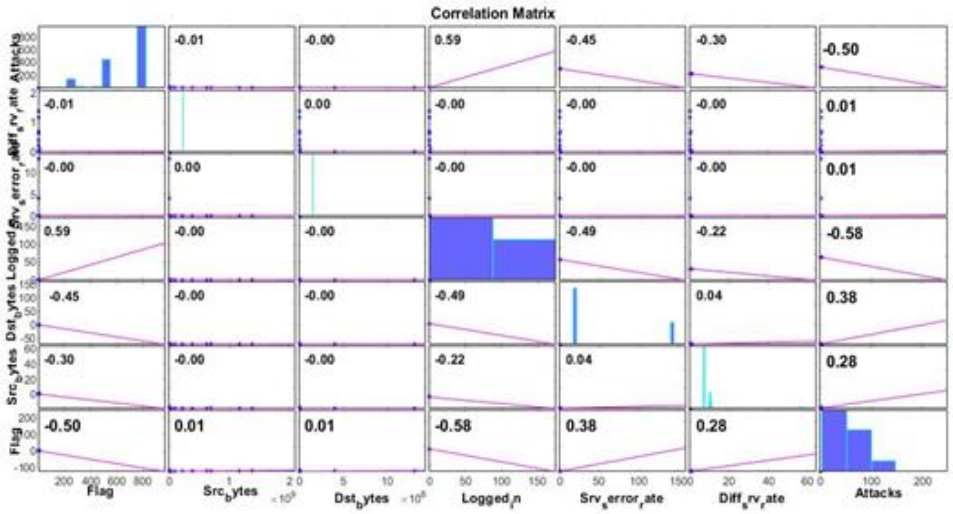
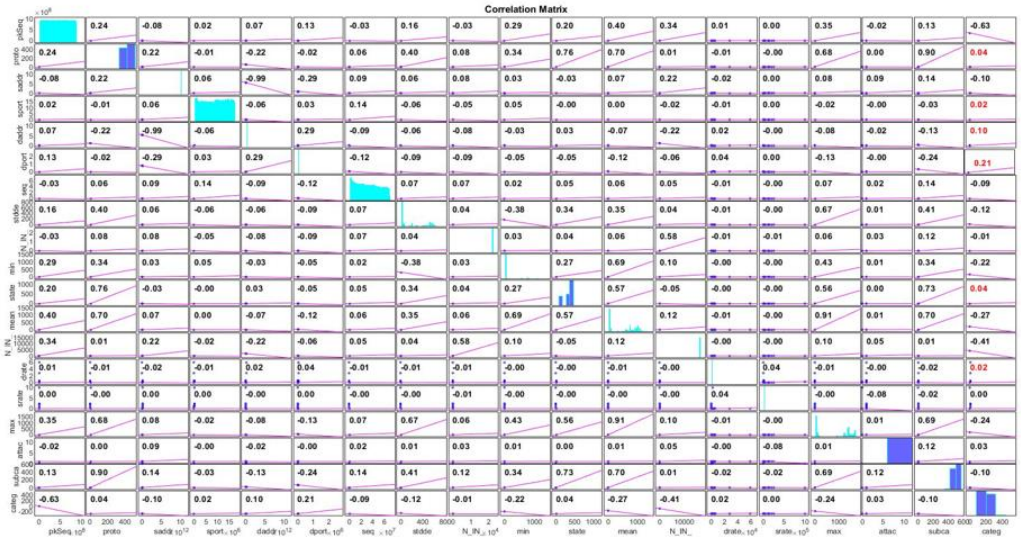


Figure 4(a): Co-relation Plot for risk analysis KDD+ Dataset
Figure 4(b): Co-relation Plot for risk analysis BoT-IoT Dataset.

4. Results and Discussion

The dataset with different input features with variant values is considered for the experiment. The dataset is prepossessed before analyzing self-taught learning on it. Categorical at- tributes with string values are converted into discrete numerical attributes using the one-hot conversion method. As discussed in the methodology section 3.3, optimal features are considered to train and test the model. The testbed is then trained with autoencoders, and the resulted data is used to develop a cascade classification model. These two approaches are applied for the evaluation of NIDSs on the selected samples with a random data split. The proposed model achieved very high accuracy and less false-alarm rates compared to the training implemented for the entire dataset. Cascading model is applied and tested for both datasets with the same parameters. As discussed in [26], the detection model is implemented only for the subset, evaluated with best features. This reduces the training time and results in the highest accuracy. Evaluation metrics project the performance of the model, it helps to determine the capabilities and discriminate the model results. We have tested the model for multi classes and analyzed the results using a confusion matrix.



4.1. Performance Metrics

A confusion matrix is the most appropriate technique to analyze the performance of the classification model. The results of this technique identify the types of errors encountered by the model in the process of training and testing. The number of incorrect predictions is analyzed for each class assigned to the model with the target variable. The difference in the prediction and actual assumptions are projected in the matrix; it also includes the errors made by the classifier and the category which is wrongly analyzed. The elements of the confusion matrix are used to construct the accuracy of the overall model. The formula to calculate each element of the matrix and the precision is displayed in Figure 5.

False Negative Rate (FNR): The ratio of false cases marked as true.
Accuracy (A): The ratio of correctness for the classified samples.
Precision (P): The ratio of the true positive samples to predict the positive samples.
Recall (R): Represent the ratio of true positive values to the total value. This reflects the model's ability to recognize the attacks from a given class.

	Predicted Class		
Actual Class	True Positive(TP)	False Negative (FN)	Recall-(R) =TP/(TP+FN)
	False Positive(FP)	True Negative(TN)	True Negative = TN/(TN+FP)
	Precision-(P) TP/(TP+FP)	Negative Predicative TN(TN+FN)	Accuracy=TP+TN/(TP+TN+F+FN)

Measure	Formula
F-Measure	2xPxR/(P+R)
False positive	FP/(FP+TN)
Error Rate	FP+FN/(TP+TN+FP+FN)

Figure 5: Confusion Matrix – Calculations

Our proposed CFBP-based Neural Network (CFBNN) model is used for mapping the patterns between input and target values. Various compositions of threshold functions are used in the layers with multiple combinations and the final results are projected in Figure 6.

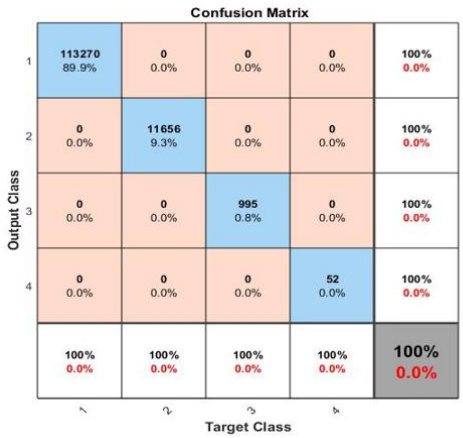


Figure 6(a): KDD+ Dataset

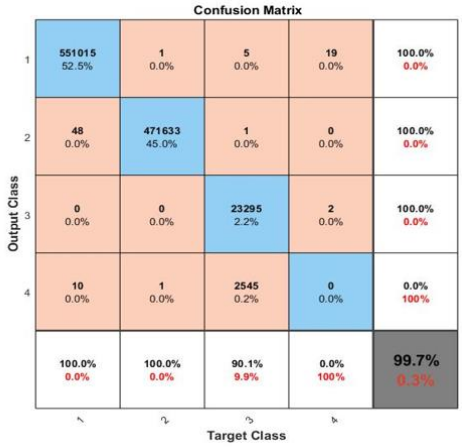


Figure 6(b): KDD+ Dataset

Figure 6: Confusion Matrix - Calculations

The model results in 100\% accuracy for the KDD+ dataset tested on the best features subset, and 99.7\% accuracy for the BoT-IoT dataset. A minimum false rate is observed with 0.3\% for the theft category. The detailed analysis of attack detection ration for each class is given in Figure 7.

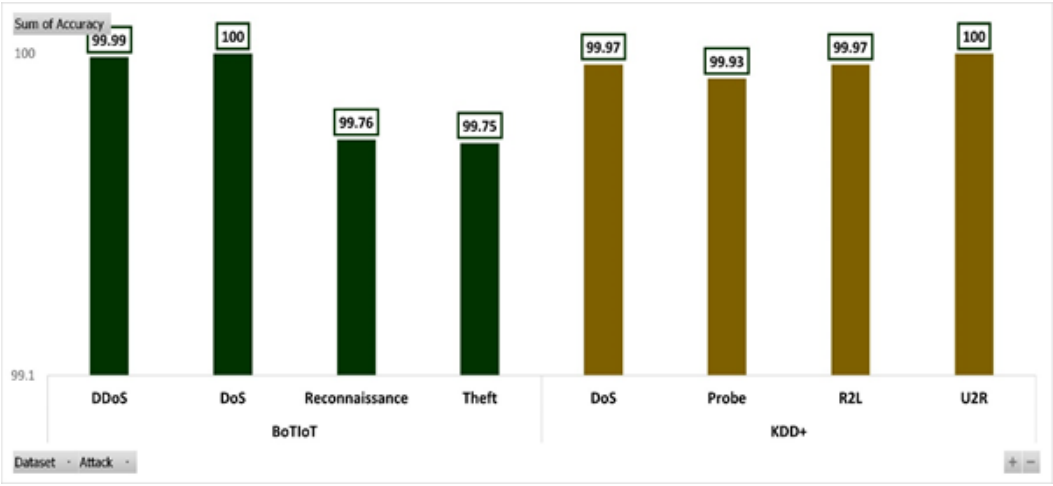


Figure 7: Performance of model for each attack class.

4.2. Comparative Analysis

We have tested and compared the model of both the dataset samples and the result matrix is projected in Table 3. The CFBP model is proved more suitable for detecting the attacks with multi-class classification for both the sample datasets. The observational point is that the model shows excellent results for the KDD+ dataset, in which the testing and training are implemented for the subset generated using the feature selection method. The results for the BoT-IoT dataset are quite good compared to other state of art models, the model is most apt in identifying Dos attacks for various samples. There is a slight variation in identification of reconnaissance and theft attack with 0.1%. A detailed projection of the attack class with the resultant matrix is projected in Table 3.

Table 3: Performance Metrics

KDD+ Dataset					BoT-IoT Dataset				
Attack	Accuracy	Precision	Recall	F1 Score	Attack	Accuracy	Precision	Recall	F1 Score
DoS	100%	1.0	1.0	1.0	DDoS	99.99	1.0	1.0	1.0
Probe	100%	1.0	1.0	1.0	DoS	100	1.0	1.0	1.0
R2L	100%	1.0	1.0	1.0	Reconnaissance	99.76	1.0	0.9	1.0
U2R	100%	1.0	1.0	1.0	Theft	99.75	0.0	0.0	0.0

The comparison with existing systems is shown in Table 4. The rule-based Decision tree (TDTIDS) model has the highest accuracy of all the existing models [18] with 99.98%. CFBP model has the benchmark of 100% in the identification of all class attack values. The DBN [12] and RNNIDS [22] models have a poor performance comparatively, while the other models' performance are very close to each other. The multi-class feed forward neural network [21] has a very close accuracy with our CFBNN with a raised recall score but, CFBNN has less FPR which indeed reduces the scope of error. CFBNN performs much better than [14],[15],[22],[18] on the basis of accuracy, precision, and false rate which are the most important metrics for a detection system. The average false-positive ratio of our model is lower than all these models. However, the disadvantage with the regular deep learning techniques is to determine values with the next layer but, our cascade model has the advantage of processing the previous weight and bias values to the next hidden layers; this improves the detection rate and reduces the error factor. The main goal of CFBNN is to improve the detection rate and reduce the error rate which is successfully accomplished with the six selected features executed in the CFBNN model. The test proves with 100% accuracy and 0% false rates.

Table 4: Comparison of DL- based IDS models for IoT

Author and Reference	Technique	Data set	Accuracy
Zhang et al. [12]	DBN for anomaly detection in IoT mobile network	Simulated data	94%.
Thamilarasu G. et al. [14]	three phase model with DBN and DNN	Real-time	97%.
S. Smys et al. [15]	Hybrid Convolutional Neural Network	UNSW NB15	98.6%.
Mohamed Amine Ferrag et al. [18]	RDTIDS: Rules and Decision Tree-Based Intrusion Detection System	CICIDS2017	96.995%.
Abdelouahid Derhab et al. [19]	Temporal Convolution Neural Network (TCNN) with Synthetic Minority Oversampling Technique-Nominal Continuous (SMOTE-NC)	Bot-IoT	99.998%.
Alkahtani.H et al. [20]	Hybrid convolution neural network with the Long Short-Term Memory (CNN-LSTM)	IoTID20	98.80%.
Mengmeng.Ge et al. [21]	Multiclass Feed-Forward Neural Networks (FNN)	BoT–IoT	999.79%.
Qureshi et al. [22]	Random Neural Network -IDS (RNNIDS)	NSL-KDD	95.25%.
Fatani A et al. [23]	Deep learning and Meta Heuristics (MH) algorithms	KDD Cup	99.62%.
Proposed Model	Cascade Forward Back Propagation	KDD+,BoT-IoT	100%,99.7%

5. Conclusion

We propose SID-RIS, an intrusion detection model for IIoTs. The purpose of the present study is to improve the detection solution for IoT and IIoT devices and establish an accurate monitoring environment that handles unsafe structure and detect abnormal behavior. SID-RIS is based on deep learning. It classifies the given input based on cascading forward method. A CFS-subset evaluation technique is used to select the optimal features from KDD+ dataset and then process the subset for the training detection model. The model is examined on both KDD+ and BoT-IoT data set and evaluated using the confusion matrix. Our solution, CFBNN achieves better performance in terms of accuracy 100% for the KDD+ dataset, and 99.7% accuracy and 1.1% false rate for BoT-IoT data set for multi-class classification. To identify the risk factor, we have implemented a co-relation plot to trace the impact of the variable with the target and identified three variables for the KDD+ dataset and seven for the BoT-IoT dataset. In the future, we would like to extend our work to experiment with prevention techniques for other open datasets to generalize the results.

6. References

[1] Barzashka, I., 2013. Are Cyber-Weapons Effective? Assessing Stuxnet's Impact on the Iranian Enrichment Programme. The RUSI Journal, 158(2), pp.48-56.

[2] Shea, J., 2017. NATO: Stepping up its game in cyber defence. Cyber Security: A Peer-Reviewed Journal, 1(2), pp.165-174.

[3] Tsiknas, K., Taketzis, D., Demertzis, K. and Skianis, C., 2021. Cyber Threats to Industrial IoT: A Survey on Attacks and Countermeasures. IoT, 2(1), pp.163-188.

[4] Muraleedharan, R.; Osadciw, L. Cross layer denial of service attacks in wireless sensor network using swarm intelligence. In Proceedings of the 2006 40th Annual Conference on Information Sciences and Systems, Princeton, NJ, USA, 22–24 March 2006; pp. 1653–1658

[5] Formby, D.; Beyah, R. Temporal execution behavior for host anomaly detection in programmable logic controllers. IEEE Trans. Inf.Forensics Secur. 2020, 15, 1455–1469

[6] Demertzis, K.; Kikiras, P.; Tziritas, N.; Sanchez, S.; Iliadis, L. The next generation cognitive security operations center: Network flow forensics using cybersecurity intelligence. Big Data Cogn. Comput. 2018, 2, 35
[7] Yan, X.; Xu, Y.; Cui, B.; Zhang, S.; Guo, T.; Li, C. Learning URL embedding for malicious website detection. IEEE Trans. Ind. Inform. 2020, 16, 6673–6681.
[8] Stalmans, E.; Irwin, B. A framework for DNS based detection and mitigation of malware infections on a network. In Proceedings of the 2011 Information Security for South Africa, Johannesburg, South Africa, 15–17 August 2011; pp. 1–8.
[9] Butun, I.; Osterberg, P.; Song, H. Security of the internet of things: Vulnerabilities, attacks, and countermeasures. IEEE Commun Surv. Tutor. 2020, 22, 616–644.
[10] Yazan Otoum, Dandan Liu, and Amiya Nayak. "DL-IDS: a deep learning–based intrusion detection framework for securing IoT". In: Transactions on Emerging Telecommunications Technologies (2019), e3803
[11] Olakunle Ibitoye, Omair Shafiq, and Ashraf Matrawy. "Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks". In:2019 IEEE Global Communications Conference (GLOBECOM). IEEE. 2019, pp. 1–6.
[12] Ying Zhang, Peisong Li, and Xinheng Wang. "Intrusion detection for IoT based on improved genetic algorithm and deep belief network". In: IEEE Access7 (2019), pp. 31711–31722
[13] Monika Roopak, Gui Yun Tian, and Jonathon Chambers. "Deep learning models for cybersecurity in IoT networks". In:2019 IEEE 9th annual computing and communication workshop and conference (CCWC). IEEE. 2019, pp. 0452–0457
[14] Thamilarasu, G. and Chawla, S., 2019. Towards deep-learning-driven intrusion detection for the internet of things. Sensors, 19(9), p.1977.
[15] Smys, S., Abul Basar, and Haoxiang Wang. "Hybrid intrusion detection system for internet of Things (IoT)." Journal of ISMAC 2, no. 04 (2020): 190-199.
[16] Balakrishnan, N., Rajendran, A., Pelusi, D. and Ponnusamy, V., 2021. Deep Belief Network enhanced intrusion detection system to prevent security breach in the Internet of Things. Internet of things, 14, p.100112.
[17] Liang, C., Shanmugam, B., Azam, S., Karim, A., Islam, A., Zamani, M., Kavianpour, S. and Idris, N.B., 2020. Intrusion detection system for the internet of things based on blockchain and multi-agent systems. Electronics, 9(7), p.1120.
[18] Ferrag, M.A., Maglaras, L., Ahmim, A., Derdour, M. and Janicke, H., 2020. Rdtids: Rules and decision tree-based intrusion detection system for internet-of-things networks. Future internet, 12(3), p.44.
[19] Derhab, A., Aldweesh, A., Emam, A.Z. and Khan, F.A., 2020. Intrusion detection system for Internet of Things based on temporal convolution neural network and efficient feature engineering. Wireless Communications and Mobile Computing, 2020.
[20] Alkahtani, H. and Aldhyani, T.H., 2021. Intrusion detection system to advance internet of things infrastructure-based deep learning algorithms. Complexity, 2021.
[21] Ge, M., Syed, N.F., Fu, X., Baig, Z. and Robles-Kelly, A., 2021. Towards a deep learning-driven intrusion detection approach for Internet of Things. Computer Networks, 186, p.107784.
[22] Qureshi, A.U.H., Larijani, H., Ahmad, J. and Mtetwa, N., 2019, July. A heuristic intrusion detection system for Internet-of-Things (IoT). In Intelligent computing-proceedings of the computing conference (pp. 86-98). Springer, Cham.
[23] Fatani, A., Abd Elaziz, M., Dahou, A., Al-Qaness, M.A. and Lu, S., 2021. IoT Intrusion Detection System Using Deep Learning and Enhanced Transient Search Optimization. IEEE Access, 9, pp.123448-123464.
[24] KDD Cup 99, "http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
[25] Koroniotis, Nickolaos, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset." Future Generation Computer Systems 100 (2019): 779-796.
[26] Jayalaxmi, P.L.S., Saha, R., Kumar, G. and Kim, T.H., 2021. Machine and deep learning amalgamation for feature extraction in Industrial Internet-of-Things. Computers \& Electrical Engineering, p.107610.

[27] Kamalov, F., Zgheib, R., Leung, H.H., Al-Gindy, A. and Moussa, S., 2021, October. Autoencoder-based Intrusion Detection System. In 2021 International Conference on Engineering and Emerging Technologies (ICEET) (pp. 1-5). IEEE.
[28] Qiao, J., Li, F., Han, H. and Li, W., 2016. Constructive algorithm for fully connected cascade feedforward neural networks. Neurocomputing, 182, pp.154-164.
[29] Budi Warsito, Rukun Santoso, Hasbi Yasin, et al. "Cascade forward neural network for timeseries prediction". In:Journal of Physics: Conference Series. Vol. 1025. 1. IOP Publishing.2018, p. 012097.