Received: February 13, 2021. Revised: March 29, 2021.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Detecting Internet of Things Attacks Using Post Pruning Decision Tree-Synthetic Minority Over Sampling Technique

M. Ganesh Karthik[1]* **M. B. Mukesh Krishnan[1]**

[1]*Department of Computer Science and Engineering,*
SRM Institute of Science and Technology (SRMIST), Chennai, India
* Corresponding author's Email: ganeshkarthik16@gmail.com

Abstract: In recent decades, the Internet of Things (IoT) is a growing technology in smart applications, where it is highly susceptible to security breaches and the resources are constrained in nature. Hence, the growth of IoT devices allows the hackers to take benefit of the communication capabilities to detect different types of attacks such from NSL-KDD such as Denial of Service (DoS), Probe, Remote to Local (R2L) and User to Root (U2R) attacks. The existing deep networks used larger data in training because the size of non-predictive parameters for learning required huge potential to attack detection. In order to overcome the problem occurred in the existing models, an efficient Post Pruning Decision Tree-Synthetic Minority Over-Sampling Technique (PPDT-SMOTE) is proposed in the research work. The proposed PPDT-SMOTE eliminates or pruned the non-predictive parameters from the huge data samples and SMOTE solves the data imbalance problems as they use the prominent samples from the non-pruned regions. Thus the PPDT models improves the learning rate of the system for huge data and SMOTE will overcome the problem of class imbalance problem as the learning rate is improved. The results obtained from the proposed PPDT-SMOTE approach effectively detects the DoS, Probe, U2R, and R2L attacks in terms of accuracy as 98.04% better when compared to the existing shallow models of 95.22%, Routing Protocol for Low Power and Lossy Networks (RPL) of 91.5 % and Multi-Convolution Neural Network (CNN) fusion model of 64.81% in the IoT environment.

Keywords: Denial-of-service, Imbalance data, Internet of things, Post pruning decision tree, Synthetic minority over-sampling technique.

1. Introduction

IoT devices are increasing rapidly and there are 8.4 billion IoT devices used worldwide as per the survey taken in 2017, also it estimated to reach 21.4 billion by 2021. The vulnerabilities include open telnet ports, Unencrypted transmission, outdated Linux firmware affects the sensitive data [1, 2]. The existing models were involved in intruder detection for IoT system faced huge security issues, so effective security approaches were required in the IoT environment. The IoT is prone to distinct security issues in the internet infrastructure that was taken place during information exchange in heterogeneous networks [3-5]. An intelligent management system is required for DoS attack detection occurred due to malicious intrusions. The attacks are detected successfully will threaten sensitive data in the network through the internet [6]. Recently, the existing methods that were undergone for Dos attacks revealed loopholes in IoT at the first stage. However, precautions were needed to be taken for the IoT devices for the majority of sources accomplished for DoS attacks [7]. The IoT is necessary for developing advanced mechanisms to provide security levels during cyber-attack mitigation [8]. The IoT devices face the challenge to provide security for the sensitive information and also commercial IoT low-end devices were not used for supporting strong security mechanisms that targeted the malicious networks in different attack detection [9]. Therefore, in the proposed PPDT-SMOTE eliminates or pruned the non-predictive parameters from the huge data samples and SMOTE solves the data imbalance problems as they use the prominent samples from the

non-pruned regions. Thus the PPDT models improves the learning rate of the system for huge data and SMOTE will overcome the problem of class imbalance problem as the learning rate is improved using the NSL-KDD dataset for attack classification. The NSL KDD data set is a benchmark data set used in the researches for Intrusion Detection techniques in classifying attacks as normal and abnormal classes [10]. Also, the same dataset is used for classifying the attacks into four classes such as DoS, Probe, R2L and U2R attacks. The NSL-KDD data set has huge number of redundant records that causes the learning algorithms to be biased towards the frequent records and thus prevent them from learning infrequent records which are harmful to networks. The present research uses NSL-KDD dataset to compare machine learning algorithms such as existing shallow models with the proposed PPDT-SMOTE that performs binary classification (normal or attack) and multi classification identifies the typical attacks such as DoS, Probe, R2L, and U2R.

The organization of the paper is given as follows: Section 2 describes the literature survey of the existing methods and Section 3 presents the proposed PPDT-SMOTE method. The results and discussion of the proposed PPDT-SMOTE method are illustrated in Section 4. The conclusion and future work of the research is explained in Section 5.

2. Literature review

The existing methods that were involved in attacks detection were as follows:

Roldán [11] developed an intelligent architecture in the IoT environment for detecting security attacks using Integrated complex event processing and machine learning algorithms. The existing models faced the challenge during detection of attacks such as malware, privacy breaches and denial of service attacks. Therefore, to overcome the problem that occurred in the existing model, an intelligent architecture with Complex Event Processing (CEP) was integrated. The classification of attacks was done by using Machine Learning (ML) algorithms that validated the ability in attack detection was made by malicious attacks. However, the developed model failed to predict automatically in finding the features and also obtained a lower rate of accuracy values because of fewer data training.

Qureshi [12] developed a Secure Attacks Detection Framework for Smart Cities Industrial IoT. The existing methods were used for a susceptible variety of security concerns that were needed to be addressed. Therefore, in the developed RPL detecting HELLO-Flood attack, Sinkhole attacks, and black hole attacks were needed to be detected. However, in the developed model the detection of attacks with similar scenarios was easier whereas detection of attacks for distinct scenarios was difficult.

Shafiq [13] developed a Selection of effective machine learning algorithm for detecting Bot-IoT attacks in IoT smart city. The existing methods faced problems during building the security endeavour for cyber-attacks that identified the traffic using ML algorithms in providing security in IoT. To overcome such issues, the developed hybrid algorithm used the BoT (Botnet) IoT dataset from where the features were extracted for attack classification. The ML algorithm was effective and used to detect IoT anomaly for traffic identification detection. The results showed that the developed model evaluated the performance for Bot-IoT attacks identification. However, the developed model failed to detect IoT anomalies for identifying the attacks during traffic detection.

Mehmood [14] developed the Naïve Bayesian (NB) classification technique, which is a Multi-Agent System (MAS) enriched for IoT security against DoS attacks. The main aim of the developed model was to focus and protected IoT infrastructure from DoS attacks that were attacked by the generated intruders. The developed model used a NB classification algorithm for Intrusion Detection System (IDS) detection deployed to form multi-agents throughout the network. The NB classification algorithm practiced with multiple agents detected the DoS attacks obtained better performance when compared with traditional IDs. The developed scheme aimed securely at the IoT network layers that detected the malicious attacks. However, due to the distributed agents on MAS, the load was distributed among the network participants and the prevention, detection of attacks was slow which needed to be enhanced further.

Liu [15] developed an Efficient DoS attack mitigation for state full forwarding in IoT. In the existing models, the varietal DoS was developed that showed complications for state operations that were exhausted during forwarding nodes. In the developed model, a game model was used to analyse the attack significantly among the defender and attacker. The proposed model used defender for obtaining more significantly managed the expired state entries for full forwarding. The developed model performed an enhanced Distributed Low-Rate Attack Mitigating (eDLAM) mechanism for attack detection. The developed eDLAM was used to maintain the lightweight Malicious Request Table (MRT) that detected the attacks for small to offload burned the practical stated table. However, the developed

eDLAM expired the state entry numbers and less distance indicated that larger size was required during state table forwarding.

Li [16] developed a robust detection for network intrusion of industrial IoT based on multi-CNN fusion. The developed deep learning approach was used for intrusion detection using the multi-CNN fusion method which classifies the attacks using the NSL-KDD dataset. The experiments showed that the performance of the multi-CNN fusion model classified the attacks into normal, DoS, Probe, R2L, and U2R. However, the developed model did not focus on protecting data in industrial-related IoT applications.

Abeshu Diro [17] developed Distributed attack detection scheme using deep learning approach for IoT. The existing models with advanced mechanisms face difficulty of detecting small mutants of attacks over time. In order to overcome the problem occurred in the existing model, the developed deep learning approached showed excellence for traditional machine learning schemes using Softmax for data classification classified into normal or attack. However, the deep learning model consumed more time for training as the parameters size of learning was more.

3. Proposed method

The block diagram of the proposed PPDT-SMOTE method is shown in Fig. 1. The data from the NSL-KDD datasets use label encoding and one significant encoding approach as pre-processing techniques. The pre-processed data is undergone for the process of feature selection to select the prominent ones using the Recursive Feature Elimination (RFE) technique. The classification is performed for the selected features into 4 classes such as DoS, Probing, R2R and U2R attacks.

3.1 Dataset

The data collected from NSL-KDD datasets are used as the most common dataset for research in an IoT environment. The proposed PPDT-SMOTE uses different parts of the NSL-KDD dataset for data duplications and data redundancies. The NSL-KDD dataset includes 41 attributes, which are labelled normal connections or attack types. The NSL-KDD dataset is used in the research as it is better when compared with KDD'99 datasets as the number of records includes training and testing data which is shown in table 1.

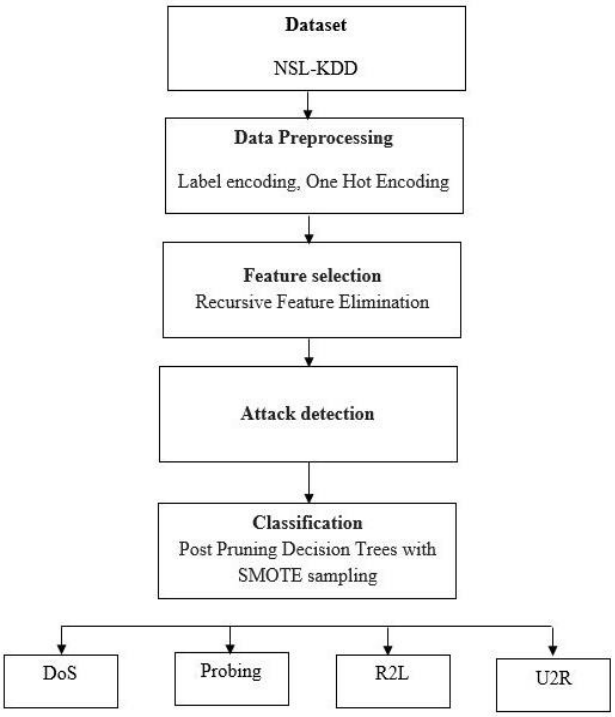

Figure. 1 Block diagram of the proposed PPDT-SMOTE technique

An advantage of using the NSL-KDD dataset in the proposed method is to run the simulations using a dataset without a selection of a small portion. The evaluation of results is consequently different for the research work that will be consistent and comparable. The NSL-KDD contain four attacks such as DoS, U2R, R2L, and probe Attack, which are detailed as follows:

Probe attack: The probe attack is affected due to misused information that weakens the strength of the network. The probe attacks include Portsweep, Satan, Ipsweep, Mscan, Saint, and Nmap.

R2L: By transmitting the packets to the machine from the user detects the weakness in the network. Several attacks presented in R2L are Snmpget attack, Send mail, Phf, Snmpguess, Warez client, Guess-Password, Ftp-write, Multihop, Xsnoop, Httptunnel, Spy, Xlock, Imap, and Warezmaster.

U2R: U2R gets access to the root account once the ordinary account needs to be set up. Some of the attacks in U2R are Buffer-overflow, Load module, Perl, Sqlattack, Xterm, Rootkit, and Ps.

DoS: This attack is increased due to network traffic usage, and the service cannot be provided by the system for DoS attack detection. The types of attacks in DoS are Neptune, Apache2, Udp storm, Back, Land, Smurf, Teardrop, Worm, and Pod.

Table 1. Statistical information about the NSL-KDD dataset

	Abnormal				Normal	Total
Dataset	DoS	Probe	R2L	U2R		
KDD Train +	45927	11656	995	52	67343	125973
KDD Test+	7458	2754	2421	200	9711	22544

3.2 Data preprocessing

After the acquisition of data from the NSL-KDD dataset, data pre-processing is accomplished using label encoding and one hot encoding. Usually, the labelled data will be not in the machine-readable format and needs to be converted to a numerical format. Therefore, the unstructured data will be converted into structured data using pre-processing techniques which are given as follows:

3.2.1. Label encoding

Label encoding is a process that is used to encode the categorical values by converting each value present in a column into a number. There are many columns in the dataset that are having the column names with the desired bridge-type values. As there were datasets used for understanding label encoding, categorical values are only needed to be focused on. The proposed PPDT- SMOTE approach requires the categorized datatype having the label 'category'.

3.2.2. One hot encoding

One hot encoding allows the representation of categorical data to be more expressive and the categories must be converted into numbers. Thus, the ordering issues are addressed using a One-Hot Encoding approach. The One hot encoding addresses the category value that is converted to a column assigned with a value as 0 or 1. The first column value (Arch/None) will be having the row values that indicate true and value as '1'. The other values in columns indicate false which is represented as '0'. If in case the rows are matching with the column values then the whole values are assigned with 0 and 1. Thus, the conversion of labels into a numeric value is performed using pre-processing one hot encoding technique. The pre-processed data is undergone for the process of feature selection using the Recursive Feature Elimination technique.

3.3 Feature selection using recursive feature elimination

RFE is a popular feature selection algorithm that has the advantage of configuring easily and is used for selecting the relevant features during dataset training. Based on relevant data, the target variable values are also predicted. There are 2 important configurations that uses RFE and perform the following functions for using the hyper parameters:

- The choice in the number of features to select the most relevant data in predicting the target variable.
- The choice of the algorithm used to help choose features in predicting the target variable.

Based on the above-mentioned choice, the performance of the proposed method is dependent strongly on hyper parameters and configuring well. The samples are present in rows and the features are present in the columns will faces problems in the domain. The selection of features is done based on the relevant features from the column dataset. The various machine learning algorithms are used for running efficiently with less space and complexity effectively. The existing models utilized the irrelevant features for the classification that lowered the predictive performances. The present research uses RFE that will select the features by choosing and selecting the trained features appropriately until the number of the desired removal of features are successful.

The process is achieved by fitting a machine learning algorithm using the core model ranked the features based on discarding the important features, refitting model. The present research uses RFE that ranked the predicted features based on the numbered priority. Each of the predictors fitted is ranked based on the sequence of ordered numbers which is having the candidate values. The pseudo-code for RFE is shown below that explains the process of feature extraction. The sequence of ordered numbers S is used to predict the ranking based on their importance. The candidate values for the predictors retain the values that are an ordered sequence of numbers as $(S_1 > S_2, \dots S_i)$. For each of the iteration, feature selection is performed for the ranked predictors which are topped at S_i and the model refits the assessed performance. The best performance for the proposed PPDT-SMOTE method is determined and the top predictors finally fit the model.

Pseudocode for RFE

Train the model on the training set for all predictors
The model performances are calculated
Calculate the importance of the variables
For each subset size S_i where $i = 1 \ldots S$
Do
Keep the important variables S_i thereby preprocess the data
Train the model using S_i predictors
Calculate the performance of the model
End
Calculate the performance over S_i
The appropriate number of predictors are determined.

3.4 Post pruning decision trees with SMOTE

The Decision Tree (DT) induction process has two major phases:

- Growth phase
- pruning phase

The pruning phase is generalized based on the growth phase of DT to overcome the problem of overfitting when data have undergone training. The post pruning process addresses the issues occurred when the accuracy is maximized and validated. The proposed PPDT-SMOTE aimed at generalizing the DT at the growth phase which overcomes the problem of overfitting when data is trained. The overfitting problems have occurred when the model memorizes the data used during training so it has learning noise at the top of the signal. The under fitting problems are used for finding the best patterns in the data. The size of the decision tree is reduced by using the pruning process which removes the tree parts for data classifying instances.

Post-pruning

Post pruning is the most commonly used process for tree simplification. The nodes and the subtrees present will be replaced with leaves thereby overcome the problem of complexity. The pruning process reduces the size of the data and also improves the accuracy for the unseen objects during classification.

Bottom-up pruning

Bottom-up pruning is the procedure that starts at the tree's last node and follows the recursively upwards that will determine each and every individual node's relevance. If the classification process is not completed, then the dropped node will be replaced with the leaf.

Top-down pruning

In contrast, the top-down method is used for starting the process at the root of the tree. The relevance check process will be carried out that decides whether the relevant data is used for item classification.

The feature values obtained from the PPDT are embedded for SMOTE that overcomes the problem of imbalanced classification that develops predictive models on classification datasets for severe class imbalance. The imbalanced datasets face the challenge using the machine learning technique obtains the poor performance for the minority classes. The test set is applied for the imbalanced dataset that yields an accuracy optimistically. The classifier is assigned with a single test case that assigns the majority class obtains the best accuracy and equals the test case proportion for the majority class. The present research uses random values generated by under-sampling trimmed the number of samples in the majority class and used SMOTE for oversampling in minority class distribution. The proposed PPDT-SMOTE method classifies the attack into binary classes such as normal or abnormal attacks. Also, the same dataset NSL-KDD is used for multi attacks classification of four, such as DoS, Probe, L2R and U2R attacks. An advantage of using PPDT is the linear computational complexity as it has the ability to test smaller than the training set better. The PPDT calculates error cost of each node and the error cost at the node is calculated by using the below Eq. (1).

$$R_c(t) = r_m(t) \times r_e(t) \tag{1}$$

where $R_c(t)$ is the error cost of each node of PPDT.

$$r_m(t) = \frac{Number\ of\ examples\ misclassified\ in\ node}{Number\ of\ all\ examples\ in\ node} \tag{2}$$

$$r_e(t) = \frac{Number\ of\ examples\ in\ node}{Number\ of\ total\ examples} \tag{3}$$

If the parameters that are not useful are checked as each node and if node t was not pruned then error cost of subtree $R_c(T)$ are rooted at t is defined by using the below Eq. (4).

$$R_c(T) = \sum_{i=number\ of\ leaves} R_c(i) \quad (4)$$

The pseudo code for the proposed PPDT-SMOTE is as shown below.

Pseudo code for PPDT-SMOTE

Input: Features
Output: PPDT with SMOTE function
The initial decision tree is generated
For each node,
If node is parent node, then
The total leaf risk rate (R_l) and parent risk rate (R_p) are computed
If ($R_p > R_l$)
The parent node is converted to the leaf
Calculate the Error cost using Eqs. (1) to (4)
Endif
Endif
Endfor
Return the final tree

4. Results and discussion

The results of the proposed PPDT-SMOTE are simulated by Anaconda navigator and python 3.6 software with Windows 10 operating system, 128 GB RAM, 1 TB memory, 22 GB configured with RTX 2080 Ti GPU and i9 processor. In this research work, the proposed PPDT-SMOTE model is compared with a benchmark model to validate the overall performance.

The results of the proposed PPDT-SMOTE method evaluated using the following parameters.

- **Accuracy**

Accuracy is defined as the ratio of correctly predicted observations to the total number of observations. The accuracy is calculated by using Eq. (5).

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \times 100 \quad (5)$$

- **Precision**

The proportion of positively or correctly identified samples is defined using the precision, which is given in Eq. (6).

$$Precision = \frac{TP}{TP + FP} \times 100 \quad (6)$$

- **Recall**

When the actual number of traditional fault-prone models are considered, the ratio of correctly predicted as fault-modules is defined as recall. The proportion of actual positives or correctly predicted is known as recall, which is shown in Eq. (7).

$$Recall = \frac{TP}{TP + FN} \times 100 \quad (7)$$

- **F 1-Measure**

The harmonic mean of recall and precision is defined as F1-Measure, which is shown in Eq. (8).

$$F1 - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100 \quad (8)$$

- **False Alarm Ratio**

A False Alarm Ratio (FAR) is defined as the number of false alarms per the total number of warnings or alarms in a given study or situation, which is given in Eq. (9).

$$FAR = \frac{Number\ of\ False\ Alarms}{Total\ Number\ of\ warnings} \quad (9)$$

- **Area Under the Curve**

Area Under the Curve (AUC) provides an aggregate measure of performance across all possible classification thresholds calculated using the Eq. (10).

$$AUC = \int_a^b f(x)\,dx \times 100 \quad (10)$$

The AUC is determined using the curve equation $y = f(x)$ that ranges among $x = a$ and $x = b$. The integration of the function operating among the limit $x = a$ and $x = b$. Areas under the x-axis will be a negative area and above the x-axis will be positive.

FN is the number of False Negatives
TP is the number of true positives
FP is the False Positive
TP is the True Positive

Table 2. The performance measures obtained for the proposed PPDT-SMOTE for binary classes in terms of precision, recall, F-measure, accuracy, AUC and FAR

Model Type Class	Precision (%)	Recall (%)	F1 Measure (%)	Accuracy (%)	AUC (%)	FAR
Ada Boost Classifier	92.55	87.22	90.04	89.00	95.90	0.23
Stochastic Gradient Descent Classifier	83.23	71.68	83.75	81.99	90.95	0.26
Proposed PPDT-SMOTE	97.03	98.05	96.97	96.52	98.61	0.19

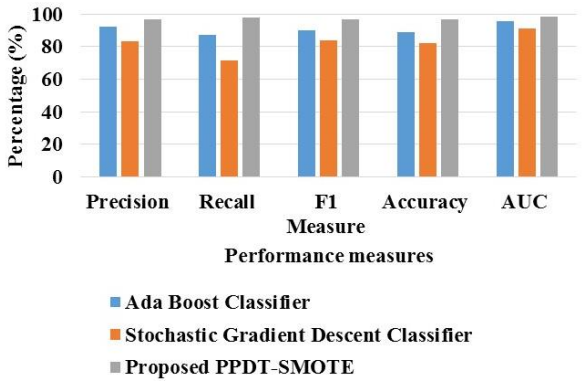

Figure. 2 Graphical representations for the proposed PPDT-SMOTE for binary classes in terms of precision, recall, accuracy, AUC and F-measure

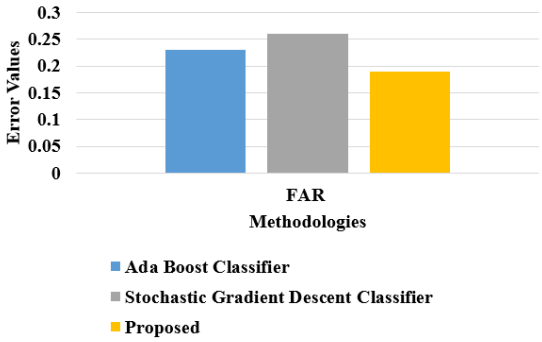

Figure. 3 Graphical representations for the proposed PPDT-SMOTE for binary classes in terms of FAR

4.1 Quantitative Analysis for NSL-KDD dataset

The results obtained for the proposed PPDT-SMOTE model in terms of the performance measures are presented in table 2. The existing models Adaboost classifier and Stochastic Gradient Descent Classifier obtained the precision values as 92.55 %, 83.23 % and 97.03 %. Similarly, the Recall values obtained are 87.22 %, 71.68 % and 98.05 %. The values obtained for the F-measure are 90.04 % for the Adaboost classifier, 83.75 % for the Stochastic Gradient Descent classifier and the proposed method obtained an F-measure of 96.97 %. The graphical representation for Table 2 is shown in Fig. 2. The results obtained for the binary attacks classes showed that the proposed method obtained better results in terms of Precision, Recall and F-measure when compared to the existing methods.

Table 2 shows the tabulation of the performance measures obtained for the proposed PPDT-SMOTE that are expressed in terms of AUC and accuracy. The Accuracy values obtained for the Adaboost classifier were 89%, the Stochastic Gradient Descent classifier obtained an accuracy of 81.99 % whereas the proposed PPDT-SMOTE obtained an accuracy of 96.52 %.

Similarly, the AUC values obtained for the Adaboost classifier was 95.90 %, the Stochastic Gradient Descent classifier obtained AUC of 90.95 % whereas the proposed PPDT-SMOTE obtained the AUC of 98.61 %. Table 2 shows the values obtained for the proposed method in terms of Accuracy and AUC was better when compared to the existing methods. The graphical representation of table 2 is shown in Fig. 2.

Table 2 shows the values for the FAR in terms of error values for the existing Ada Boost Classifier and Stochastic Gradient Descent Classifier. The FAR values obtained for them are 0.23 and 0.26. Whereas the proposed PPDT-SMOTE obtained FAR of 0.19 is better when compared to the existing methods. The graphical representation for table 2 is shown in Fig. 3. The results obtained for the proposed PPDT-SMOTE model in terms of the performance measures are presented in table 3. The results obtained for multi-classification to NSL-KDD are evaluated and shown in table 3.

The proposed PPDT-SMOTE model obtains, Precision of 97.12 % for DoS, 98.46% for probe, 95.05 % for, and 99.31 % for U2R. The proposed method obtained the Recall of 98.46 %, 98.46 %, 94.54 %, and 99.28% for DoS, Probe, R2L, and U2R respectively. The proposed method obtained the F-measure of 97.78 %, 98.46 %, 94.79 %, and 99.31% for DoS, Probe, R2L and U2R respectively. The Ada Boost Classifier failed to improve the overall network security postures by analysing with other network traffic datasets. Similarly, the stochastic gradient

Table 3. Results evaluation for the proposed PPDS-SMOTE for multi classes in terms of Precision, Recall and F-measure

Model Type Class	Class	Precision (%)	Recall (%)	F1 Measure (%)	Accuracy (%)	AUC (%)	FAR
Ada Boost Classifier	DoS	93.47	92.32	92.89	93.87	98.65	0.16
	Probe	93.18	96.15	94.54	96.39	98.67	0.19
	R2L	85.12	80.55	81.96	88.09	95.32	0.22
	U2R	90.45	73.16	78.78	99.31	50.00	0.01
Stochastic Gradient Descent Classifier	DoS	94.04	78.18	86.53	89.82	86.65	0.15
	Probe	90.45	89.68	88.11	92.33	96.40	0.19
	R2L	76.76	66.69	68.89	82.73	83.09	0.23
	U2R	79.83	72.36	66.81	99.31	50.00	0.01
Proposed PPDT-SMOTE	DoS	97.12	98.46	97.78	98.06	99.31	0.13
	Probe	98.46	98.46	98.46	98.46	99.17	0.11
	R2L	95.05	94.54	94.79	96.34	97.83	0.20
	U2R	99.31	99.28	99.31	99.33	71.80	0.01

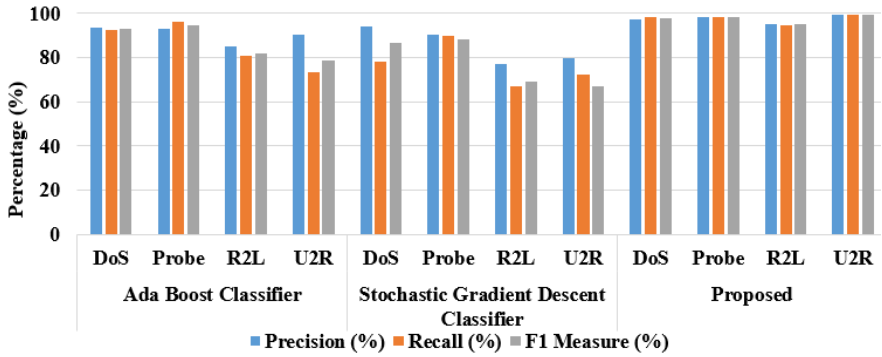

Figure. 4 Graphical representations for the proposed PPDT-SMOTE for multi classes in terms of Precision, Recall, and F-measure

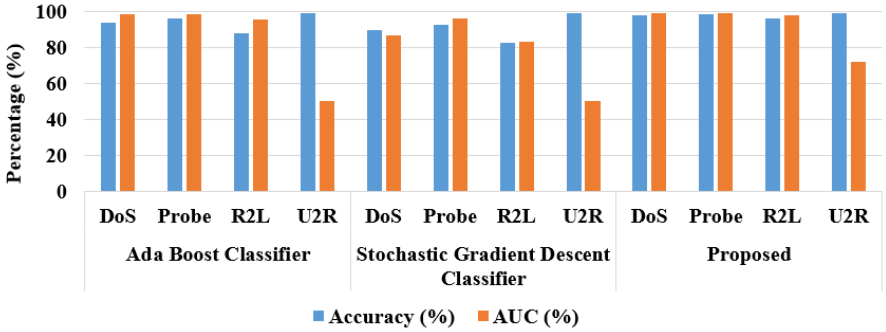

Figure. 5 Graphical representations for the proposed PPDT-SMOTE for multi classes in terms of AUC and Accuracy

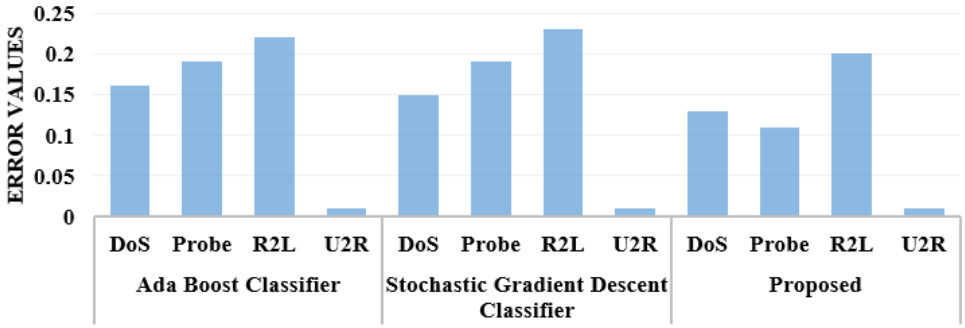

Figure. 6 Graphical representations for the proposed PPDT-SMOTE for multi classes in terms of FAR

descent classifier loses the ability to perform vectorized operations as it deals better only with respect to single attack at a time. The Proposed PPDT-SMOTE showed better classification results for different dataset that classified the attack for both binary as well as multiclass. The graphical representation for table 3 is shown in Fig. 4. The proposed method obtained better results in terms of Precision, Recall and F-measure when compared to the existing methods.

Table 3 shows the Accuracy and AUC values obtained for the proposed PPDT-SMOTE model. The accuracy of 98.06 %, 98.46 %, 96.34 %, 99.33 % was obtained for DoS, Probe, R2L and U2R respectively. The AUC of 99.31%, 99.17%, 97.83% and 71.80% for DoS, Probe, R2L and U2R respectively. Fig. 5 shows the graphical representation for the proposed PPDT-SMOTE for multi classes in terms of AUC and Accuracy. Table 3 shows the results evaluated for DoS attacks in terms of FAR. The FAR values were obtained for the 4 different types of attack classes such as 0.13, 0.11, 0.20, and 0.01 for DoS, Probe, R2L, and U2R respectively. The graphical representation for the FAR is shown in table 3 and Fig. 6 shows the graphical representation in terms of FAR.

4.2 Comparative analysis

The comparative analysis for the proposed PPDT-SMOTE with the existing method RPL in IIoT and shallow model is presented in table 4. In the existing [12] RPL model the detection of attacks with similar scenarios was easier whereas detection of attacks for distinct scenarios was difficult and obtained accuracy of 91.5% which required improvement further. Yanmiao Li [16] developed multi-CNN fusion considered number of samples was used for different attack categories for different training set varies greatly and therefore, for multiclass classification lowered the accuracy of 64.81% performance. Similarly, the existing Shallow model [17] obtained moderate accuracy of 95.22 %

Table 4. Comparative analysis for the proposed and the existing methods

Authors	Methodology	Accuracy (%)
Kashif Naseer Qureshi [12]	RPL in IIoT	91.5
Yanmiao Li [16]	Multi-CNN fusion	64.81
Diro and N. Chilamkurti [17]	Shallow model	95.22
Proposed	PPDT-SMOTE	98.04

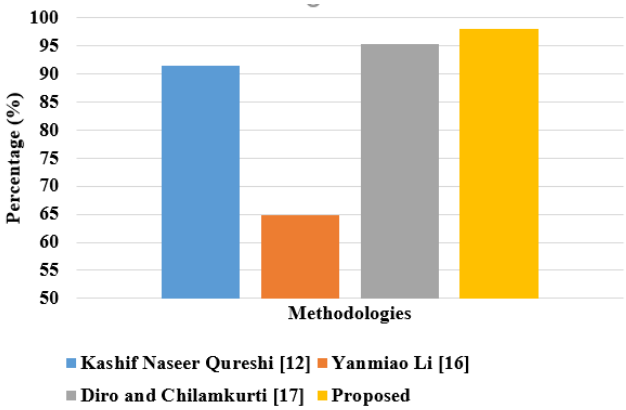

Figure. 7 Graphical representations for the comparative analysis for the proposed and the existing methods

due to training more size of parameters at the learning phase of classification. An advantage of the proposed PPDT-SMOTE reduces the training data and improves the system performance thereby achieves better classification accuracy 98.04 % when compared to existing models. The comparison graph for the proposed model and existing methods are plotted in terms of accuracy and shown in Fig. 7.

5. Conclusion

The intruders or attackers will steal the sensitive information present in the network through the internet. The prevention of attacks by intruders through the internet at an early stage will keep the data safe. The DoS attacks in IoT environments are occurred because of the lack of strong security monitoring and protection techniques. In the present research, NSL-KDD data set was used that has huge number of redundant records that causes the learning algorithms to be biased towards the frequent records. Therefore, the dataset prevents them from learning infrequent records which are harmful to networks. The present research uses NSL-KDD dataset to compare machine learning algorithms such as existing shallow models with the proposed PPDT-SMOTE that performs binary classification (normal or attack) and multi classification identifies the typical attacks such as DoS, Probe, R2L, and U2R. The SMOTE solve the data imbalance problem and uses the data for the detection of vulnerable attacks and PPDT improves the learning rate by eliminating the non-predictive parameters. In this research, a PPDT- SMOTE is used to detect the attacks and classifies as DoS, Probe, R2L and U2R, which gives a 3% better accuracy value when compared to the existing shallow models of 95.22%, RPL of 91.5 % and Multi-CNN fusion model of 64.81% in the IoT environment. In the future, optimization problems of

the proposed PPDT-SMOTE model can be reduced to improve the overall performance of the model.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, have been done by 1st author. The supervision and project administration, have been done by 2nd author.

References

[1] M. Elnour, N. Meskin, K. Khan, and R. Jain, "A Dual-Isolation-Forests-Based Attack Detection Framework for Industrial Control Systems", *IEEE Access*, Vol. 8, pp. 36639-36651, 2020.

[2] N. F. Syed, Z. Baig, A. Ibrahim, and C. Valli, "Denial of service attack detection through machine learning for the IoT", *Journal of Information and Telecommunication*, pp. 1-22, 2020.

[3] K. N. Qureshi, A. Iftikhar, S. N. Bhatti, F. Piccialli, F. Giampaolo, and G. Jeon, "Trust management and evaluation for edge intelligence in the Internet of Things", *Engineering Applications of Artificial Intelligence*, Vol. 94, pp. 103756, 2020.

[4] A. Tewari and B. B. Gupta, "Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework", *Future Generation Computer Systems*, Vol. 108, pp. 909-920, 2020.

[5] K. Mabodi, M. Yusefi, S. Zandiyan, L. Irankhah, and R. Fotohi, "Multi-level trust-based intelligence schema for securing of internet of things (IoT) against security threats using cryptographic authentication", *The Journal of Supercomputing*, pp. 1-26, 2020.

[6] S. S. Seshadri, D. Rodriguez, M. Subedi, K. K. R. Choo, S. Ahmed, Q. Chen, and J. Lee, "Iotcop: A blockchain-based monitoring framework for detection and isolation of malicious devices in internet-of-things systems", *IEEE Internet of Things Journal*, 2020.

[7] A. Samy, H. Yu, and H. Zhang, "Fog-Based Attack Detection Framework for Internet of Things Using Deep Learning", *IEEE Access*, Vol. 8, pp. 74571-74585, 2020.

[8] G. D. L. T. Parra, P. Rad, K. K. R. Choo, and N. Beebe, "Detecting Internet of Things attacks using distributed deep learning", *Journal of Network and Computer Applications*, pp. 102662, 2020.

[9] G. Thamilarasu and S. Chawla, "Towards deep-learning-driven intrusion detection for the internet of things", *Sensors*, Vol. 19, No. 9, pp. 1977, 2019.

[10] M. Wazid, P. R. Dsouza, A. K. Das, K.V. Bhat, N. Kumar, and J. J. Rodrigues, "RAD-EI: A routing attack detection scheme for edge-based Internet of Things environment", *International Journal of Communication Systems*, Vol. 32, No. 15, pp. e4024, 2019.

[11] J. Roldán, J. Boubeta-Puig, J. L. Martínez, and G. Ortiz, "Integrating complex event processing and machine learning: An intelligent architecture for detecting IoT security attacks", *Expert Systems with Applications*, Vol. 149, pp. 113251, 2020.

[12] K. N. Qureshi, S. S. Rana, A. Ahmed, and G. Jeon, "A novel and secure attacks detection framework for smart cities industrial internet of things", *Sustainable Cities and Society*, Vol. 61, pp. 102343, 2020.

[13] M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, "Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city", *Future Generation Computer Systems*, Vol. 107, pp. 433-442, 2020.

[14] A. Mehmood, M. Mukherjee, S. H. Ahmed, H. Song, and K. M. Malik, "NBC-MAIDS: Naïve Bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks", *The Journal of Supercomputing*, Vol. 74, No. 10, pp. 5156-5170, 2018.

[15] G. Liu, W. Quan, N. Cheng, H. Zhang, and S. Yu, "Efficient DDoS attacks mitigation for stateful forwarding in Internet of Things", *Journal of Network and Computer Applications*, Vol. 130, pp. 1-13, 2019.

[16] Y. Li, Y. Xu, Z. Liu, H. Hou, Y. Zheng, Y. Xin, Y. Zhao, and L. Cui, "Robust detection for network intrusion of industrial IoT based on multi-CNN fusion", *Measurement*, Vol. 154, pp. 107450, 2020.

[17] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things", *Future Generation Computer Systems*, Vol. 82, pp. 761-768, 2018.