Journal of Information Systems Engineering and Management
2025, 10(13s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Distributed Intrusion Detection System using Ensemble Learning Technique for Securing IoT Environments

[1]Mr.B.Karthikeyan, 2 Dr.K.Kamali

*[1]*Assistant Professor, Dept. of English, Annamalai university, Annamalainagar, Tamilnadu*

*Corresponding Author: * karthikeyanphd@yahoo.com*

[2]*Assistant Professor. , Dept. of CSE, Annamalai university, Annamalainagar, Tamilnadu kamaliaucse2006@gmail.com*

ARTICLE INFO

Received: 29 Nov 2024
Revised: 14 Jan 2025
Accepted: 28 Jan 2025

ABSTRACT

Introduction: Conventional anomaly detection systems fall short in Internet of Things (IoT) as the range of possible normal behaviors for devices is much wider and more dynamic compared to traditional environments. Main issues of attack detection in IoT include dataset management and optimal feature selection. But existing Intrusion Detection Systems (IDSs) did not concentrate on this issue.

Objectives: The main objective is to classify the data as normal or anomaly using Machine Learning (ML) based algorithms and to test the dataset against an ensemble model involving number of ML classifiers.

Methods: In this paper, a Distributed Intrusion Detection System (DIDS) for IoT environments, using Ensemble learning technique is designed. Principal Component Analysis (PCA) technique is applied for reducing the dimension of data. For classifying the data as normal or anomaly, the classifiers Decision Tree, Gaussian Naive Bayes and XGBoost are applied. Ensemble Learning technique based on maximum voting is used, which combines the multiple learners in order to improve accuracy.

Results: By experimental results, it has been shown that the proposed maximum voting ensemble technique has higher attack detection accuracy with low false positives.

Conclusion: This DIDS can detect and classify nearly 32 attacks..

Keywords: Internet of Things (IoT), Distributed Intrusion Detection System (DIDS), Principal Component Analysis (PCA), Machine Learning (ML), Ensemble Learning.

INTRODUCTION

In today's rapidly developing digital environment, every device is connected with the real-world using IoT [1]. In addition, IoT systems usually store and maintain data in a distributed manner [2]. The prevalent deployment of smart devices has considerably increased possible security risks. Hackers are mainly drawn to extensively used technologies since a vulnerability in a popular device provides greater opportunities for exploitation and, therefore, greater rewards [3]. Moreover, smart devices are infrequently updated, even when manufacturers offer patches for known susceptibilities. Given that these interrelated devices directly influence users' lives, there is an urgent requirement for a robust security infrastructure and a well-defined security threat classification in IoT networks [4].

Efficient IDSs are required for protecting IoT smart devices while lessening resource consumption. A DIDS for IoT environment is mainly efficient since it can detect abnormal behavior in a component by using the cooperation between different IoT devices. Owing to the unique characteristics of IoT networks, implementing DIDS in IoT environments presents several challenges [5][6].

Machine Learning (ML) methods have been progressively proposed to recognize and alleviate security threats. However, conventional approaches did not have optimal feature selection and dataset management, which can affect the detection accuracy. Managing inappropriate features may cause overfitting, making decisions depending on training time [7]. In contrast, Deep Learning (DL) approaches have been used in DIDSs with improved results, especially in maintaining large data sets [8].

The main Objectives of this work are: (i) To classify the data as normal or anomaly,using ML based algorithms (ii) To test the dataset against an ensemble model involving number of ML classifiers.

RELATED WORKS

Javed et al. [6] proposed a novel ML-based two-layered IDS designed for IoT smart home devices, which improves both accuracy and computational efficiency. The initial layer is applied on a smart thermostat to upload data to a cloud server, whereas the next layer works on the cloud, categorizing possible threats.

Gassais et al. [8] proposed a host-based automated IDS framework that incorporates user space and kernel space information with ML techniques. Various ML algorithms, including DL, were implemented for improving detection accuracy while reducing the impact on device performance. Bakhsh et al. [9] proposed an adaptive IDS and Prevention System (IDPIoT) to reinforce security when the number of connected devices grows. IDPIoT improves security by incorporating both host-based into network-based functionalities. Qaddoura et al. [10] proposed a novel three-stage approach for IDS, which includes oversampling, clustering, data reduction and classification using a Single Hidden Layer Feed-Forward Neural Network (SLFN). It generates balanced and effective training data along with its hybrid utilization of unsupervised and supervised methods for detecting intrusions.

PROPOSED METHODOLOGY

Overview

In this work, we propose to design a DIDS for IoT home automation, using ML models and ensemble learning techniques. The KDD cup dataset is used in the training process. During pre-processing phase, PCA technique is applied for dimensionality reduction of data. For classifying the data as normal or anomaly, Decision Tree, Gaussian Naive Bayes and XGBoost classifiers are applied. In order to improve accuracy, an ensemble learning method based on maximum voting is used which combines multiple learners to solve the specific problems. In this work, the KDD cup dataset [11] is used for training. It contains the following 42 features with 494021 records.

Preprocessing using PCA

For dimensionality reduction, PCA technique [12] is applied which is an unsupervised non-linear technique. PCA is utilized to decrease the dimensionality of datasets with many interrelated variables while maintaining as much of the original variance as possible. This is attained by generating principal components (PCs), which are organized so that the first few PCs capture the majority of the variance from all the original variables. The objective of PCA is to create a d x k transformation matrix P, which maps the dataset vector into a new subspace with less dimensions compared to the original one.

Classification using Ensemble Learning

For classifying the data as normal or anomaly, the following ML classifiers are applied: Decision Tree, Gaussian Naive Bayes and XGBoost

Decision Tree (DT)

Decision trees (DTs) are an extensively used machine learning method for regression and classification tasks. The IoT IDS solution utilizes a DT algorithm to detect and manage specific threats. The process is illustrated as follows: The DT classifier is trained utilizing a labeled dataset that includes data of both normal and attack behaviour. The model learns to distinguish between instances depending on selected features. After training, the DT can classify new patterns as either malicious or normal based on their feature weights. When an instance is classified as malicious, the system will implement suitable security measures, such as issuing an alert and blocking network access.

The DT algorithm splits the dataset into subsets based on feature values. For classification, the entropy (information gain) measure is used. The process is recursive, meaning that the tree is built by splitting nodes into branches based on the selected criteria until a stopping condition is met. Common conditions to stop splitting include: (i) A maximum tree depth is reached (ii) A minimum number of samples is reached in a node. (iii) The information gain from further splitting is below a threshold. After the initial tree is built, it may be too complex and overfit the training data. Pruning is defined as, removing branches of the tree which participate less in predicting the target variables to improve generalization.

Gaussian Naive Bayes (GNB)

GNB is based on the Naive Bayes theorem, where continuous features are considered following a Gaussian distribution. It predicts membership probabilities for every class, representing the likelihood that data points belong to a particular class. The class with the highest probability is selected, a concept also considered as Maximum A Posteriori (MAP). To classify each new data point, GNB determines the maximum value of the posterior probability of each class and assigns that point to that class.

XGBoost

XGBoost is a popular and powerful ML algorithm which constructs an ensemble of decision trees through a process called boosting, where each new tree corrects errors made by the previous trees. eXtreme Gradient Boosting (XGBoost) is a boosting technique that belongs to the ensemble-based method. It includes constructing a series of decision trees, which is called as a sequential ensemble method. This method produces results with low bias and high variance, since the model has a strong capability to fit the training information.

Below is a description of XGBoost.

XGBoost Algorithm

1. Initialization

- Initialize the model with the average of the target values.

2. Iterative Process (Boosting)

- Compute Residuals**:** For each instance, compute the residuals or errors from the current model's predictions.
- Fit a New Tree**:** Train a new decision tree to predict the residuals. This tree will correct the errors of the current model.
- Update Model**:** Add the predictions of the new tree to the current model's predictions.
- Apply Regularization**:** Apply regularization to the new tree to control complexity and prevent overfitting.

3. **Repeat:** Iterate the process of computing residuals, fitting new trees, and updating the model for a specified number of boosting rounds or until convergence.

4. **Prediction:** The final model is the ensemble of all the trees, and predictions are made by aggregating the outputs of these trees.

Testing with Ensemble Learning

Ensemble Learning is a method that integrates multiple models to address specific problems and improve the accuracy of classifiers. In this method, the maximum voting method is utilized, where prediction of each model is treated as a vote. The final prediction is depending on the majority of these votes. When there are y voters and every voter has a probability p of making the correct prediction, then the overall probability of the voter to make correct prediction is:

$$P_y = \sum_{\square=\lfloor \frac{\square}{2} \rfloor}^{\square} \left[\frac{\square!}{(\square-\square)!.\square!}\right] . \square^{\square} . (1-\square)^{\square-\square} \quad (1)$$

Thus,

If $p>0.5$, then $p_y>p$. This shows that the majority voter has a greater probability of making the correct prediction than any single voter.

The maximum voting algorithm for ensemble learning, is presented below:

Maximum Voting Algorithm

1. **Train Multiple Models:**

- Train a set of different models (base learners) on the same training dataset. These models can be of the same type (e.g., different decision trees) or different types (e.g., decision trees, support vector machines, logistic regression).

2. **Make Predictions:**

- For each instance in the test dataset, have each base model make a prediction.

3. **Aggregate Votes:**

- For each instance, collect the predictions from all base models.
- Count the number of votes for each class label.

4. **Determine Final Prediction:**

- The class label with the maximum votes is returned as the final prediction for that instance.

EXPERIMENTAL RESULTS

The proposed optimized feature selection technique for DIDS has been implemented in Python 3.0 with Google Colab environment. The KDD cup dataset contains the following 42 features with 494021 records.

Classification Results

The results of the classification algorithms GNB, DT and XGBoost are compared with the results of maximum voting ensemble classifier.

Accuracy is defined by the following formula

$$\textbf{Accuracy} = \frac{TN + TP}{FP + FN + TP + TN} \quad (2)$$

Table 1 presents the performance comparison of the 4 classifiers in terms of accuracy and false positives.

Classifier	Accuracy (%)	False positives
GNB	96.15	453
DT	99.7	56
XGBoost	99.85	32
Max.Voting	99.92	14

Table 1 Performance Comparison

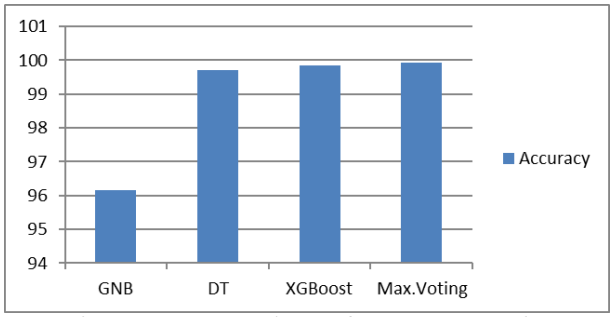

Figure 1 Comparison of Accuracy results

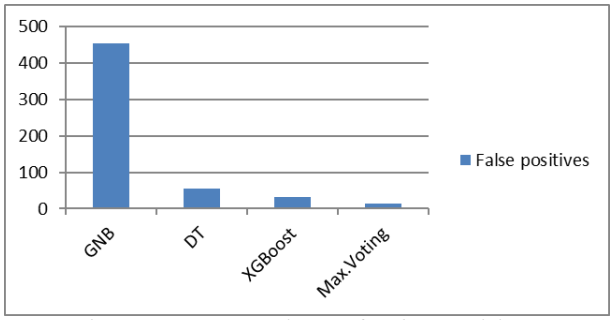

Figure 2 Comparison of False positives

Figure 1 and 2 show that the maximum voting ensemble classifier with proper preprocessing activities, gives improved and efficient performance with high accuracy and less false positives. Moreover, since XGBoost yields almost equally efficient results similar to the ensemble classifier, it can be considered as the most efficient classifier among the others.

CONCLUSION

This work proposes a DIDS for IoT environments, using ML models. During preprocessing phase, PCA technique is applied for dimensionality reduction of data. For classifying the data as normal or anomaly, the Decision Tree, Gaussian Naive Bayes and XGBoost classifiers are applied. In order to improve accuracy, Ensemble Learning using maximum voting algorithm is used. The KDD cup dataset is used for training. It contains the following 42 features with 494021 records. The results of the classification algorithms GNB, DT and XGBoost are compared with the results of maximum voting ensemble classifier. By experimental results, it has been shown that the proposed maximum voting ensemble technique has higher attack detection accuracy with low false positives.

REFERENCES

[1] Madhu, B., Chari, M. V. G., Vankdothu, R., Silivery, A. K., & Aerranagula, V. (2023). Intrusion detection models for IOT networks via deep learning approaches. *Measurement Sensors*, *25*, 100641. https://doi.org/10.1016/j.measen.2022.100641

[2] Facchini, S., Giorgi, G., Saracino, A., & Dini, G. (2020). Multi-level Distributed Intrusion Detection System for an IoT based Smart Home Environment. *6th International Conference on Information Systems Security and Privacy*. https://doi.org/10.5220/0009170807050712

[3] Adeel Abbas, Muazzam A. Khan, Shahid Latif, Maria Ajaz, Awais Aziz Shah, Jawad and Ahmad,"A New Ensemble-Based Intrusion Detection System for Internet ofThings",Arabian Journal for Science and Engineering (2022) 47:1805–1819

[4] Anthi, E., Williams, L., Slowinska, M., Theodorakopoulos, G., & Burnap, P. (2019). A supervised intrusion detection system for smart home IoT devices. *IEEE Internet of Things Journal*, *6*(5), 9042–9053. https://doi.org/10.1109/jiot.2019.2926365

[5] Poongodi, M., & Hamdi, M. (2023). Intrusion detection system using distributed multilevel discriminator in GAN for IoT system. *Transactions on Emerging Telecommunications Technologies*, *34*(11). https://doi.org/10.1002/ett.4815

[6] Javed, A.; Ehtsham, A.; Jawad, M.; Awais, M.N.; Qureshi, A.-u.-H.; Larijani, H. Implementation of Lightweight Machine Learning-Based Intrusion Detection System on IoT Devices of Smart Homes. Future Internet 2024, 16, 200. https://doi.org/10.3390/fi16060200

[7] Otoum, Y., Liu, D., & Nayak, A. (2019). DL-IDS: a deep learning–based intrusion detection framework for securing IoT. *Transactions on Emerging Telecommunications Technologies*, *33*(3). https://doi.org/10.1002/ett.3803

[8] Gassais, R., Ezzati-Jivan, N., Fernandez, J. M., Aloise, D., & Dagenais, M. R. (2020). Multi-level host-based intrusion detection system for Internet of things. *Journal of Cloud Computing Advances Systems and Applications*, *9*(1). https://doi.org/10.1186/s13677-020-00206-6

[9] Bakhsh, S. T., Alghamdi, S., Alsemmeari, R. A., & Hassan, S. R. (2019). An adaptive intrusion detection and prevention system for Internet of Things. *International Journal of Distributed Sensor Networks*, *15*(11), 155014771988810. https://doi.org/10.1177/1550147719888109

[10] Qaddoura, R.; Al-Zoubi, A.M.; Almomani, I.; Faris, H. A Multi-Stage Classification Approach for IoT Intrusion Detection Based on Clustering with Oversampling. Appl. Sci. 2021, 11, 3022. https://doi.org/10.3390/app11073022

[11] http://kdd.ics.uci.edu/databases/kddcup99/kddcup.data_10_percent.gz

[12] Moses Njue and Billy Franklin,"Dimensionality Reduction on MNIST dataset using PCA, T-SNE and UMAP",2020.