Utilizing the Ensemble Learning and XAI for Performance Improvements in IoT Network Attack Detection

Chathuranga Sampath Kalutharage[1], Xiaodong Liu[1][0000−0002−7612−9981], Christos Chrysoulas[1][0000−0001−9817−003X], and Oluwaseun Bamgboye[1]

Edinburgh Napier University, Scotland, UK
c.kalutharage,x.liu,c.chrysoulas,O.Bamgboye@napier.ac.uk
https://www.napier.ac.uk/

Abstract. As Internet of Things (IoT) networks continue to expand, it has become increasingly crucial to safeguard the security of these interconnected devices. This research study proposes a novel method for enhancing the effectiveness of IoT network threat detection by employing ensemble learning techniques and Explainable Artificial Intelligence (XAI).

The proposed method involves the utilization of an ensemble model combining an Autoencoder and eXtreme Gradient Boosting (XGBoost), a popular gradient-boosting algorithm. The workflow begins with quantizing the dataset to reduce computational complexity. Subsequently, the autoencoder is trained to learn a compressed representation of the quantized data, while XGBoost simultaneously performs classification tasks. To enhance the efficiency and accuracy of attack detection, feature importance analysis is conducted using XGBoost's feature_importances_ attribute. This analysis enables the identification of the most influential features, which are then used to prepare a refined dataset, further reducing computational requirements. A Logarithmic layer is introduced within the autoencoder, enabling the linearization of relationships and handling of exponential characteristics.

The ensemble model, combining the Logarithmic Autoencoder's and XGBoost's strengths, is trained on the refined dataset. This unified model significantly enhances attack detection performance by leveraging the compressed representations learned by the autoencoder and the predictive power of XGBoost. Our proposed model is evaluated in the experiment on the CICIDS2017 data set. The evaluation metrics include accuracy, recall, precision, and runtime. For detection performance, our proposed model achieves an impressive 99.92% detection accuracy on the CICIDS2017 dataset, surpassing most state-of-the-art intrusion detection methods. Moreover, our proposed model exhibits the lowest runtime, further highlighting its efficiency.

Keywords: Attack detection · Ensemble learning · Explainable AI · XGBoost · Autoencoder.

1 Introduction

IoT devices run a significant risk of being the target of cyberattacks such as impersonation, interception, and unauthorised access by both people and viruses because of their connectedness via the Internet [20]. It is crucial to create a strong security system for these devices as a result. Due to the complexity and complexity of contemporary threats, traditional signature-based intrusion detection systems (IDS) have been inefficient in identifying these attacks [21]. As a result, the majority of current IoT security research focuses on using Artificial Intelligence (AI) techniques such as SVM [7], decision trees [17], neural networks [4], Autoencoder [15] and so on. AI systems offer sophisticated solutions that are better able to recognise and mitigate the effects of cyberattacks and possible threats to IoT data and infrastructures. AI systems are distinguished by their iterative and dynamic nature. IoT security measures may adapt to the changing threat landscape by utilising ML, providing improved defence and security capabilities [12].

IoT devices are characterized by limited resources, posing challenges when implementing traditional computing security solutions in IoT networks. The application of AI-based security mechanisms on IoT devices faces memory capacity limitations, necessitating the design of lightweight models [6]. However, many security datasets are complex and demand significant computational power. Data normalization serves as a necessary preprocessing step in numerous AI systems prior to feature engineering and classifier training. While this step can potentially improve model performance, it also carries the risk of diminishing data accuracy and detection accuracy, particularly in intrusion detection datasets. Furthermore, data normalization can introduce challenges when attempting to modify the dataset, as adding or removing data may necessitate a complete reconsideration of the training procedure, potentially resulting in increased runtime. In this study, we employ eXplainable Artificial Intelligence (XAI) technology to reduce the number of features by identifying the most influential ones for attack identification. This feature reduction process aims to refine the dataset, enhance performance, and achieve low computational power requirements while maintaining higher data accuracy and detection accuracy.

Machine learning-based detection techniques have yielded intriguing findings in the field of security. Among the available methodologies, binary detection techniques fall short in providing comprehensive security as they merely identify the presence of intrusions. Consequently, it is crucial to recognize the specific attack category to implement appropriate defences against each unique threat. This identification proves valuable for decision-making by network administrators, who can then take steps to address the vulnerabilities exploited by the attack. However, it is worth noting that existing multi-class detection techniques, aimed at categorizing attacks, often have lower hit rates compared to binary methods [2, 28], which can make it challenging to detect certain types of attacks [22, 13]. Our work makes a significant contribution to the state of the art in attack detection and identification.

In this paper, Our proposed approach is based on ensemble methods and a dual analytic architecture. Ensemble learning is used because there are machine learning scenarios where even the best model is not accurate enough. Hybrid methods are used to merge models to reduce model instability. State-of-the-art methods had several limitations in terms of identifying specific attacks. The ensemble method for IoT network attack detection combines an autoencoder as an anomaly detection model in the first stage and a multi-class detection method in the second stage. In the first stage, the autoencoder learns normal patterns during training and identifies anomalies by measuring the reconstruction error of input data during testing. Instances with high reconstruction error beyond a threshold are classified as potential attacks. In the second stage, a multi-class classifier, XGBoost utilizes features extracted from network traffic features to determine the specific attack type. The classifier is trained to classify the detected anomalies into different attack categories.

This ensemble approach leverages the strengths of both the autoencoder's anomaly detection and the multi-class classifier's attack type identification, resulting in a more comprehensive and accurate IoT network attack detection system. It enables a nuanced understanding of detected attacks and facilitates the implementation of appropriate defense strategies. Careful model training, feature selection, and threshold setting are crucial for the reliability and performance of the ensemble method. This paper makes several key contributions, which include the following:

- We proposed a new methodology for refining datasets in attack detection by reducing the features to include only the most influential ones. This methodology leverages the principles of XAI (Explainable Artificial Intelligence) to identify and prioritize the features that have the most significant impact on the detection of attacks.
- This paper presents a proposed new ensemble approach for detecting IoT attacks. The approach combines unsupervised learning using an autoencoder with a multiclass classifier, XGBoost, to achieve higher accuracy in attack detection.
- We present a comprehensive evaluation of the proposed method on the CIC-IDS dataset and implement a lightweight model as it need to deploy on IoT devices

The paper is organized as follows: Section 2 provides an overview of the background and related work in the field. Section 3 presents a detailed description of the proposed method. The obtained results using the CIC-IDS benchmark dataset are discussed in Section 4. Finally, Section 5 concludes the paper, summarizing the findings and highlighting potential future research directions.

2 Realated Work

Extensive research has been conducted on the detection of IoT network attacks, as well as traditional network intrusion detection methods. Sumaiya Thaseen

Ikram et al. [14] proposed an ensemble intrusion detection model that combines various neural network architectures, including Long Short-Term Memory (LSTM), Backpropagation Network (BPN), and Multilayer Perceptron (MLP). The proposed model consisted of two core modules: the learning algorithm module and the evaluation module. The learning algorithm module included LSTM, BPN, and MLP, which were utilized to generate training data as input. The outputs of these three models were then sent to XGBoost in the evaluation module for further analysis. Hagos et al. [10] proposed a detection strategy that utilizes Support Vector Machines (SVMs) and the l1-regularized method with Least Absolute Shrinkage and Selection Operator (LASSO). This strategy aims to achieve robust regression for binary and multiclass attack categorization in network intrusion detection. Chunhe Song et al. [26] proposed an intrusion detection system method that combines deep learning and feature-based techniques. The method uses a Bayesian approach to tune XGBoost's hyperparameters for maximum performance while minimizing performance loss due to incorrect parameter selection. Additionally, a genetic algorithm-based crossover technique is proposed to reduce the likelihood of the Bayesian algorithm becoming trapped in local optimization during the optimization phase. The final detection results are merged based on the outputs of both LSTM and XGBoost, which are used as detectors in the system.

Vinayakumar et al. [28] also explored deep networks for intrusion detection. They proposed a Deep Belief Networks (DBN)-based approach. The authors [19] proposed a new intrusion detection method based on Sequential Online Extreme Learning Machine (OS-ELM) for fog computing environments. ELM is a neural network that is known for its fast training speed and good generalizability. OS-ELM is a modification of ELM that can be used to handle online applications. The proposed method uses multiclass detection, but this makes it more difficult to identify attacks that involve privilege escalation and probing. Chaofei Tang et al. [27] proposed a network intrusion detection system based on LightGBM and autoencoders. The LightGBM method was used for feature selection, and the AE was then used as a classifier for training and detection. The authors [32] introduced a novel model for intrusion detection system (IDS) that combines a deep neural network (DNN) with an enhanced conditional variational autoencoder (ICVAE). The ICVAE is employed to automatically learn potential sparse representations between features and classes in the dataset. This helps improve detection accuracy by balancing the training data and generating new attack samples based on the defined intrusion categories. Furthermore, the weights of the hidden layers in the DNN are initialized using the ICVAE encoder, simplifying the forward and backward propagation processes.

Blanco et al. [3] proposed a method for multi-class network attack classification that can be installed in a router. The method is based on a Convolutional Neural Network (CNN), which is a type of neural network that was originally developed for image classification. The method was validated using two open datasets, UNSW-NB15 and NSL-KDD. However, the work did not consider the IoT setting. The authors [25] proposed a lightweight machine learning-based in-

trusion detection system (IDS) that uses a new feature selection technique called Correlated-Set Thresholding on Gain-Ratio (CST-GR) and Decision Tree (DT). The IDS was implemented on a Raspberry Pi. The method was tested on the Bot-IoT dataset, but results for benign traffic classification were not provided. Xukui Li et al. [16] proposed an efficient deep learning technique that combines an autoencoder (AE) and random forest (RF). The model uses RF to select the most useful features from the dataset. The AP clustering algorithm is then used to divide the selected features into subgroups, which provides information for the AE to determine the root mean square error (RMSE). The network traffic is then classified as normal or abnormal using either K-means or the Gaussian mixture model (GMM), depending on the RMSE values. Shafiq et al. [22] proposed a new feature selection method called CorrAUC. The new algorithm uses the wrapper technique to effectively filter features and select useful features for the machine learning technique using the Area Under the Curve (AUC) metric. The proposed methodology was validated using the Bot-IoT dataset and four machine learning algorithms, including Random Forest (RF), which had the best performance. RFs are ensemble algorithms that use a collection of combined decision trees to improve generalization and robustness. Maniriho et al. [18] proposed an anomaly-based intrusion detection system (IDS) for IoT networks. The system uses a hybrid resource selection mechanism to select the most relevant resources and Random Forest to classify each stream of traffic as normal or abnormal. The system was evaluated using the IoTID20 dataset, which is one of the most recent datasets for anomaly detection in the IoT ecosystem. The results showed that the system was effective in detecting anomalies. However, the system has not been evaluated in the multi-class detection scenario.

Many approaches have been proposed for detecting and identifying network intrusions in this area. Some of these methods have used machine learning techniques, but they have not taken into account the specific context of IoT networks. Others have been able to identify certain types of attacks, but they have been difficult to use in practice. None of the existing methods have used XAI, ensemble methods, or been able to achieve the same level of accuracy as the proposed model with low computational power requirements.

3 Methodology

3.1 Overview of Approach

The increasing use of IoT-based systems, their security flaws, and research on attack detection in these settings all point to the importance of being able to identify the specific class of an attack. This is essential for implementing countermeasures and making decisions. However, current multi-class detection methods often have difficulty identifying specific attack types [8]. This paper presents an ensemble approach aimed at identifying and classifying attack types in IoT contexts. The proposed method involves several key steps. First, we employ XAI to refine the dataset and identify the most influential features for attack detection.

Next, we train an XGBoost model exclusively using these selected features, focusing on the top 20 most influential ones. Additionally, an autoencoder model is trained using benign data. The CIC flow meter is then utilized to extract features from PCAP files, which are subsequently preprocessed for inference using both the autoencoder and XGBoost models. Figure 1 illustrates the main steps of the engineering pipeline in our proposed method.

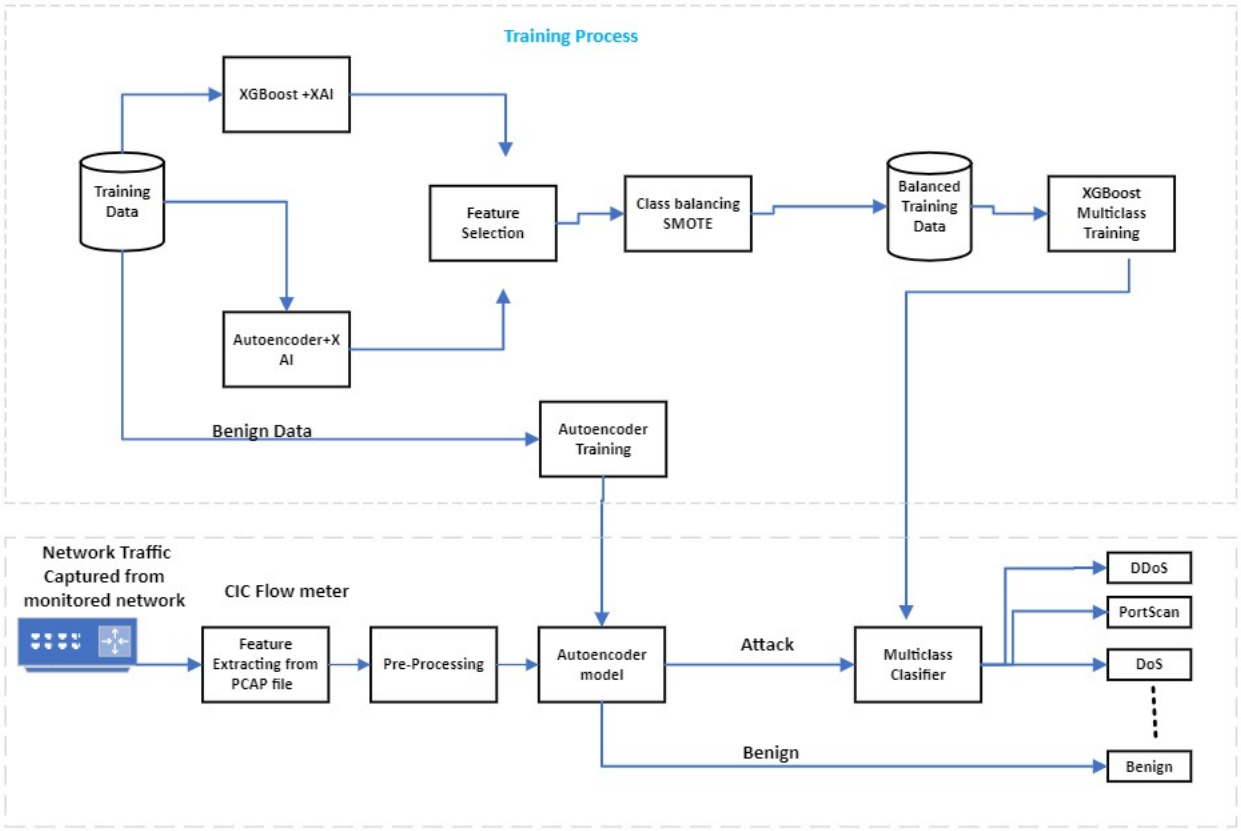

Fig. 1. Engineering pipeline of the proposed method.

3.2 Dataset Refine with XAI

Many security datasets are large and complex, requiring a lot of processing power. Data normalization is a common preprocessing step used by many AI systems before feature engineering and classifier training. While this can improve model performance, it also risks reducing data accuracy and detection accuracy, especially in datasets used for intrusion detection. Additionally, adding

or removing data may require a complete re-evaluation of the training process, significantly increasing runtime. As a result, data normalization can present challenges when trying to adjust the dataset.

In this study, we use XAI technology to select the most important attributes for attack identification, thereby reducing the number of features. XAI is a type of AI that allows us to understand how models make decisions. This is important for intrusion detection, as it helps us to identify the specific features that are most indicative of an attack. By reducing the number of features, we can improve the performance of the model while also making it more interpretable. First, we trained both models and applied XAI techniques to identify the specific features that are most indicative of an attack. The steps for applying XAI are outlined in Algorithm 1 and Algorithm 2. We successfully reduced the dataset to 20 features while maintaining higher accuracy for each model. In addition, we conducted data type casting to determine the most efficient data type for the CIC-IDS dataset. Through our analysis, we found that utilizing a 32-bit data type proved to be the most efficient choice. This casting resulted in a significant reduction in memory usage, up to 48.75%, compared to the usage with the default 64-bit data type.

Algorithm 1 Finding Most Influential Features using SHAP Values with Autoencoder

Require: Autoencoder model, X test data
Ensure: Top features based on SHAP values
function FindTopFeatures($autoencoder, x_{test}$)
    Create SHAP explainer object
    Compute SHAP values for the test data
    Compute mean absolute SHAP values for each feature
    Get indices of top features based on SHAP values
    Get top feature names
    return Top feature names
end function

Algorithm 2 Finding Most Influential Features using SHAP Values

Require: XGBoost model, X test data
Ensure: Top features based on SHAP values
function FindTopFeatures(xgb, x_{test})
    Create SHAP explainer object
    Compute SHAP values for the test data
    Compute mean absolute SHAP values for each feature
    Get indices of top features based on SHAP values
    Get top feature names
    return Top feature names
end function

3.3 Attack Detection

The next step in the proposed approach is anomaly detection. An autoencoder is used to perform this task, which distinguishes between attack and benign traffic. Autoencoders are important tools for anomaly detection. Recent research has focused on anomaly-based attack detection for security systems [24]. Autoencoders are a type of neural network that can learn to reconstruct its input. This means that after training, an autoencoder can take an input and produce an output that is identical to the input. Autoencoders are made up of two parts: an encoder and a decoder. The encoder takes the input and transforms it into a latent representation. The decoder then takes this latent representation and reconstructs the input. We used autoencoder models to identify anomalies. We did this by training an autoencoder to reconstruct input data. We then defined an anomaly score as the difference between the input value and the reconstructed output value. Anomalies were identified as data points with high anomaly scores. Equation 1 shows the reconstruction error calculation in our work. Given an input row (X) with an array of features (xi) and its output row (X') with reconstructed feature values (xi'), and employing an anomaly detection model (F), the sum of the reconstruction errors for each feature that is specific to a certain row produces the reconstruction error for that row. If the reconstruction error exceeds the input value, it is identified as an anomaly. We trained the model on benign data. We used a fully connected autoencoder model with RELU activation. To reduce the network load, we used only 2 hidden layers, with 10 and 32 neurons in each layer, respectively. We used the highest mean squared error (MSE) from benign data as the threshold for anomalous data.

$$F(X, X') = \sum_{i=1}^{n} (x_i - x'_i)^2 \tag{1}$$

3.4 Attack Identification

The subsequent stage in the proposed approach involves identification, which focuses on analyzing traffic that has been previously identified as an anomaly and determining the specific type of attack. This step employs a more robust method by utilizing XGBoost for the identification process. We first explain the gradient boosting decision tree (GBDT) before introducing XGBoost. In a nutshell, the classification and regression tree (CART) results are accumulated by GBDT to reach the conclusion. The fundamental principle is that every tree learns the difference between the true value and the predicted value of all prior CART results. However, each iteration of GBDT necessitates numerous traversals of the full data set. The size of the data can only be as much as what can fit in memory; otherwise, time-consuming read and write operations must be performed repeatedly. Therefore, GBDT is unable to satisfy its needs when presented with huge and highdimensional data. XGBoost was created to address GBDT's problem in handling big samples and high-dimensional data [5]. XGBoost is a powerful distributed gradient boosting library that excels in terms of

performance, flexibility, and portability. It provides extensive support for various machine learning techniques within the gradient boosting framework. Compared to traditional gradient boosting methods like GBDT (Gradient Boosting Decision Tree), XGBoost offers several notable advantages:

- The algorithm introduces a distinct loss function and elevates the objective function from the first order to the second order
- The inclusion of the L2 regularization term on the number of leaf nodes and leaf weights effectively reduces the model's variance and prevents overfitting
- The base learner of the algorithm supports both CART
- To leverage the block structure and accurately locate data separation points, the method stores the presorted dataset in a sparse matrix storage format. This approach reduces the computational workload significantly, enabling more efficient calculations
- The introduction of sparsity-aware algorithms incorporates automatic handling of default values. This feature automatically splits samples with default values by evaluating the gain of default-valued samples in both left and right branches. The algorithm selects the branch with the highest gain, enabling effective division of samples with default values
- To mitigate overfitting and reduce computational complexity, the algorithm incorporates column sampling. This technique draws inspiration from the sampling process used in random forests. By randomly selecting a subset of columns (features) during the training process, the algorithm effectively reduces overfitting and computational requirements
- The shrinkage approach is presented. The weight of each leaf node of the tree is multiplied by the reduction weight in each iteration, which reduces the influence of each tree and increases the opportunity for optimisation of the subsequent trees

4 Results and Evaluation

In this section, we provide an overview of the methodology employed to evaluate the proposed approach. We discuss the key aspects of the evaluation process and present the results obtained from conducting experiments. Evaluating detection methods is crucial to assess their applicability in real-world scenarios. The primary objective is to obtain an estimation of the practical accuracy of the approaches under investigation.

4.1 Dataset and Experimental Environment

This paper conducted comparative experiments on the CICIDS2017 dataset to validate the simulation results in the field of network intrusion detection. The experiments were divided into two sets: an ablation experiment that compared the impact of each module on performance, and a comparison with other existing intrusion detection algorithms to evaluate the algorithm's overall performance.

The simulations were implemented using Python 3.6 in a Windows environment. The experiments were conducted on an ASUS ZenBook equipped with a 2.30 GHz Intel Core i7 processor and 16 GB of RAM.

4.2 Performance Metrics

The evaluation of our proposed model relies on several key metrics: recall, precision, F1-score, and accuracy. These metrics can be calculated using the scikit-learn machine learning module in Python. See the Equation 2, 3, 4, 5. Accuracy represents the percentage of accurately predicted samples out of the total samples. Precision measures the proportion of correctly predicted positive samples among all predicted positive samples, while recall measures the proportion of correctly predicted positive samples among all actual positive samples in the original sample. The F1-score is a weighted harmonic average of precision and recall.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{2}$$

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{3}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

$$Accuracy = \frac{Numberofcorrectlypredictedsamples}{Totalnumberofsamples} \tag{5}$$

The datasets utilized in the studies were divided into training and testing sets, with a stratification ratio of 70% for training and 30% for testing. The metrics mentioned above were computed based on the predictions made by the models on the testing set.

4.3 Experimental Evaluation on CIC-IDS dataset

The suggested method was evaluated using the CSE-CIC-IDS dataset [1], which was developed through a collaboration between the Canadian Institute for Cybersecurity (CIC) and the Communications Security Establishment (CSE). The objective of this project was to create a comprehensive dataset comprising diverse sets of instructional data. These datasets include user profiles with various events paired with network-observed behaviours. The CSE-CIC-IDS dataset serves as a valuable resource for assessing the proposed method. This dataset consists of 13 classes including benign classes which are Benign, Bot , DDoS, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, FTP Patator, Heartbleed, Infiltration, Portscan, SSH-Patator, Web Attack. The proposed approach demonstrated high detection rates for most types of attacks, surpassing or matching the detection rates achieved by current state-of-the-art methods. The results

[1] https://www.unb.ca/cic/datasets/ids-2017.html

obtained by the proposed approach, along with the results from existing state-of-the-art methods, are summarized in Table 1. In this study, we specifically employ recall (REC) and precision (PRE) as evaluation metrics. Recall measures the proportion of true positive instances correctly identified out of the total actual positive instances (true positives and false negatives). It provides insights into the model's ability to capture positive instances and is particularly useful when identifying all positive instances is important. Precision, on the other hand, measures the proportion of true positive instances correctly predicted out of the total predicted positive instances (true positives and false positives). It assesses the model's ability to avoid false positives and is valuable when minimizing false alarms is a priority. By considering both recall and precision, we gain a more comprehensive understanding of the model's performance in correctly identifying positive instances and minimizing false positives. Further, all types of DoS attacks, including DoS GoldenEye, DoS Hulk, and DoS Slowhttptest, exhibit similar detection rates, we refer to them collectively as DoS attacks for the sake of brevity and clarity. With the exception of the BOTNET attack class, the proposed approach achieves higher detection rates for all other attack classes compared to the current state of the art. Notably, the detection rates for Infiltration and Web attacks are particularly higher than those reported in previous works. The effectiveness of the proposed approach in detecting these specific attack classes demonstrates its superiority over existing methods. Furthermore, the overall performance of the proposed model significantly outperforms the current state of the art, as demonstrated in Table 2. The comprehensive evaluation of the model across various metrics highlights its superior performance and effectiveness compared to existing methods.

Table 1. Results of comparison of each attack type with Current State of the art

	DNN [6]		RF [6]		ET+Multi Cla [6]		Proposed model	
	PRE	REC	PRE	REC	PRE	REC	PRE	REC
Benign	98.81	99.91	98.88	99.78	98.96	99.58	99.96	99.80
BOT	99.77	99.88	99.62	99.77	99.99	99.90	86.78	82.89
DDoS	99.47	99.88	99.91	99.91	99.99	99.94	100	100
DoS	95.32	89.61	96.35	89.26	97.38	88.13	100	100
FTP-Patator	-	-	-	-	-	-	100	100
Heartbleed	-	-	-	-	-	-	100	67.97
Infiltration	44.10	01.79	29.60	07.08	28.46	13.50	100	64.87
Portscan	-	-	-	-	-	-	99.77	100
SSH-Patator	-	-	-	-	-	-	100	100
Web Attack	100	39.29	92.31	42.86	39.34	85.71	99.92	99.23

Table 2. Overall model Performance

Detection Method	Accuracy	Time(s)
NB-SVM [9]	98.92	-
T-SNERF [11]	99.78	-
MTH-IDS [31]	99.89	478.2
LMDRT-SVM2 [29]	99.28	-
DT+rule-based model [1]	96.66	160.07
KNN [23]	96.30	152463.6
RF [23]	98.82	1848.3
LogAE-XGBoost [30]	99.92	1092.35
ET+MultiClass Clasi [6]	98.21	-
Proposed Model	99.92	826.30

5 Conclusion

This paper introduces a novel IoT network attack detection system that leverages XAI and ensemble learning techniques. The proposed model incorporates XAI to identify the most influential features for attack detection and to reduce the feature space for a lightweight model. An autoencoder is employed for anomaly detection in the first stage, allowing agile release of benign traffic and enabling more robust inspection using the XGBoost approach for unidentified events. Additionally, the identification of attack types provides valuable information for mitigation mechanisms and network management. The proposed approach demonstrates comparable or superior performance, highlighting its robustness. The experiments conducted using the CIC-IDS dataset serve as a starting point, and future work will focus on evaluating the proposed approach using other benchmark datasets. Furthermore, the intention is to simulate the system on real-world IoT networks to assess its effectiveness in real-time detection.

References

1. Ahmim, A., Maglaras, L., Ferrag, M.A., Derdour, M., Janicke, H.: A novel hierarchical intrusion detection system based on decision tree and rules-based models. In: 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS). pp. 228–233. IEEE (2019)
2. Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S., Razaque, A.: Deep recurrent neural network for iot intrusion detection system. Simulation Modelling Practice and Theory **101**, 102031 (2020). https://doi.org/https://doi.org/10.1016/j.simpat.2019.102031, https://www.sciencedirect.com/science/article/pii/S1569190X19301625, modeling and Simulation of Fog Computing
3. Blanco, R., Malagón, P., Cilla, J.J., Moya, J.M.: Multiclass network attack classifier using cnn tuned with genetic algorithms. In: 2018 28th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS). pp. 177–182 (2018). https://doi.org/10.1109/PATMOS.2018.8463997

4. Canêdo, D.R.C., Romariz, A.R.S.R.: Intrusion detection system in ad hoc networks with artificial neural networks and algorithm k-means. IEEE Latin America Transactions **17**(07), 1109–1115 (2019)
5. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794 (2016)
6. de Souza, C.A., Westphall, C.B., Machado, R.B.: Two-step ensemble approach for intrusion detection and identification in iot and fog computing environments. Computers Electrical Engineering **98**, 107694 (2022). https://doi.org/https://doi.org/10.1016/j.compeleceng.2022.107694, https://www.sciencedirect.com/science/article/pii/S0045790622000155
7. Deng, H., Zeng, Q.A., Agrawal, D.P.: Svm-based intrusion detection system for wireless ad hoc networks. In: 2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No. 03CH37484). vol. 3, pp. 2147–2151. IEEE (2003)
8. Diro, A.A., Chilamkurti, N.: Distributed attack detection scheme using deep learning approach for internet of things. Future Generation Computer Systems **82**, 761–768 (2018). https://doi.org/https://doi.org/10.1016/j.future.2017.08.043, https://www.sciencedirect.com/science/article/pii/S0167739X17308488
9. Gu, J., Lu, S.: An effective intrusion detection approach using svm with naïve bayes feature embedding. Computers & Security **103**, 102158 (2021)
10. Hagos, D.H., Yazidi, A., Kure, , Engelstad, P.E.: Enhancing security attacks analysis using regularized machine learning techniques. In: 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA). pp. 909–918 (2017). https://doi.org/10.1109/AINA.2017.19
11. Hammad, M., Hewahi, N., Elmedany, W.: T-snerf: A novel high accuracy machine learning approach for intrusion detection systems. IET Information Security **15**(2), 178–190 (2021)
12. Hussain, F., Hussain, R., Hassan, S.A., Hossain, E.: Machine learning in iot security: Current solutions and future challenges. IEEE Communications Surveys Tutorials **22**(3), 1686–1721 (2020)
13. Ieracitano, C., Adeel, A., Morabito, F.C., Hussain, A.: A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. Neurocomputing **387**, 51–62 (2020). https://doi.org/https://doi.org/10.1016/j.neucom.2019.11.016, https://www.sciencedirect.com/science/article/pii/S0925231219315759
14. Ikram, S.T., Cherukuri, A.K., Poorva, B., Ushasree, P.S., Zhang, Y., Liu, X., Li, G.: Anomaly detection using xgboost ensemble of deep neural network models. Cybernetics and information technologies **21**(3), 175–188 (2021)
15. Kalutharage, C.S., Liu, X., Chrysoulas, C., Pitropakis, N., Papadopoulos, P.: Explainable ai-based ddos attack identification method for iot networks. Computers **12**(2), 32 (2023)
16. Li, X., Chen, W., Zhang, Q., Wu, L.: Building auto-encoder intrusion detection system based on random forest feature selection. Computers & Security **95**, 101851 (2020)
17. Luna, J.M., Gennatas, E.D., Ungar, L.H., Eaton, E., Diffenderfer, E.S., Jensen, S.T., Simone, C.B., Friedman, J.H., Solberg, T.D., Valdes, G.: Building more accurate decision trees with the additive tree. Proceedings of the national academy of sciences **116**(40), 19887–19893 (2019)
18. Maniriho, P., Niyigaba, E., Bizimana, Z., Twiringiyimana, V., Mahoro, L.J., Ahmad, T.: Anomaly-based intrusion detection approach for iot networks using machine learning. In: 2020 International Conference on Computer Engi-

neering, Network, and Intelligent Multimedia (CENIM). pp. 303–308 (2020). https://doi.org/10.1109/CENIM51130.2020.9297958
19. Prabavathy, S., Sundarakantham, K., Shalinie, S.M.: Design of cognitive fog computing for intrusion detection in internet of things. Journal of Communications and Networks **20**(3), 291–298 (2018). https://doi.org/10.1109/JCN.2018.000041
20. Samaila, M.G., Neto, M., Fernandes, D.A., Freire, M.M., Inácio, P.R.: Challenges of securing internet of things devices: A survey. Security and Privacy **1**(2), e20 (2018)
21. Sampath Kalutharage, C., Liu, X., Chrysoulas, C.: Explainable ai and deep autoencoders based security framework for iot network attack certainty (2022)
22. Shafiq, M., Tian, Z., Bashir, A.K., Du, X., Guizani, M.: Corrauc: A malicious bot-iot traffic detection method in iot network using machine-learning techniques. IEEE Internet of Things Journal **8**(5), 3242–3254 (2021). https://doi.org/10.1109/JIOT.2020.3002255
23. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp **1**, 108–116 (2018)
24. Singh, J., Nene, M.J.: A survey on machine learning techniques for intrusion detection systems. International Journal of Advanced Research in Computer and Communication Engineering **2**(11), 4349–4355 (2013)
25. Soe, Y.N., Feng, Y., Santosa, P.I., Hartanto, R., Sakurai, K.: Towards a lightweight detection system for cyber attacks in the iot environment using corresponding features. Electronics **9**(1) (2020). https://doi.org/10.3390/electronics9010144, https://www.mdpi.com/2079-9292/9/1/144
26. Song, C., Sun, Y., Han, G., Rodrigues, J.J.: Intrusion detection based on hybrid classifiers for smart grid. Computers & Electrical Engineering **93**, 107212 (2021)
27. Tang, C., Luktarhan, N., Zhao, Y.: An efficient intrusion detection method based on lightgbm and autoencoder. Symmetry **12**(9), 1458 (2020)
28. Vinayakumar, R., Soman, K.P., Poornachandran, P.: Evaluating effectiveness of shallow and deep networks to intrusion detection system. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). pp. 1282–1289 (2017). https://doi.org/10.1109/ICACCI.2017.8126018
29. Wang, H., Gu, J., Wang, S.: An effective intrusion detection framework based on svm with feature augmentation. Knowledge-Based Systems **136**, 130–139 (2017)
30. Xu, W., Fan, Y., et al.: Intrusion detection systems based on logarithmic autoencoder and xgboost. Security and Communication Networks **2022** (2022)
31. Yang, L., Moubayed, A., Shami, A.: Mth-ids: A multitiered hybrid intrusion detection system for internet of vehicles. IEEE Internet of Things Journal **9**(1), 616–632 (2021)
32. Yang, Y., Zheng, K., Wu, C., Yang, Y.: Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. Sensors **19**(11), 2528 (2019)