Explainable AI and Deep Autoencoders Based Security Framework for IoT Network Attack Certainty

Chathuranga Sampath Kalutharage[1], Liu.X[1][0000−0002−7612−9981], and Chrysoulas.C[1][0000−0001−9817−003X]

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

Abstract. Over the past few decades, Machine Learning (ML)-based intrusion detection systems (IDS) have become increasingly popular and continue to show remarkable performance in detecting attacks. However, the lack of transparency in their decision-making process and the scarcity of attack data for training purposes pose a major challenge for the development of ML-based IDS systems for Internet of Things (IoT). Therefore, employing anomaly detection methods and interpreting predicted results in terms of feature contribution or performing feature-based impact analysis can increase stakeholders confidence. To this end, this paper presents a novel framework for IoT security monitoring, combining deep autoencoder models with Explainable Artificial Intelligence (XAI), to verify the credibility and certainty of attack detection by ML-based IDSs. Our proposed approach reduces the number of black boxes in the ML decision-making process in IoT security monitoring by explaining why a prediction is made, providing quantifiable data on which features influence the prediction and to what extent, which are generated from SHaply Adaptive values exPlanations (SHAP) linking optimal credit allocation to local explanations. This was tested using the USB-IDS benchmark dataset and a detection accuracy of 84% (benign) and 100% (attack) was achieved. Our experimental results show that integrating XAI with the autoencoder model obviates the need of malicious data for training purposes, but can provide attack certainty for detected anomalies, proving the validity of the proposed methodology.

Keywords: IoT Security · Anomaly Detection · Explainable AI.

1 Introduction

Since IoT devices are connected through the Internet, there is a high possibility that they are vulnerable to cyberattacks such as impersonate, interception and penetration by unauthorized users and viruses [24]. So these devices require a proper security mechanism. Since traditional signature-based intrusion detection systems (IDS) are no longer effective at detecting attacks, as modern attacks are

sophisticated and complex, most IoT security research is currently based on Artificial Intelligence. Since ML systems are iterative and dynamic, advanced solutions based on ML are better suited to detect and mitigate the impact of cyberattacks and potential threats to IoT data and infrastructures [13].

Most of the ML-based solutions proposed in the literature are supervised learning methods that require labeled training data on attack and benign activities with certainty in ground truth. However, labeled attack data is expensive to obtain, and legal, ethical, and privacy concerns may not allow realistic data to be shared across research communities. Therefore, the use of anomaly-based detection methods is encouraged in the security field, as these models can be trained using benign data only. The main drawback of the anomaly-based method is that it often triggers false positives since it flags all unusual patterns as potential attacks even when they are not [9]. Understanding the reasons for instance prediction can reduce these false alarms and be the first step for domain experts to make decisions to prevent future attacks. Moreover, most ML-based mechanisms in security applications solve the attack detection problem and only give results whether it is an attack (anomaly) or not, and often work as a black box for the end user without providing much details on their decision-making process [20]. As a result, in operational environments, interpreting IDS outputs from the operator's point of view and transferring them into actionable reports is a challenge. Therefore, explaining the reasons behind a model's decisions has become an integral part of IDS solutions as ML becomes much more widely used in the operation of critical systems, to the point that governments are beginning to include it in legislation [11]. The ML community has recently concentrated on developing XAI methods that are easier for users to understand [12]. XAI uses natural language explanations and visualizations to show how the machine learning model arrived at its decisions.

To overcome the aforementioned limitations in IoT security monitoring, this paper presents a novel framework, combining deep autoencoder models with XAI, to identify the most influential features of anomalous behavior that violates predefined cybersecurity policies. Explainable models help to understand and diagnose the decisions made by the model, thereby increasing confidence in the data-driven IoT network security model. A domain expert can easily interpret the decisions offered by explainable models since it simplifying the knowledge discovery process. The main contributions of this research therefore are as follows. The model will detect anomalies in the IoT network and the model will demonstrate the certainty of the detected anomaly rather than providing a false attack. Additionally, the model will demonstrate the most influential features with a weight for each anomalous behavior. This model decision-making process (model explainability) can be mapped to domain expert knowledge for greater attack certainty. Thus, consequently, the model meets all the fundamental needs of modern IoT networks, providing accurate, reliable and transparent anomaly detection.

The rest of the paper is organized as follows: Section 2 presents an overview of background and related work. Section 3 describes the proposed Explainable AI

and Deep Autoencoders based methodology. Section 4 describes the experimental results carried out using the USB-IDS benchmark data set and finally Section 5 concludes the paper.

2 Background and Related work

This research focuses on the development of a security framework for IoT security monitoring, combining deep autoencoders with XAI. Therefore, the work associated with each area is discussed separately in this section.

2.1 Explainable AI (XAI)

An Explainable AI (XAI) system aims to make its behavior more understandable to humans by providing explanations. There are several XAI concepts that can be used to help develop more efficient and human-understandable AI systems [3]. The XAI system should be able to describe its capabilities and concepts, as well as what it has done, what it is currently doing, and what will happen next. It must also be able to reveal the key information on which it acts on [3]. Several ML-based IDSs have been proposed over the past decades to protect cyber networks from malicious threat actors with exceptional performance [23]. However, these complex models are often known as black box models and difficult to understand for end users. In the context of security, a single incorrect IDS prediction can expose systems and networks to major cyber risks. Therefore, XAI should be integrated with traditional IDS to enhance its credibility and reliability. Mahbooba et al. presented on explaining each predicted outcome by extracting rules from the decision tree trained and tested on the dataset. Only the expected results and the overall model response were explained using these extracted rules [19]. Similarly, Sinclair et al. and Ojugo et al. presented two separate papers to improve model performance [26, 21]. In this work, rules were derived using decision trees and genetic algorithms (GA). Instead of having an optimal rule, authors argued that IDSs should be created using a set of rules generated by machine learning. This concept was further expanded by Dais et al. by making decision-making processes more transparent [8]. However, none of these works focus on improving the IDS using the explanations of XAI tools.

2.2 Unsupervised Model Explanations

Clustering is a popular technique for solving unsupervised learning problems. The issue of cluster interpretability has had a poor track record of success [5]. A widely used explanation is to represent a cluster of points by their centroid or by a group of distant points in the collection [22]. When the clusters are compact or isotropic it works fine, but it fails in all other cases. Due to complex patterns in data distributions, it is unrealistic to expect isotropic data in the cyber domain. Another popular technique is to use principal component analysis (PCA) projections or T-distributed Stochastic Neighbourhood Embedding

(t-SNE) to visualise clusters in a two-dimensional network [18]. But, the connection between the clusters and the original variables is obscured by the reduction in the dimensionality of the features. Van der Maaten et al. suggest Interpretable Clustering via Optimal Trees (ICOT), in which the clusters are represented by the leaves, and decision tree (unsupervised) built using feature values [4]. Liu et al. and Lundberg et al. presented two different papers on clustering method based on decision trees [15, 17]. Both papers present a method that builds explainable clusters instead of explaining clusters generated by algorithms. Due to the aforementioned limitations, clustering would not be a suitable unsupervised method for our problem.

2.3 Explaining Anomalies

In the field of cybersecurity, unsupervised learning techniques such as anomaly detection are gaining popularity because a large number of labeled attack examples are needed for supervised learning, and new types of attacks will continue to emerge [25]. Almost as important as the model's predictive accuracy is the capacity to explain an anomaly detection methodology in critical sectors, such as infrastructure security [1]. Therefore, an effective anomaly explanation will greatly increase the usefulness of anomaly detection methods in real-world applications. Explaining outliers can significantly reduce the need of manual inspection of false alarms by security analysts. Goodal et al. presented a system for detecting and interpreting streaming anomalies in computer network traffic and logs, visualization of the contexts of the anomaly serves as the basis for the explanation [10]. Liu et al. presented a new Contextual Outlier Interpretation (COIN) method to explain existing outlier anomalies spotted by detectors [16]. Collaris created two dashboards using a combination of state-of-the-art explanatory techniques. These two dashboards allow the domain expert to understand the prediction. Explanations are based on currently used explanation techniques, including partial dependency diagrams, instance-level feature importance method, and local rule mining (a variant of LIME). Other research presents an SVM-based malware detection and explanation approach to explaining output made by recognizing the features that most strongly influence detection and verifying if the extracted features that influence a detection match common vulnerable characteristics [2]. Valerio La Gatta et al. presented the local explanation method CASTLE (Cluster-Aided Space Transformation for Local explanations), which provides decision rules proposing how the model prediction can be generalized to unseen instances and provides local information about the importance of the feature [14]. However, none of the above studies explained the IoT network anomalies detected by autoencoders, therefore, our work is unique and different from the above studies.

3 Methodology

To the best of our knowledge, most explainable approaches are developed for supervised learning methods (classification algorithms). But unlike existing ap-

proaches to explain a prediction, our goal is to develop an approach to explain an anomaly detected by an autoencoder model in the context of IoT security monitoring. To this end, we use the reconstruction error (see equation 1) of an autoencoder model to define IoT network anomalies (Anomaly Score). Anomalies are instances with a high reconstruction error values. In other words, a high difference (error) between the input and output (reconstructed) value is known as an anomaly. A threshold for the reconstruction error is estimated using a benign training dataset. If an anomaly exists in the incoming data, the explanatory model should be able to explain why this instance could not be well predicted (reconstructed) by the autoencoder model. As a result, the error is linked to an explanation and the proposed method calculates the SHAP values of the output features and compares them to the true (anomalous) values of the input.

$$L(A, A') = \sum_{i=1}^{n} (a_i - a'_i)^2 \tag{1}$$

Equation 1 denotes the computation of reconstruction error in our work. Given input row A with a set of features a_i, and its output A' with reconstructed feature values a'_i, and using an autoencoder model f, the reconstruction error of row is sum of the reconstruction errors of each feature. Then the features in error list need to be reordered in a descending order such that $|a_1 - a'_1| > |a_n - a'_n|$, to find top R features which includes a set of selected features for which the total corresponding errors indicate a modifiable percentage of $L(A, A')$. The model uses SHAP[1] values to describe which features were responsible for each of the high reconstruction errors in top R features.

In the explanation process, we first detect the anomalous instance using the model. Then we take the features with the highest reconstruction error and save them in the top R feature list. To get the SHAP values of each feature (i.e. a_i) in the list, Kernel SHAP is used. Then the result is displayed in a two-dimensional array, in which each of the rows represents the SHAP values for features in the top R features. The model divides SHAP values into two categories in the next step. One of the categories corresponds to contributing values that push the predicted value away from the input value and the other category is offsetting values that push the predicted value towards the true value. The division process is as follows. If the value of the input feature is greater than the output value, negative SHAP values are contributing features and positive values are offset features. If the output feature value is greater than the input value, positive SHAP values are contributing features and negative shape values are offset features. These steps return two list those are SHAP contributing and SHAP offsetting.

Finally, it selects the features with high SHAP values of the features in the top R features. From each row of contributing and offsetting SHAPs, we extract the highest values. Our goal is to explain the result with the most influencing

[1] https://www.kaggle.com/code/dansbecker/shap-values

features to the user to understand the reason for the anomaly. Figure 1 illustrates the proposed approach.

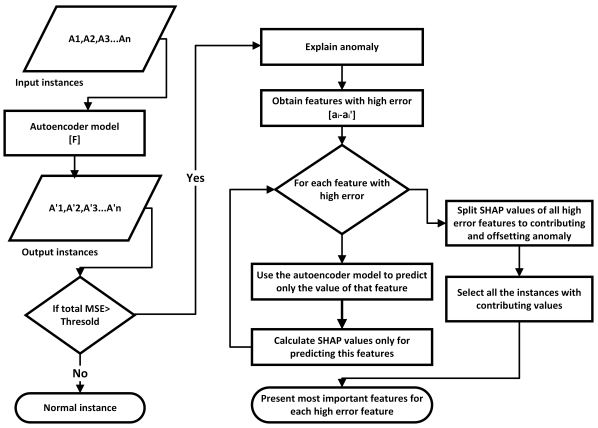

Fig. 1. Process of explanation of each anomaly detected by an autoencoder

4 Experimental Evaluation

4.1 Dataset

The USBIDS dataset [6] was used in our experimental evaluation because it provides clear feature descriptions compared to other alternative datasets. It consist of 17 csv files of labelled network flow data. A combination of denial of service (DoS) attack and defensive module consists of 16 files in addition to a benign (unaffected by an attack) network traffic data file. CIC FlowMeter[2] has used to derive network flows in the dataset. The naming convention of the 16 non-normative CSV files helps to identify the collection scenario. For example, TCPFlood-NoDefense.csv provides flows obtained by executing TCPFlood with no defense in place.

4.2 Experimental Setting

We trained the model using benign data only and two sets of attack data together with benign data were used to evaluate the model. A fully connected autoencoder model with RELU activation was used. To lighten the model, only 2 hidden layers are used in the network. The hidden layers contain 10 and 32 neurons respectively. Using benign data, the maximum Mean squared error(MSE) value set as the anomaly threshold. The proposed algorithm was implemented using

[2] https://github.com/ahlashkari/CICFlowMeter

Python 3.8 with TensorFlow and the Keras library. 40 epochs and a learning rate of 0.01 were used with the Adam optimizer. The experiments were run on a ZenBook 2.30 GHz Intel Core i7 with 16 GB of RAM.

Table 1. Results on Hulk Attack of USBIDS dataset with a comparison (Recall) to the current state of the art [7]

Detection Method	Attack Hulk No Defence	Attack Hulk Evasive	Attack Hulk reqtime-out
DT [7]	0.97	0.06	0.97
RF [7]	0.98	0.00	0.98
DNN [7]	0.67	0.05	0.66
Proposed Method	0.98	1.0	1.0

4.3 Results and Discussion

We experimented with different models to find the best performing model with the lightest architecture. Among them, the above model performed the best and the results are shown in Table 1 with a comparison to the current state of the art. In recent years, many tools and libraries are released to open black box models. However, there are no standard performance metrics to compare the performance of such algorithms. No single explainability method is better than the others. Thus, to evaluate the proposed model, we mapped XAI outcomes of our model to domain expertise. To this end, we consulted three cybersecurity experts and presented the feature set of the dataset together with attack types and asked them to rank the importance/influence of each feature in detecting the attack. For example, according to the domain experts, Forward packets per second (Fwd Packets/s), Backward packets per second (Bwd Packets/s), Flow packets per second (Flow Packets/s), Backward Packet Length Max (Bwd Packet Length Max), Packet Length Max are the most influential features in detecting a DoS attack, which comply with the out of the proposed approach. Further to model evaluation, such a list can be used in our approach to compare the XAI output with that list to further verify the certainty of the detected anomaly as an attack.

After deploying our model, we get anomalous instances as output. These anomalous instances are explained by the explainable model with an influence weight as shown in figure 3. According to this explanation, forward packets per second (Fwd packets/s) is the most influential feature (contribute) with a weight value of 0.0882. After that, Backward Packets per second, Flow Packets per second, Backward Packet Length Max have an effect with their respective values 0.0845, 0.0749, 0.016. Forward Packet Length Standard (Fwd Packet Length Std) is the offsetting feature for this anomalous instance. This offsetting features do not affect to the attack certainty as they are not contributing to the mean squared error.

We found forwarding packets per second (FWD packets/s) as one of the most influencing features by using our explainable model (anomaly instance 548271) and expertise knowledge. Then we compared the feature value of forward packets per second (FWD packets/s) respectively benign and attack classes. Packets per second feature has a value ranging between 0 and 3000 for a benign class, but in the attack state, this feature value increases up to 8000 per second. Backward packet per second also showing similar result, backward packets per second feature values vary from 0 to 3500 benign states and up to 8000 packets per second in the attack state.

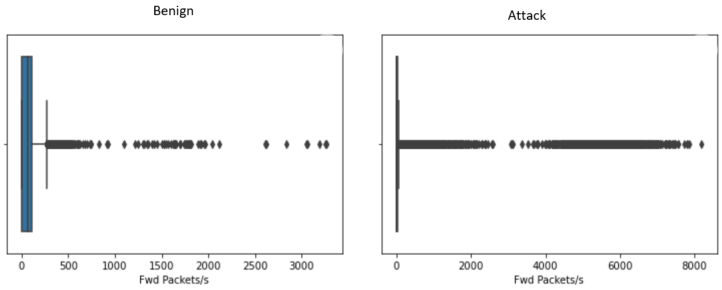

Fig. 2. Packets Per Second feature values of benign and Attack classes

Considering these facts, we can confirm that most of the influenced features explained by our model are correct. Finally, considering these results, we can confirm that our explainable model is more efficient in finding the attack certainty of the anomaly that is detected by the existing anomaly detection method.

5 Conclusion

ML-based IDSs are attracting a lot of attention from security researchers, but have limited use in the operational environment due to their black box nature. It is unclear what things contribute to their decisions, and most anomaly detection detects the anomalies, but there is no certainty about the attack. To address these issues, we have proposed a framework in which instance-wise explanations, local and global explanations, and relationships between features and system outcomes help in obtaining key decision-making features, which will eventually lead to estimate the attack certainity. By analysing the model explanations, the cybersecurity expert will also be able to make the final decision regarding the anomaly. In addition, the explanations allow the end user to better understand the decision and influencing features with weighting. In the future, we plan to extend this work to map XAI outputs to local security policies in the IoT network to detect which policy is being violated by the reported anomaly. In operational environments, this will certainly be useful for interpreting IDS

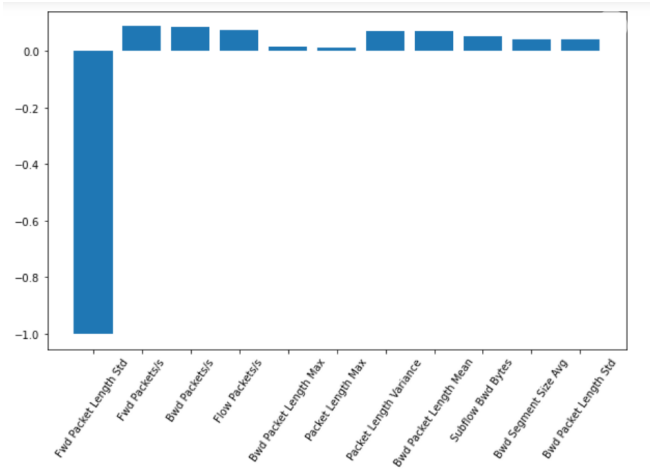

Fig. 3. Feature influenced of the selected anomalous anomalous instance (548271)

outputs from the operator's point of view and transferring them into actionable reports. Additionally, this framework will be deployed in a real IoT network environment to investigate its capabilities in production environments.

References

1. Amarasinghe, K., Kenney, K., Manic, M.: Toward explainable deep neural network based anomaly detection. In: 2018 11th International Conference on Human System Interaction (HSI). pp. 311–317. IEEE (2018)
2. Arp, D., Spreitzenbarth, M., Hubner, M., Gascon, H., Drebin, K.: Effective and explainable detection of android malware in your pocket. In: Network and Distributed System Security Symposium. pp. 1–15
3. Bellotti, V., Edwards, K.: Intelligibility and accountability: human considerations in context-aware systems. Human–Computer Interaction **16**(2-4), 193–212 (2001)
4. Bertsimas, D., Dunn, J.: Optimal classification trees. Machine Learning **106**(7), 1039–1082 (2017)
5. Bertsimas, D., Orfanoudaki, A., Wiberg, H.: Interpretable clustering via optimal trees. arXiv preprint arXiv:1812.00539 (2018)
6. Catillo, M., Del Vecchio, A., Ocone, L., Pecchia, A., Villano, U.: USB-IDS-1: a public multilayer dataset of labeled network flows for IDS evaluation. In: 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). pp. 1–6. IEEE (2021)
7. Catillo, M., Del Vecchio, A., Pecchia, A., Villano, U.: Transferability of machine learning models learned from public intrusion detection datasets: the cicids2017 case study. Software Quality Journal pp. 1–27 (2022)
8. Dias, T., Oliveira, N., Sousa, N., Praça, I., Sousa, O.: A hybrid approach for an interpretable and explainable intrusion detection system. In: International Conference on Intelligent Systems Design and Applications. pp. 1035–1045. Springer (2022)

9. Elshafie, H.M., Mahmoud, T.M., Ali, A.A.: Improving the performance of the snort intrusion detection using clonal selection. In: 2019 International Conference on Innovative Trends in Computer Engineering (ITCE). pp. 104–110 (2019)
10. Goodall, J.R., Ragan, E.D., Steed, C.A., Reed, J.W., Richardson, G.D., Huffer, K.M., Bridges, R.A., Laska, J.A.: Situ: Identifying and explaining suspicious behavior in networks. IEEE transactions on visualization and computer graphics **25**(1), 204–214 (2018)
11. Goodman, B., Flaxman, S.: European union regulations on algorithmic decision-making and a "right to explanation". AI magazine **38**(3), 50–57 (2017)
12. Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., Yang, G.Z.: Xai—explainable artificial intelligence. Science Robotics **4**(37), eaay7120 (2019)
13. 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)
14. La Gatta, V., Moscato, V., Postiglione, M., Sperli, G.: Castle: Cluster-aided space transformation for local explanations. Expert Systems with Applications **179**, 115045 (2021)
15. Liu, B., Xia, Y., Yu, P.S.: Clustering through decision tree construction. In: Proceedings of the ninth international conference on Information and knowledge management. pp. 20–29 (2000)
16. Liu, N., Shin, D., Hu, X.: Contextual outlier interpretation. arXiv preprint arXiv:1711.10589 (2017)
17. Lundberg, S.M., Erion, G.G., Lee, S.I.: Consistent individualized feature attribution for tree ensembles. arXiv preprint arXiv:1802.03888 (2018)
18. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
19. Mahbooba, B., Timilsina, M., Sahal, R., Serrano, M.: Explainable artificial intelligence (xai) to enhance trust management in intrusion detection systems using decision tree model. Complexity **2021** (2021)
20. Marino, D.L., Wickramasinghe, C.S., Manic, M.: An adversarial approach for explainable ai in intrusion detection systems. In: IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society. pp. 3237–3243. IEEE (2018)
21. Ojugo, A., Eboka, A., Okonta, O., Yoro, R., Aghware, F.: Genetic algorithm rule-based intrusion detection system (gaids). Journal of Emerging Trends in Computing and Information Sciences **3**(8), 1182–1194 (2012)
22. Radev, D.R., Jing, H., Styś, M., Tam, D.: Centroid-based summarization of multiple documents. Information Processing & Management **40**(6), 919–938 (2004)
23. Salih, A.A., Abdulazeez, A.M.: Evaluation of classification algorithms for intrusion detection system: A review. Journal of Soft Computing and Data Mining **2**(1), 31–40 (2021)
24. 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)
25. Siddiqui, M.A., Stokes, J.W., Seifert, C., Argyle, E., McCann, R., Neil, J., Carroll, J.: Detecting cyber attacks using anomaly detection with explanations and expert feedback. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2872–2876 (2019)
26. Sinclair, C., Pierce, L., Matzner, S.: An application of machine learning to network intrusion detection. In: Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99). pp. 371–377 (1999)