Explainable Intrusion Detection System in IoT Scenarios: A Cross-Device Model Training and Evaluation for Traffic Classification

Pietro Ducange
Dep. of Information Engineering
University of Pisa
pietro.ducange@unipi.it

Michela Fazzolari
Inst. of Informatics and Telematics
National Research Council
michela.fazzolari@iit.cnr.it

Francesco Marcelloni
Dep. of Information Engineering
University of Pisa
francesco.marcelloni@unipi.it

Abstract

The proliferation of Internet of Things (IoT) devices in our daily lives has raised concerns about the security of transmitted data. Due to their limited resources, IoT devices are vulnerable to malware attacks and cyber threats. Detecting and classifying these attacks is crucial to mitigate their impact. Various intrusion detection techniques have been proposed for IoT, including approaches based on Machine Learning (ML) and Artificial Intelligence (AI). Most of the ML/AI-based intrusion detection techniques, though effective, often lack transparency and trustworthiness. To address these aspects, eXplainable Artificial Intelligence (XAI) has emerged as a promising solution, providing insights on AI model decisions. In this work, we describe an explainable Intrusion Detection System (IDS) in IoT networks which embeds a multi-way Fuzzy Decision Tree (FDT) as an XAI model for traffic classification. We propose a Cross-Device training and evaluation approach in which we evaluate the generalization capability of the IDS when new devices are connected to the IoT network without retraining the FDT.

Keywords: Explainable Artificial Intelligence, Intrusion Detection Systems, Internet of Things, Trustworthy AI

This work has been partly funded by the PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI", by the project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU and by the Italian Ministry of Research (MUR) in the framework of the CrossLab and FoReLab projects (Departments of Excellence)

1. Introduction

The widespread adoption of IoT devices in our daily lives has raised concerns about the security of the data they transmit over networks. Due to their limited resources, IoT devices are more susceptible to malware attacks and other forms of cyber threats (Abdul-Ghani et al., 2018). Such attacks can have severe consequences, including data theft, physical damage, network disruption, and device malfunction. Therefore, it is crucial to detect and classify these attacks to mitigate their impact.

In the field of IoT, various intrusion detection techniques have been proposed These techniques include signature-based detection (Soe et al., 2019), behavior-based detection (Devesa et al., 2010), ML-based detection (Gaurav et al., 2022), and sandboxing approaches (Le and Ngo, 2020). In the last decades, ML-based intrusion detection techniques have gained prominence in the field of cybersecurity due to their effectiveness and flexibility in detecting cyber attacks (Asharf et al., 2020). However, many of these techniques rely on black-box ML models, such as deep learning networks and random forests, thus lacking transparency and trustworthiness, which are essential requirements in cybersecurity.

To address these limitations, eXplainable Artificial Intelligence (XAI) has emerged as a promising solution for detecting cyber attacks in IoT networks (Patil et al., 2022; Wang et al., 2020). XAI provides insights into how and why AI models make decisions, enhancing the transparency and trustworthiness of Intrusion Detection Systems (IDSs) (Arrieta et al., 2020). XAI models can be categorized into two families: explainable-by-design models and models explained using post-hoc methods. The models belonging to the first family include Decision Trees (DTs) and rule-based systems, and

URI: https://hdl.handle.net/10125/106602
978-0-9981331-7-1

have an intrinsic interpretability of their structure. As regards the second family, post-hoc methods, such as Local Interpretable Model-Agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) Values, introduced in Ribeiro et al., 2016 and Lundberg and Lee, 2017 respectively, can be used to explain black-box models such as ensemble models and Deep Neural Network (DNN).

In this work, we propose a novel experimental training and evaluation setup of an explainable IDS for IoT networks, that we have recently introduced in Fazzolari et al., 2023. Specifically, we consider an IoT network in which an IDS has been deployed on an edge node, such as an edge computing gateway (see Beniwal and Singhrova, 2022 for details on this type of gateway). The IDS was implemented by adopting a Fuzzy Decision Tree (FDT) as XAI model. In particular, we employed the model introduced in Segatori et al., 2017. This Multi-way FDT has been successfully applied in various domains, including next-generation networks (Renda et al., 2021) and smart vehicles (Aversano et al., 2022).

In our experimental analysis, first, we train the FDT embedded in the IDS, with a set of labeled historical data regarding benign and malicious network traffic data generated by the devices connected to the network. Then, we evaluate the generalization capability of the IDS whenever a new device is connected to the IoT network, without retraining the FDT. We labeled this methodology as *cross-device* model training and evaluation approach. We evaluated the effectiveness of this approach using a real dataset, namely the well-known n-BaIoT dataset Meidan et al., 2018: it consists of network traffic data from 9 IoT devices. We considered two experimental scenarios: in each scenario, we trained the FDT, used to distinguish network traffic into malicious or benign in real-time, with labeled data from 8 devices and tested it with traffic data from the remaining device. We chose to consider only 2 devices for the testing phase because only the 2 selected devices in the dataset turn out to be completely different from the remaining 7 devices. We compare the results achieved following the proposed cross-device training and evaluation approach with the ones achieved by building a specific XAI model for each device considered in the 2 experimental scenarios. In the comparison, we achieve similar results in terms of classification performance, while the complexity of the generated FDTs with the cross-device approach is always the highest.

The rest of the paper is organized as follows: in Section 2 we analyze the most recent state-of-the-art literature on AI and XAI models for IDS in IoT networks. In Section 3 we describe the proposed cross-device training and evaluation methodology and provide some details on the datasets and on FDTs adopted in our experiments. Section 4 illustrates the experimental setup and discusses the achieved results. Finally, in Section 5 we draw some conclusions.

2. Related Work

In recent years, the proliferation of IoT devices has increased the need for robust security mechanisms, including intrusion detection. IDSs are designed to detect and respond to unauthorized activities or potential threats in a network. With the unique characteristics and challenges posed by IoT networks, IDS solutions tailored for IoT environments have gained significant attention (Benkhelifa et al., 2018). In the following, we provide a synthesis of the recent state of the art regarding IDS in IoT Networks, including also some recent works which describe XAI models for cybersecurity in IoT Networks.

Traditional IDS techniques, such as signature-based detection and anomaly detection, have been applied to IoT networks. However, they face challenges and limitations, due to the resource constraints, network heterogeneity, scalability, and susceptibility to zero-day attacks inherent in IoT environments. To address these challenges, research has shifted towards more advanced approaches, such as ML-based IDSs, which leverage AI techniques to handle the complexities of IoT networks and improve detection accuracy (see Verma and Ranga, 2020 for a recent survey on this topic).

The specialized literature features several highly cited works in the field of AI models for cybersecurity in IoT networks. Notably, Meidan et al., 2018 and Aminanto et al., 2018 discuss the use of deep autoencoders for detecting anomalous network traffic and impersonation attacks, respectively. They employ stacked feature extraction and weighted feature selection to provide meaningful representations of raw input data. Yin et al., 2017 explore the use of Recurrent Neural Networks for intrusion detection, studying their performance in binary and multi-class classification tasks.

Jiang et al., 2020 propose a hybrid intrusion detection method that combines sampling strategies, Convolutional Neural Network (CNN)s for spatial feature extraction, and Long Short-Term Memory (LSTM) networks for temporal feature extraction. They utilize a hierarchical network model to address data imbalance. Banaamah and Ahmad, 2022 argue for the efficacy of DNNs in intrusion detection, comparing the performance of CNNs, LSTMs, and Gated Recurrent

Units.

While the aforementioned works prioritize accuracy over explainability, recent research has shown a growing interest in XAI for cybersecurity in IoT scenarios. Several papers and surveys (see Capuano et al., 2022; Zhang et al., 2022) provide comprehensive overviews of XAI approaches in the field. Specifically, the recent survey of Zhang et al., 2022 presents a taxonomy of XAI models and describes available datasets for cybersecurity applications.

In the realm of IDS development for IoT scenarios, recent contributions on XAI include the work of Patil et al., 2022 who propose an IDS based on ML ensemble models, incorporating the LIME technique. Houda et al., 2022 introduce a framework for intrusion detection in IoT networks, utilizing a DNN for real-time attack detection and prediction. They employ XAI post-hoc strategies such as RuleFit and SHAP to enable model transparency and provide explanations. Wang et al., 2020 propose an AI-based IDS that leverages SHAP to enhance interpretation and transparency.

Notably, all the aforementioned approaches employ post-hoc methods to achieve explainability. In contrast, the present paper exploits an explainable IDS for IoT networks based on FDT. These models are well known for their inherent global and local explainability due to their linguistic representation and the possibility of extracting semantic rules for decision-making.

3. Cross-Device Model Training and Evaluation for IDS in IoT Networks

This section provides a detailed description of the proposed cross-device approach. We envision a smart home IoT network consisting of a set of devices such as security cameras, webcams, baby monitors, and doorbells. We assume that the network is equipped with an edge computing node, such as an IoT Smart Gateway, which acts both as a modem/router for connecting the devices on the Internet, and as a computing unit close to the devices. Thanks to the IoT Smart Gateway, we can deploy our IDS on the IoT network itself, without the need of offloading the data elaboration on the cloud.

The task of IDS is to analyze the generated network traffic of IoT devices and distinguish malicious traffic from genuine traffic. In fact, devices could be infected, for example by botnets (see Ali et al., 2020 for more details on botnets in IoT networks), and could generate malicious traffic in addition to normal network traffic

In Figure 1, we show the scheme of the envisioned IoT home scenario, equipped with our explainable IDS.

The overall IDS is composed of three main modules:

- *Data collector*: it is in charge of collecting and buffering real-time network traffic generated by the different devices connected to the IoT network. It may be implemented using a network sniffing tool such as Wireshark Chaabouni et al., 2019.

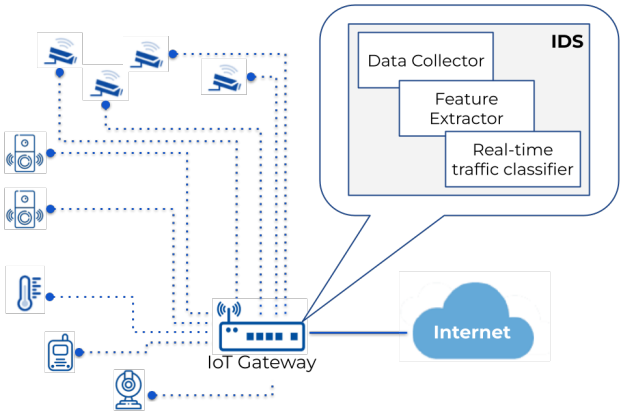

Figure 1. The Envisioned Home IoT Scenario.

- *Feature extractor*: it extracts numerical features from the network flow. This flow is composed of network packets: when a packet flows in the network, we take a snapshot of the hosts, both senders and receivers, and of the protocols used for transmitting the packets themselves by extracting a set of statistics. In our IDS, we consider the feature extraction strategy discussed in Meidan et al., 2018.

- *Real-time traffic classifier*: it discriminates genuine from malicious traffic. The latter is also classified into several predefined attack types. As mentioned above, with the aim of also providing explanations to the classification task, an XAI model by design, i.e., a multi-way FDT, was designed and experimented to be embedded in this module. To improve the transparency of the overall IDS, this module can be equipped with a logging service that can detect the decision tree paths that have been activated and adopted to classify each traffic trace.

It is worth noting that the FDT must be trained using a set of labeled data, namely a set of historical network flows representing both genuine and malicious traffic. To this aim, we suppose to have such labeled historical data generated by the set of devices connected to the network. In the literature, we have found that two approaches are usually adopted: a local classification model is built for each specific device (Cunha et al.,

2022; Tran and Dang, 2022) or a global centralized classification model, used for the classification of the traffic generated by all the devices in the network, is built using the historical data collected from all the devices connected to the network (Faysal et al., 2022). In both cases, if a new device is connected to the network, a new model training procedure or a model updating task is usually required.

In this work, we carried out an experimental analysis in which we: i) train the classification model with the historical and labelled data collected from the existing devices in the network and ii) test the generalization capability of our IDS, namely its capability of detecting malicious traffic generated by a new device, without re-training or updating the classification model. We labeled this approach as *cross-device* training and evaluation.

3.1. Datasets

In order to test the cross-device training and evaluation approach of our explainable IDS, we resort to a well-known dataset, namely n-BaIoT dataset (Meidan et al., 2018). It is publicly available from UCI's machine learning repository [1]. It includes *real* traffic data, gathered from 9 IoT devices, which include both benign and malicious traffic. In particular, the malicious traffic encompasses 10 different cyber attacks, generated by two botnets, namely BASHLITE and Mirai.

The nine IoT devices available in the dataset are: Danmini Doorbell, Ecobee Thermostat, Ennio Doorbell, Philips B120N10 Baby Monitor, Provision PT 737E Security Camera, Provision PT 838 Security Camera, Samsung SNH 1011 N Webcam, SimpleHome XCS7 1002 WHT Security Camera, and SimpleHome XCS7 1003 WHT Security Camera. The majority of these devices were infected by both BASHLITE and Mirai. However, Ennio Doorbell and Samsung SNH 1011 N Webcam were solely infected by BASHLITE. For this reason, for each device, we extracted the data belonging to the BASHLITE botnet. Therefore, the final number of classes considered is equal to 6, one representing benign traffic and 5 representing different types of cyber attacks, generated by a BASHLITE botnet. The five types of attacks in the dataset are:

- **Junk**: it consists in sending spam data;
- **Scan**: it consists in scanning the network for vulnerable devices;
- **Combo**: it identifies two attacks carried out at the same time, i.e. sending spam data and opening a connection to a specified IP address and port;
- **TCP**: it consists in TCP flooding;
- **UDP**: it consists in UDP flooding.

Table 1 describes the class distribution for each device.

[1] urly.it/3vx20

3.2. Feature Selection

In the original dataset, the network traffic generated by each device is represented by using 115 features, which are traffic statistics over several temporal windows, as discussed in Meidan et al., 2018. Specifically, these statistics include counting, mean and variance of values such as packet size and packet jitter, extracted at different time windows. Experts in IoT networks, who are the main users of our IDS, are familiar with these statistics and we expect that they may easily interpret the explanations provided in terms of linguistic rules by our MFDT. However, different types of features may be extracted from the real-time network traffic flow generated by the different devices connected to the IoT network.

However, in order to reduce the computation effort required both to extract features from network traffic and to train the classification model, in our experiments we carried out a preliminary feature selection process. Specifically, we used a filter method based on the minimum-redundancy-maximum-relevance feature evaluation strategy. It is worth to recall that filter methods for feature selection, unlike wrapper methods that adopt a specific classification algorithm for evaluating the goodness of a feature subset, are general methods independent from the classification model that will be then trained using the selected features. Moreover, filter methods are more computationally efficient than the wrapper ones. In our experiment, we adopted the Correlation-based Feature Selection Subset (CFS) Evaluator (Hall, 1999) which evaluates the worth of a subset of features by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low correlation between them are preferred.

To identify the optimal subset of features, both wrapper and filter methods should test exhaustively all the possible combinations of the features. Since the number of these combinations increases factorially with the number of features, this approach becomes unfeasible in high dimensional problems. Thus, heuristic approaches are generally adopted and in our experiments we adopted the Best First algorithm. This algorithm starts with an empty set of features and generates all possible single feature expansions. The

expansion with the best rank, calculated using CFS evaluator, is chosen and expanded in the same manner by adding single features. If expanding a subset results in no improvement, the search drops back to the next best unexpanded subset and continues from there. We used a stopping criterion of five consecutive fully expanded non-improving subsets.

We adopted the Python wrapper of the WEKA (Frank et al., 2005) implementation of the Best First algorithm with the CFS evaluator[2]. We ran the feature selection on the training sets of two cross-device scenarios composed by 8 out 9 devices included in the n-BaIoT dataset. The following 6 relevant features were identified:

1. N_PACK_100ms: Number of packets from the sending device in the past 100ms
2. $N\_PACK\_1.5s$: Number of packets from the sending device in the past 1.5s
3. N_PACK_10s: Number of packets from the sending device in the past 10s
4. AV_PACK_10s: Average number of packets from the sending device in the past 10s
5. AV_JITT_10s: Average number of traffic jitters from the sending device to the destination device in the past 10s
6. AV_JITT_1m: Average number of traffic jitters from the sending device to the destination device in the past minute

A feature sensitivity analysis for ML-based botnet detection in IoT networks, including the N-BaIoT dataset, has been discussed in Kalakoti et al., 2022. We carried out some experimental comparisons and, using the 6 features selected with our approach, we achieved similar or better results than the ones achieved considering different combinations of the input features identified in Kalakoti et al., 2022.

3.3. The Adopted XAI model as Network Traffic Classifier

In this work, we adopted the multi-way FDT, introduced in Segatori et al., 2017, as XAI model to be embedded in our IDS as Network Traffic Classifier. The FDT is a variant of the traditional decision tree that incorporates fuzzy logic principles. In a multi-way FDT, each decision node evaluates linguistic conditions instead of crisp thresholds.

Let $\mathbf{X} = \{X_1, X_2, \ldots, X_F\}$ be the set of input attributes of a dataset used for classification. Each instance $\mathbf{x_k} = \{x_{k,1}, x_{k,2}, \ldots, x_{k,F}\}$ has an associated label y_k which takes values in the set $C = \{C_1, C_2, \ldots, C_M\}$, with M denoting the number of possible classes. Consider a generic input attribute X_f defined on the universe U_f as a bounded interval in $\mathbb{R}$. Let $Z_f = \{A_{f,1}, A_{f,2}, \ldots, A_{f,T_f}\}$ be a partition over U_f consisting of T_f fuzzy sets. In this paper, we adopt strong triangular fuzzy partitions: each fuzzy set $A_{f,i}$ is fully described through a tuple ($a_{f,i}$, $b_{f,i}$, $c_{f,i}$), where $a_{f,i}$ and $c_{f,i}$ correspond to the left and right extremes of the support of $A_{f,i}$, and $b_{f,i}$ to its core. Figures 2 and 3 show, respectively, an example of strong triangular fuzzy partition and an example of multi way-FDT.

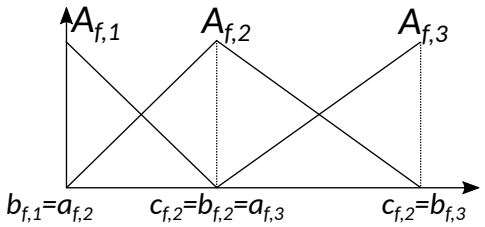

Figure 2. An example of a strong triangular fuzzy partition.

It is worth to notice that unlike binary DTs, which split the data at each node into two branches based on a specific attribute, a multi-way FDT can have multiple branches at each node.

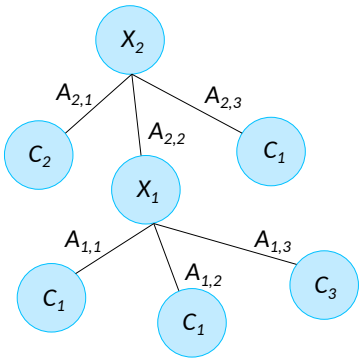

Figure 3. An example of multi-way FDT.

In our FDT, the number of branches originating from each decision node is equal to the number of fuzzy sets used for partitioning the input feature X_f selected in the node. In our experiments, we employed the fuzzy discretization algorithm described in Segatori et al., 2017, which is based on fuzzy information entropy, to generate a strong fuzzy partition for each input feature. To ensure control over the complexity and explainability level of the FDT, we set the maximum number of fuzzy sets in each fuzzy partition to T_{max}.

[2]https://fracpete.github.io/python-weka-wrapper3/api.html

Table 1. Datasets description.

Device	benign	junk	scan	combo	tcp	udp
Danmini Doorbell	40395 (12%)	29068 (9%)	29849 (9%)	59718 (18%)	85227 (25%)	100182 (30%)
Ecobee Thermostat	13111 (5%)	30312 (10%)	27494 (9%)	53012 (18%)	87877 (29%)	99195 (32%)
Ennio Doorbell	35410 (11%)	29797 (9%)	28120 (9%)	53014 (16%)	93903 (28%)	98355 (30%)
Philips Baby Monitor	165141 (36%)	28349 (7%)	27859 (6%)	58152 (13%)	85628 (19%)	100093 (22%)
Provision 737E SecCam	55169 (15%)	30898 (9%)	29297 (8%)	61380 (17%)	96658 (26%)	98424 (27%)
Provision 838 SecCam	93995 (25%)	29068 (8%)	28397 (8%)	57530 (15%)	82687 (22%)	99028 (26%)
Samsung 1011 Webcam	49559 (14%)	28305 (8%)	27698 (8%)	58669 (17%)	90454 (26%)	104683 (30%)
SimpleHome 1002 SecCam	42784 (13%)	28579 (9%)	27825 (9%)	54283 (17%)	82147 (25%)	98181 (30%)
SimpleHome 1003 SecCam	17936 (6%)	27413 (9%)	28572 (9%)	59398 (19%)	90709 (29%)	97438 (31%)

While classical Decision Tree (DT) typically use metrics such as Gini Index or Information Gain to select the most discriminative feature during the recursive tree-building process (see Breiman et al., 1984 for more details), our FDT employs the Fuzzy Information Gain metric for splitting. In our experiments, we stopped the tree-building procedure when the value of the fuzzy information gain fell below a predefined threshold of 10^{-6}, as recommended in Segatori et al., 2017.

Once the FDT has been generated, we can extract a rule base, namely a set of *linguistic if-then fuzzy rules*, following each path from the root of the tree to a leaf. Each rule has a weight which is equal to the *fuzzy cardinality* of the leaf associated with the rule itself. The fuzzy cardinality, introduced in Segatori et al., 2017, is calculated as the sum of the matching degrees of each training instance in each node from the root to the specific leaf. In practice, the weight is a sort of fuzzy support of the rule. It is worth clarifying that since the FDT is built recursively, choosing at each recursive step the most relevant feature (a feature can be chosen only once), it is impossible to generate contradictory and overlapped rules.

The rule base can be used to classify any unlabeled instance $\hat{\mathbf{x}}$. Since an unlabeled instance $\hat{\mathbf{x}}$ may simultaneously activate multiple rules, the inference process computes the *matching degree* of each rule, i.e. the strength of activation. In this paper, the instance is assigned to the class that corresponds to the maximum *association degree*, i.e. the product of the matching degrees of a specific rule and the weight associated with each class label in the rule. Using this inference strategy, a single rule can explain how classification output has been obtained, thus ensuring a high local explainability level.

As regards the global explainability of FDTs, it is apparent that large trees with numerous branches and nodes pose challenges in terms of comprehension and interpretation. Concerning local explainability, a higher value of T_{max} corresponds to an increased number of potential testing conditions within the tree and potentially deeper trees in terms of levels. These two factors impact the explainability of individual rules, particularly influencing the number of linguistic terms used in each antecedent condition and the total number of conditions in the antecedent. However, it is worth noting that when dealing with rules extracted from Fuzzy Decision Trees (FDTs), the maximum number of conditions in the antecedent of each rule is equal to the number of features selected by the feature selection process, or less, if the algorithm for building the tree stops before analyzing all the features.

More details on the adopted FDT can be found in Segatori et al., 2017.

4. Results and discussion

As stated before, in our experimental analysis, we consider two cross-device scenarios in which we: i) build the IDS system, embedding in it a multi-way FDT trained with the labeled network data extracted from 8 out of the 9 devices included in the n-BaIoT dataset and ii) evaluate the generalization capability of the system by using the network data extracted from the remaining device. Specifically, each scenario is characterized by a specific testing device, namely Ecobee Thermostat and Philips B120N10 Baby Monitor, respectively. This approach simulates the addition of a new device to an existing IoT network and the direct usage of the IDS without retraining or updating the real-time network traffic classification module.

For the sake of comparison, we also show the results achieved, for each of the two selected devices, considering a five-fold cross-validation, building a specific classification model for each device. It is worth noting that this comparison is actually fair because, for each of the two selected devices, the total number of evaluated instances in cross-validation is the same as in the cross-device approach. Through this comparison, we are able to assess how much a model trained with the cross-device approach loses, in terms of classification performance, compared to a scenario in which a specific traffic classification model is trained for each device.

Both for the cross-device scenarios and for the single-device cross-validation experiments, we carried

out the analysis of the different trade-offs between the model complexity and its capability to correctly distinguish benign network traffic from 5 different types of malicious traffic. To this aim, for each experimental analysis, 3 different FDTs have been generated setting T_{max} equal to 3, 5 and 7. It is worth recalling that the complexity of the model, expressed in terms of the number of leaves (#Leaves) and the total number of nodes (#Nodes) of the tree, is closely related to the interpretability of the model. Indeed, the greater the complexity of the FDT, the more difficult it will be to analyze its structure and understand the conditions that lead to a decision. In the table, we also show the Depth of the FDTs, which represent the maximum number of conditions that can be present in the antecedent of a rule extracted from an FDT.

Table 2 shows the results achieved for each experiment, namely for each device (Ecobee Thermostat and Philips B120N10 Baby Monitor) analyzed in cross-device and in cross-validation, respectively. Regarding the classification capability of each generated FDT, we show the average values of Precision, Recall and F1-measure metrics, weighted on the number of instances in each class. It is worth noting that for the cross-validation experiments, the table shows the average results achieved on the five folds.

As expected, results highlight that the classification performance of the FDTs increases as the number of fuzzy sets increases, but at the same time, the complexity also increases.

On the other hand, if we compare the results obtained by applying the cross-device approach with the ones obtained in cross-validation, we see that the classification capability remains more or less at the same level. However, it is worth observing that in some experimental setups, the models trained using the cross-device approach lose around 5%-7% in terms of F1-measure with respect to the models trained in cross-validation. There exists also a couple of situations in which the achieved F1-measures are slightly higher than the ones achieved in cross-validation.

As regards the complexity of the model, it is slightly higher for the cross-device approach than the cross-validation approach, but it remains of the same order of magnitude. This is possibly due to the fact that in cross-device the number of the training instances is much higher than in cross-validation.

We also analyzed the results shown in a couple of recent works in which the N-BaIoT dataset has been used (Cunha et al., 2022; Tran and Dang, 2022). Authors adopted a cross-validation evaluation strategy building a model for each device. We realized that the results obtained using the FDTs are mostly in line,

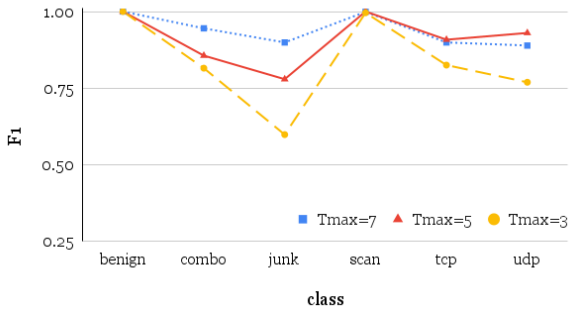

Figure 4. ET: F1-measure for each class at different values of T_{max} (Cross-device approach)

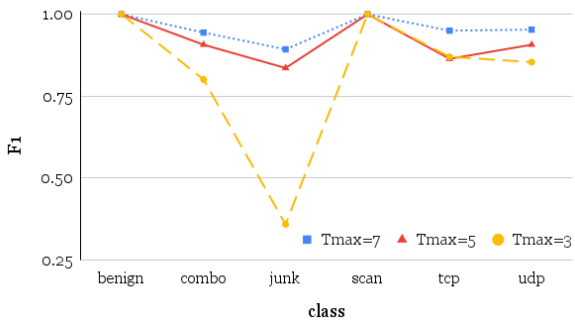

Figure 5. BM: F1-measure for each class at different values of T_{max} (Cross-device approach)

especially when considering the most complex ones, with those obtained from black-box models, namely Neural Network (NN), DNN and Random Forests.

To better understand the behaviour of each FDT experimented in the cross-device scenario, we report in Figures 4 and 5 the value of the F1-measure for each class, considering Ecobee Thermostat and Philips B120N10 Baby Monitor devices, respectively.

We observe that for both devices the F1-measure of the *benign* class is always equal to one. This indicates that this class is recognized perfectly, and no attack is misclassified as benign traffic. The F1-measure exhibits a similar pattern for the other classes across the 2 devices. For example, the F1-measure value for the *junk* class is lower than the F1-measure values of the other classes. This is particularly evident when the models with a value of T_{max} equal to 3 are considered. In this case, we observed that the *junk* class is frequently mistaken for the *combo* class. A similar situation also arises with the *tcp* and *udp* classes, as they are often confused with each other, leading to lower F1-measure values.

Table 2. Results obtained in the different experiments

Experiment	T_{max}	Precision	Recall	F1	#Leaves	#Nodes	Depth
Ecobee Thermostat Cross-device	3	0.847	0.812	0.806	17	25	4
	5	0.915	0.906	0.906	109	136	6
	7	0.929	0.917	0.917	157	183	6
Ecobee Thermostat Cross-validation	3	0.865	0.854	0.849	23	34	5
	5	0.909	0.890	0.888	65	81	5
	7	0.987	0.987	0.987	115	134	5
Philips Baby Monitor Cross-device	3	0.897	0.889	0.881	33	49	5
	5	0.944	0.934	0.933	77	96	5
	7	0.970	0.967	0.967	229	267	6
Philips Baby Monitor Thermostat Cross-validation	3	0.894	0.880	0.875	21	31	5
	5	0.950	0.948	0.945	41	51	5
	7	0.947	0.940	0.939	109	127	6

From a broad perspective, it is evident that as the complexity of the model increases, there is an improvement in the model's ability to accurately detect classes. However, this comes at the cost of reduced explainability of the FDT.

In Table 3, we show a subset of explainable linguistic rules extracted from the FDT that we used for testing the traffic generated by Ecobee Thermostat and trained using the remaining devices in the dataset. Specifically, we show, for each class, the rule associated with the highest weight. In this FDT the most discriminant feature is represented by the number of packets from the sending device in the past 1.5s. Anomalies in this feature, either low or high values, are indicative of malicious traffic. On the contrary, if the value falls within a medium range, the traffic is classified as benign. However, for traffic to be actually classified as benign, it is also necessary to consider that the average number of traffic jitters between the sending and destination devices in the past 10 seconds must be high.

In conclusion, it is important to consider that in the cross-device experimental analysis, the data used to train the model came from devices completely different from those used in the two test scenarios. Therefore, the results obtained give us confidence in the feasibility of the cross-device approach for implementing an IDS in an IoT network. Furthermore, they point out its potential for monitoring the traffic generated by newly added devices within the network. Finally, the adoption of FDT as an explainable by-design classifier will facilitate to design tools for explaining how our IDS works, such as logging services and interactive dashboards for network traffic classification with explanations.

5. Conclusions

In this paper, we discussed an Internet of Things scenario in which a set of smart devices are connected to a home network and generate packet traffic flows. With the goal of protecting the network from malicious traffic, which can be generated if some devices are infected by a botnet, we designed, developed and tested an explainable Intrusion Detection System to be implemented on an edge node of the network. The explainability of the system was ensured by incorporating a multi-way fuzzy decision tree (FDT) for real-time traffic classification.

We proposed a novel cross-device approach for FDT training and evaluation to be used to distinguish network traffic between authentic and malicious. Using this approach, FDT was constructed using historical and labeled traffic data extracted from devices connected to an IoT network. The generalization capability of the trained FDT was then evaluated using data generated by a new device added to the IoT network.

The results showed that with the proposed cross-device approach, the traffic flow generated by new devices added to an IoT network, on which our explainable IDS was deployed, can be classified correctly, without the need to retrain or update the traffic classification model.

Future research will address the following challenges:

- Conducting Additional Experimental Analyses: This includes a sensitivity analysis on the extracted and selected features and the use of other datasets related to intrusion detection in IoT networks.
- Exploring Cross-Device Procedures with Black-box models: The objective is to experiment with Intrusion Detection Systems generated by using black-box models. These models will be explained through post-hoc explainability methods. The results of this experimentation will be compared with the ones obtained by

Table 3. Example rules extracted from the FDT.

1	IF (N_PACK_1.5s is LOW) AND (AV_JITT_10s is LOW) THEN UDP
2	IF (N_PACK_1.5s is LOW) AND (AV_JITT_10s is HIGH) THEN TCP
3	IF (N_PACK_1.5s is MEDIUM) AND (AV_JITT_10s is MEDIUM) AND (AV_PACK_10s is LOW) THEN SCAN
4	IF (N_PACK_1.5s is MEDIUM) AND (AV_JITT_10s is HIGH) THEN BENIGN
5	IF (N_PACK_1.5s is HIGH) AND (N_PACK_10s is LOW) THEN COMBO
6	IF (N_PACK_1.5s is HIGH) AND (N_PACK_10s is MEDIUM) THEN JUNK

using explainable models by design, such as the multi-way fuzzy decision trees adopted in this paper.

- Addressing Concept Drift: The issue of concept drift, which arises when continuous streams of new labeled data are generated, will be analyzed. This analysis will involve experimenting with incremental learning algorithms tailored for XAI models.
- Exploring Privacy-Preserving Learning Algorithms: This involves experimentation with privacy-preserving learning algorithms, such as federated learning. The primary aim is to enable collaboration among IoT devices in building accurate and explainable models, while avoiding the need for direct data sharing."

References

Abdul-Ghani, H. A., Konstantas, D., & Mahyoub, M. (2018). A comprehensive iot attacks survey based on a building-blocked reference model. *International Journal of Advanced Computer Science and Applications*, *9*(3).

Ali, I., Ahmed, A. I. A., Almogren, A., Raza, M. A., Shah, S. A., Khan, A., & Gani, A. (2020). Systematic literature review on iot-based botnet attack. *IEEE Access*, *8*, 212220–212232.

Aminanto, M. E., Choi, R., Tanuwidjaja, H. C., Yoo, P. D., & Kim, K. (2018). Deep abstraction and weighted feature selection for wi-fi impersonation detection. *IEEE Transactions on Information Forensics and Security*, *13*(3), 621–636.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion*, *58*, 82–115.

Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W., & Wahab, A. (2020). A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions. *Electronics*, *9*(7), 1177.

Aversano, L., Bernardi, M. L., Cimitile, M., Ducange, P., Fazzolari, M., & Pecori, R. (2022). An explainable and evolving car driver identification system based on decision trees. *2022 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, 1–8.

Banaamah, A. M., & Ahmad, I. (2022). Intrusion detection in iot using deep learning. *Sensors*, *22*(21).

Beniwal, G., & Singhrova, A. (2022). A systematic literature review on iot gateways. *Journal of King Saud University-Computer and Information Sciences*, *34*(10), 9541–9563.

Benkhelifa, E., Welsh, T., & Hamouda, W. (2018). A critical review of practices and challenges in intrusion detection systems for iot: Toward universal and resilient systems. *IEEE Communications Surveys & Tutorials*, *20*(4), 3496–3509.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Capuano, N., Fenza, G., Loia, V., & Stanzione, C. (2022). Explainable artificial intelligence in cybersecurity: A survey. *IEEE Access*, *10*, 93575–93600.

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., & Faruki, P. (2019). Network intrusion detection for iot security based on learning techniques. *IEEE Communications Surveys & Tutorials*, *21*(3), 2671–2701.

Cunha, A. A., Borges, J. B., & Loureiro, A. A. (2022). Classification of botnet attacks in iot using a convolutional neural network. *Proceedings of the 18th ACM International Symposium on QoS and Security for Wireless and Mobile Networks*, 63–70.

Devesa, J., Santos, I., Cantero, X., Penya, Y. K., & Bringas, P. G. (2010). Automatic behaviour-based analysis and classification system for malware detection. *ICEIS (2)*, *2*, 395–399.

Faysal, J. A., Mostafa, S. T., Tamanna, J. S., Mumenin, K. M., Arifin, M. M., Awal, M. A., Shome, A., & Mostafa, S. S. (2022). Xgb-rf: A hybrid machine learning approach for iot intrusion detection. *Telecom*, *3*(1), 52–69.

Fazzolari, M., Ducange, P., & Marcelloni, F. (2023). An explainable intrusion detection system for iot networks. *2023 IEEE International Confernce on Fuzzy Systems (FUZZ-IEEE)*.

Frank, E., Hall, M. A., Holmes, G., Kirkby, R., Pfahringer, B., & Witten, I. H. (2005). Weka: A machine learning workbench for data mining. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook: A complete guide for practitioners and researchers* (pp. 1305–1314). Springer.

Gaurav, A., Gupta, B. B., & Panigrahi, P. K. (2022). A comprehensive survey on machine learning approaches for malware detection in iot-based enterprise information system. *Enterprise Information Systems*, 1–25.

Hall, M. A. (1999). *Correlation-based feature selection for machine learning* (Doctoral dissertation). The University of Waikato.

Houda, Z. A. E., Brik, B., & Khoukhi, L. (2022). "why should i trust your ids?": An explainable deep learning framework for intrusion detection systems in internet of things networks. *IEEE Open Journal of the Communications Society*, *3*, 1164–1176.

Jiang, K., Wang, W., Wang, A., & Wu, H. (2020). Network intrusion detection combined hybrid sampling with deep hierarchical network. *IEEE Access*, *8*, 32464–32476.

Kalakoti, R., Nõmm, S., & Bahsi, H. (2022). In-depth feature selection for the statistical machine learning-based botnet detection in iot networks. *IEEE Access*, *10*, 94518–94535.

Le, H.-V., & Ngo, Q.-D. (2020). V-sandbox for dynamic analysis iot botnet. *IEEE Access*, *8*, 145768–145786.

Lundberg, S., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems 30* (pp. 4765–4774). Curran Associates, Inc.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-baiot - network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Comput.*, *17*(3), 12–22.

Patil, S., Varadarajan, V., Mazhar, S., Sahibzada, A., Ahmed, N., Sinha, O., Kumar V C, S., Shaw, K., & Kotecha, K. (2022). Explainable artificial intelligence for intrusion detection system. *Electronics*, *11*, 3079.

Renda, A., Ducange, P., Gallo, G., & Marcelloni, F. (2021). Xai models for quality of experience prediction in wireless networks. *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 1–6.

Ribeiro, M., Singh, S., & Guestrin, C. (2016). "why should I trust you?": Explaining the predictions of any classifier. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 97–101.

Segatori, A., Marcelloni, F., & Pedrycz, W. (2017). On distributed fuzzy decision trees for big data. *IEEE Trans. Fuzzy Syst.*, *26*(1), 174–192.

Soe, Y., Feng, Y., Santosa, P., Hartanto, R., & Sakurai, K. (2019). Rule generation for signature based detection systems of cyber attacks in iot environments. *Bulletin of Networking, Computing, Systems, and Software*, *8*(2).

Tran, T. C., & Dang, T. K. (2022). Machine learning for multi-classification of botnets attacks. *2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 1–8.

Verma, A., & Ranga, V. (2020). Machine learning based intrusion detection systems for iot applications. *Wireless Personal Communications*, *111*, 2287–2310.

Wang, M., Zheng, K., Yang, Y., & Wang, X. (2020). An explainable machine learning framework for intrusion detection systems. *IEEE Access*, *8*, 73127–73141.

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, *5*, 21954–21961.

Zhang, Z., Hamadi, H. M. N. A., Damiani, E., Yeun, C. Y., & Taher, F. (2022). Explainable artificial intelligence applications in cyber security: State-of-the-art in research. *IEEE Access*, *10*, 93104–93139.