Explainable AI-based Intrusion Detection in the Internet of Things

Marios Siganos
K3Y Ltd
Sofia, Bulgaria
misganos@k3y.bg

Panagiotis Radoglou-Grammatikis
K3Y Ltd, Bulgaria, University of Western Macedonia
Sofia, Kozani, Greece
pradoglou@k3y.bg
pradoglou@uowm.gr

Igor Kotsiuba
Durham University
Durham, UK
igor.kotsiuba@durham.ac.uk

Evangelos Markakis
Hellenic Mediterranean University
Heraklion, Greece
emarkakis@hmu.gr

Ioannis Moscholios
Department Informatics and Telecommunications, University of Peloponnese
Tripoli, Greece
idm@uop.gr

Sotirios Goudos
Aristotle University of Thessaloniki
Thessaloniki, Greece
sgoudo@physics.auth.gr

Panagiotis Sarigiannidis
University of Western Macedonia
Kozani, Greece
psarigiannidis@uowm.gr

ABSTRACT

The revolution of Artificial Intelligence (AI) has brought about a significant evolution in the landscape of cyberattacks. In particular, with the increasing power and capabilities of AI, cyberattackers can automate tasks, analyze vast amounts of data, and identify vulnerabilities with greater precision. On the other hand, despite the multiple benefits of the Internet of Things (IoT), it raises severe security issues. Therefore, it is evident that the presence of efficient intrusion detection mechanisms is critical. Although Machine Learning (ML) and Deep Learning (DL)-based IDS have already demonstrated their detection efficiency, they still suffer from false alarms and explainability issues that do not allow security administrators to trust them completely compared to conventional signature/specification-based IDS. In light of the aforementioned remarks, in this paper, we introduce an AI-powered IDS with explainability functions for the IoT. The proposed IDS relies on ML and DL methods, while the SHapley Additive exPlanations (SHAP) method is used to explain decision-making. The evaluation results demonstrate the efficiency of the proposed IDS in terms of detection performance and explainable AI (XAI).

CCS CONCEPTS

• **Security and privacy → Intrusion detection systems**.

ARES 2023, August 29–September 01, 2023, Benevento, Italy
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0772-8/23/08.
https://doi.org/10.1145/3600160.3605162

KEYWORDS

Artificial Intelligence, Cybersecurity, Explainable AI, Internet of Things, Intrusion Detection

ACM Reference Format:
Marios Siganos, Panagiotis Radoglou-Grammatikis, Igor Kotsiuba, Evangelos Markakis, Ioannis Moscholios, Sotirios Goudos, and Panagiotis Sarigiannidis. 2023. Explainable AI-based Intrusion Detection in the Internet of Things. In *The 18th International Conference on Availability, Reliability and Security (ARES 2023), August 29–September 01, 2023, Benevento, Italy*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3600160.3605162

1 INTRODUCTION

The evolution of cyberthreats has been driven by advancements in technology and the increasing interconnectedness of our digital world. Initially, cyberthreats focused on individual systems and networks, characterised by viruses, worms, and basic forms of malware. However, with the rapid evolution of the Internet of Things (IoT) [3] and the proliferation of connected devices, cyberthreats have grown in sophistication and scope. Today, we face a range of evolving threats, including ransomware, Advanced Persistent Threats (APTs), social engineering, phishing, and zero-day exploits. These threats are propelled by organised cybercriminal groups, state-sponsored actors, and non-state actors seeking financial gain, political influence, or the disruption of critical infrastructures. Attacks have scaled up and become more complex, targeting individuals, businesses, governments, and Critical Infrastructure (CIs). As technology advances further, cyberthreats will continue to evolve, necessitating ongoing vigilance and proactive cybersecurity measures to safeguard the digital ecosystems.

Therefore, the role of Intrusion Detection Systems (IDS) is necessary in order to detect potential cyberattacks and anomalies in a timely manner. Based on the detection techniques, IDS can be classified into two main categories: (a) signature/specification-based detection and (b) anomaly-based detection. In the first category, pre-defined patterns are used to recognise particular cyberattacks. On the other hand, statistical analysis and Artificial Intelligence (AI) methods are used to detect cyberattacks and anomaly behaviours. Despite the fact that AI-powered IDS have already demonstrated their efficiency, they suffer from false alarms and explainability issues that do not allow security administrators to trust them [4, 10]. Therefore, in light of the aforementioned remarks, in this paper, we focus our attention on the development of an AI-powered IDS for the IoT, including explainable AI (XAI) functions. Therefore, the contributions of this paper are summarised as follows.

- **Implementation of an AI-powered IDS for the IoT**: An AI-powered IDS is implemented, using `CIC-IoT-Dataset-2022` [2] and `IEC 69870-5-104 Intrusion Detection Dataset` [9]. The first dataset refers to IoT environments, while the second one refers to Industrial IoT (IIoT) environments, where IEC 60870-5-104 is used. For both datasets, various Machine Learning (ML)/Deep Learning (DL) methods are evaluated.
- **Investigating and development of explainability functions**: An explainability mechanism about the decisions of the proposed AI-powered IDS is provided. For this purpose, the SHapley Additive exPlanations (SHAP) XAI method is investigated and utilised.

The rest of this paper is organised as follows. Section 2 discusses similar works in this field. In section 3, the architecture of the proposed IDS is provided, including technical implementation details. Next, section 4 focuses on the evaluation analysis of the proposed AI-powered IDS with explainability functions. Finally, section 5 concludes this paper.

2 RELATED WORK

Several works investigate and combine cybersecurity mechanisms with XAI. Some of them are listed in [1, 6–8, 11, 12]. In particular, in [12], the authors present an explainable AI solution for the detection of Domain Name System (DNS) over Hypertext Transfer Protocol Secure (HTTPS) (DoH) attacks. They propose a balanced and stacked Random Forest classifier for classification based on DNS over HTTPS intrusion features. For this task, they utilise the publicly available `CIRA-CIC-DoHBrw-2020` dataset. Their solution can accurately recognise DoH traffic from normal HTTPS traffic and also detect malicious DNS traffic with more than 99% accuracy. Lastly, the authors explain the results of the model utilising the SHAP method to highlight the feature contributions. They also create an interactive explainer dashboard where the users can examine the feature contribution of any data sample.

In [11], the authors propose a framework that uses ML and XAI for IDS. A one-vs-all and a multiclass classifier based on fully connected networks are trained and evaluated on the `NSL-KDD dataset`. Both classifiers achieve more than 80% accuracy on the test dataset, outperforming similar benchmark methods. The proposed framework includes the application of the SHAP method to provide local and global explanations. Another work focussing on the use of XAI for intrusion detection is [8]. The authors use a voting classifier that utilises an ensemble of several models. They apply their classifier on normal and malicious network traffic samples from the `CICIDS2017` dataset and achieve around 96% accuracy. For the explanations of individual predictions, they use the Local Interpretable Model-Agnostic Explanations (LIME) method on each of the mentioned ML models.

Similarly, Mane and Rao [6] propose a method that utilises explainable AI for the creation of a Network Intrusion Detection System (NIDS). They use a fully connected network with three hidden layers to classify samples from the `NSL-KDD` dataset as normal or attack and achieve approximately 82% accuracy and F1-score. They also apply several XAI techniques to explain the effect of input features on the detection of attacks. More specifically, they utilise the SHAP method to provide both global and local explanations of the model, the LIME method for a local explanation and the Contrastive Explanation Method (CEM) method to identify the least number of features and their values that would produce the same prediction.

In [1], the authors introduce a framework for network intrusion detection using XAI. It involves the utilisation of a Gradient Boosting (XGboost) model for supervised learning, followed by the use of the SHAP method to explain the predictions. The authors also propose the use of a deep autoencoder as an unsupervised method which is trained on the explanations produced from the application of SHAP. Lastly, they evaluate the proposed method using the `NSL-KDD` dataset and achieve an accuracy of 93%.

Finally, in [7], Marino et al. introduce an adversarial approach to applying XAI in IDS. The proposed method can be employed on models with established gradients to determine the minimum modifications necessary for correctly classifying a set of previously misclassified examples. By altering the input features, visualisations are generated to highlight the features that significantly influenced the incorrect classifications. The study utilises the `NSL-KDD` dataset, partitioned into 124,926 training samples and 16,557 testing samples. Two models, a Linear classifier and a Multi-Layer Perceptron (MLP) classifier, achieve accuracy rates of 93.6% and 95.5%, respectively, on the test dataset. The authors calculate the minimal adjustments required to rectify the classifier's output and present the resulting explanations to the user through easily interpretable plots. Notably, the proposed adversarial approach can be applied to models with defined gradients without necessitating any modifications to the model structure.

Undoubtedly, the previous works provide useful solutions and methodologies. However, it is worth mentioning that none of them considers the unique characteristics of IoT and IIoT network environments of CIs, such as the smart electrical grid. In this paper, we plan to cover this gap by combining ML/DL methods and SHAP in order to detect cyberattacks against IoT and IEC 60870-5-104 IIoT environments, considering (a) Transmission Control Protocol/Internet Protocol (TCP/IP) flow statistics and application-layer flow statistics related to IEC 60870-5-104. IEC 60870-5-104 is an industrial protocol widely used in CIs, especially in the energy domain. Therefore, on the one hand, we show how SHAP behaves with respect to TCP/IP flow statistics and on the other hand, we investigate its applicability with IEC 60870-5-104 flow statistics.

3 ARCHITECTURAL DESIGN

As illustrated in Fig. 1, the architecture of the proposed IDS consists of six modules: (a) Network Traffic Data Capturing Module, (b)

Network Flow Generation Module, (c) Data Pre-processing Module, (d) Detection Module, (e) Explainability Module and (f) Notification Module. This first module is responsible for capturing the network traffic data (i.e., `pcap files`). For this purpose, a Switched Port Analyser (SAPN) (i.e., port mirroring) and `tcpdump` are used. Once the raw network traffic data is captured, the next module is responsible for generating the flow statistics. This step reduces the volume of data and provides a more meaningful representation of the network traffic data. Given that we have two kinds of environments: (a) IoT and (b) IEC 60870-5-104 IIoT, two kinds of flow statistics are generated: (a) TCP/IP network flow statistics and (b) IEC 60870-5-104 payload flow statistics. The first kind refers to bidirectional flow statistics related to TCP/IP attributes. These statistics are generated through `CICFlowMeter` and `NFStream`. On the other side, the second refers to bidirectional flow statistics related to the payload of the IEC 60870-5-104 packets. To this end, a custom Python flow generator is used.

The flow statistics are often noisy and contain redundant information. Therefore, the Data Pre-Processing module is utilised for cleaning the data, removing noise, and reducing the feature dimensionality. The Data Pre-Processing Module performs tasks such as feature scaling, feature selection, and feature extraction. More specifically, this module includes the handling of missing values where the records/rows with missing values are removed, the handling of label/target where the categorical values are encoded with numerical ones and the handling of categorical features where any categorical features remaining are also removed. Regarding the selection of features, some of the steps that are followed are the removal of features with only one unique value, low variance (0.1) or Pearson correlation (0.9). Recursive feature elimination and sequential feature selection (forward and backward) are also performed. Finally, the data are scaled to the range [0, 1] except from the case of the DL methods, where the features were standardised by removing the mean and scaling to unit variance.

The core of the proposed IDS is the Detection Module, which uses pre-trained ML/DL models to discriminate potential attacks. Based on the pre-processing step, two complement ML/DL models are used: (a) ML/DL model based on TCP/IP network flow statistics and (b) ML/DL model based on IEC 60870-5-104 payload flow statistics.

The explainability module is responsible for providing consistent and reliable explanations for the predictions of the detection module. Explainability is important to build trust in the system, to better understand the detected threats and identify the root cause of a detected intrusion.

This module focuses on the use of model-agnostic post-hoc explainability techniques to explain the results of any pre-trained ML/DL model that is adopted in the Detection Module regardless of the type and architecture of the ML model. More specifically, it leverages the SHAP method [5]. SHAP is based on the concept of Shapley values from cooperative game theory and assigns a value (importance value) to each feature in a prediction, indicating its contribution to the final outcome. This is done by computing the average contribution of each feature across all possible combinations of features. The explanation can be defined as:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{1}$$

where $z' \in 0, 1$, M: the number of simplified input features, and $\phi_i \in \mathbb{R}$. ϕ_0 is the null (average) output of the model, z' is the simplified binary input vector, and ϕ_i is the explained effect of feature i. The formula for calculating the Shapley values that are used as feature attributions is as follows:

$$\phi_i(x) = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]] \tag{2}$$

where $\phi_i(x)$ represents the Shapley value for feature i and instance x, F is the set of all features, S is a coalition, which is a subset of F that does not contain feature i, $f_S(x_S)$ represents the model's prediction for instance x using the features in coalition S, $f_{S \cup \{i\}}(x_{S \cup \{i\}})$ represents the model's prediction for instance x using the features in coalition S along with feature i, $|S|$ denotes the number of elements of the coalition S and $|N|$ denotes the total number of features.

The formula 2 calculates the average contribution of feature i across all possible coalitions S by comparing the predictions with and without the feature. It considers all possible ways of including feature i in different coalitions and computes the difference in predictions when i is added. The sum of these differences is weighted based on the number of coalitions of different sizes to determine the Shapley value for feature i and instance x.

SHAP provides both local and global explanations. Local explanations focus on explaining individual predictions by assigning importance values to each feature for a specific instance. This helps understand the contribution of each feature to a particular prediction. On the other hand, global explanations, on the other hand, provide an overview of feature importance across the entire dataset by aggregating the local explanations. They help identify the consistent and overall impact of features on model predictions.

It is worth mentioning that the explainability module provides explanations through a dashboard that offers visualizations that illustrate the importance of different features. Feature importance plots highlight the most influential features in the decision-making process. These visualisations provide a clear and intuitive representation of the model's behaviour, allowing cybersecurity analysts to identify patterns, anomalies, and potential vulnerabilities in the network traffic data. Finally, once an intrusion is detected, the Notification Module is responsible for alerting the security administrator. The notification can be in the form of an email, Short Message/Messaging Service (SMS), push notifications or a dashboard that displays the intrusion details and explanations.

4 EVALUATION ANALYSIS

In this section, we focus on the evaluation of the proposed explainable AI-based IDS to determine its detection performance. For the detection of an attack/intrusion, we perform multi-class classification with the attacks as classes/labels. Regarding the evaluation metrics, the following are considered: (a) Accuracy (Equation 3), (b) True Positive Rate (TPR) (Equation 4), (c) False Positive Rate (FPR) (Equation 5) and (d) F1-score (Equation 6).

The accuracy metric (Equation 3) quantifies the ratio of correct classifications to the total number of instances. It is a suitable evaluation measure when the training dataset is well-balanced, indicating an equal distribution of instances across all classes.

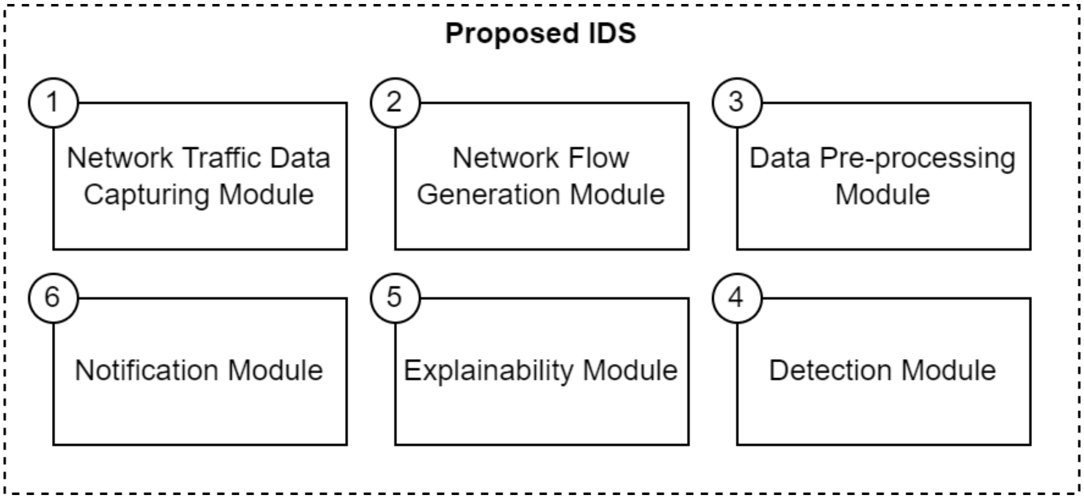
Proposed IDS
1
Network Traffic Data Capturing Module
2
Network Flow Generation Module
3
Data Pre-processing Module
6
Notification Module
5
Explainability Module
4
Detection Module

Figure 1: IDS architecture showing the included modules

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

where:

$$TP \rightarrow \text{True Positives}$$
$$TN \rightarrow \text{True Negatives}$$
$$FP \rightarrow \text{False Positives}$$
$$FN \rightarrow \text{False Negatives}$$

TPR, as defined by Equation 4, quantifies the proportion of actual intrusion instances that were accurately detected and classified as intrusions.

$$TPR = \frac{TP}{TP + FN} \tag{4}$$

FPR, as described by Equation 5, signifies the ratio of normal instances that were mistakenly classified as cyberattacks. It highlights the trade-off between correctly identifying normal instances and the occurrence of false alarms.

$$FPR = \frac{FP}{FP + FN} \tag{5}$$

F1 score (Equation 6) is a metric that combines the true positive rate (TPR) and precision to provide a balanced assessment. Precision represents the proportion of true positives out of the sum of true positives and false positives.

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{6}$$

For the performance evaluation, a detailed ML/DL comparative analysis is performed. The ML/DL methods used in this analysis are `Naive Bayes`, `Support Vector Machines (SVM)` with both `linear` and `Radial Basis Function (RBF)` kernel, `Decision Trees (DT)`, `Random Forest (RF)`, `Gragient Boosting (XGBoost)`, `Adaboost`, `Logistic Regression`, `Quadradic Discriminant Analysis (QDA)` and a `Dense Deep Neural Network (DNN)`. The `numpy`, `pandas`, `scikit-learn (sklearn)`, `tensorflow/keras` and `shap` Python libraries are utilised for the experimental results. Finally, it is noteworthy that the evaluation process includes the models' explainability to ensure that the system is transparent and understandable to the end-users.

4.1 Datasets

For the experiments and the evaluation of detection performance, two balanced datasets are utilised. The first one is the `IEC 60870-5-104 Intrusion Detection Dataset` [9], which can be used to investigate and assess the severity of cyberattacks against the IEC 60870-5-104 protocol. It includes labelled TCP/IP network flow statistics (generated through CICFlowMeter) and IEC 60870-5-104 payload flow statistics (generated through a custom Python flow generator). More specifically it contains the following cyberattacks: Man In the Middle (MITM), traffic sniffing, C_RD_NA_1, C_CI_NA_1, C_RP_NA_1, C_SC_NA_1, C_SE_NA_1, M_SP_NA_1_DOS, C_CI_NA_1_DOS, C_SE_NA_1_DOS, C_RD_NA_1_DOS, C_RP_NA_1_DOS. Cyberattacks 3-6 refer to IEC 60870-5-104 unauthorised access, while cyberattacks 6-12 are related to Denial of Service (DoS) IEC 60870-5-104 cyberattacks. The second dataset is the `CIC-IoT-Dataset-2022` [2]. It can be used for profiling, behavioural analysis, and vulnerability testing of different IoT devices. The dataset contains the network traffic of various IoT devices, including WiFi, ZigBee, and Z-Wave devices, where 1) a flood denial-of-service attack and 2) an RTSP brute-force attack were performed. Two separate data files are used, one that refers to `CICFlowMeter` and one related to `NFStream`.

4.2 Experimental Results

Our experiments begin with an Exploratory Data Analysis (EDA), which is a required step to obtain useful insights into the data before training the ML/DL models. Then, we proceed with the pre-processing of the data, where we follow the steps described in the Pre-processing Module. After making sure that they are in

Table 1: Evaluation results of the proposed IDPS - IEC 60 870-5-104 - CICFlow

AI Models	Accuracy	TPR	FPR	F1-Score
Naive Bayes	0.4196	0.4196	0.0512	0.3554
SVM Linear	0.4944	0.4944	0.0453	0.4727
SVM RBF	0.4940	0.4940	0.0448	0.4538
Decision Trees	0.6007	0.6009	0.0363	0.5994
Random Forest	0.6632	0.6634	0.0306	0.6601
XGBoost	0.6358	0.6360	0.0330	0.6324
Adaboost	0.3532	0.3532	0.0574	0.3014
Logistic Regression	0.4841	0.4841	0.0463	0.4628
Quadradic Discriminant Analysis	0.5572	0.5572	0.0395	0.5236
DNN	0.5811	0.5811	0.0381	0.5586

Table 2: Evaluation results of the proposed IDPS - IEC 60 870-5-104 - Custom

AI Models	Accuracy	TPR	FPR	F1-Score
Naive Bayes	0.5582	0.5582	0.0402	0.4749
SVM Linear	0.6514	0.6514	0.0317	0.6384
SVM RBF	0.5942	0.5942	0.0369	0.5588
Decision Trees	0.8333	0.8333	0.0152	0.8281
Random Forest	0.8521	0.8521	0.0134	0.8473
XGBoost	0.8348	0.8348	0.0150	0.8280
Adaboost	0.2826	0.2826	0.0652	0.2121
Logistic Regression	0.6223	0.6223	0.0343	0.6053
Quadradic Discriminant Analysis	0.6233	0.6233	0.0342	0.5594
DNN	0.6958	0.6958	0.0277	0.6851

Table 3: Evaluation results of the proposed IDPS - CIC IoT dataset 2022 - CICFlow

AI Models	Accuracy	TPR	FPR	F1-Score
Naive Bayes	0.7428	0.7427	0.1287	0.7409
SVM Linear	0.9312	0.9311	0.0344	0.9314
SVM RBF	0.9583	0.9583	0.0209	0.9585
Decision Trees	0.9985	0.9985	0.0007	0.9985
Random Forest	0.9983	0.9983	0.0008	0.9983
XGBoost	0.9992	0.9992	0.0004	0.9992
Adaboost	0.9583	0.9583	0.0208	0.9582
Logistic Regression	0.9308	0.9308	0.0346	0.9311
Quadradic Discriminant Analysis	0.9363	0.9363	0.0319	0.9364
DNN	0.9888	0.9888	0.0056	0.9888

a suitable format, we train our models. The default values for the hyper-parameters are used for all of the models except for the case of DT, RF and XGBoost, where we perform hyper-parameter tuning. Regarding the dense DNN, it consists of 4 layers where the activation function for the input and hidden layers is ReLu and for the output layer is Softmax. The output layer has a number of units equal to the number of classes in the target feature and, more specifically, 12 units in the case of the `IEC 60870-5104 Intrusion Detection Dataset` and three units for the `CIC IoT Dataset 2022`. The chosen optimiser is Adam, and categorical cross-entropy is used as the loss function. The DNN is trained for 100 epochs with a batch size of 32. Early stopping is utilised to reduce overfitting without compromising the model's accuracy. Lastly, the model weights from the epoch with the smaller validation loss are chosen. A different model is created for each dataset and each ML method. Lastly, the accuracy and the macro-averaged scores of the rest evaluation metrics are computed by calculating the per-class values of TPR, FPR and the F1 score for each class and then averaging them to get the overall performance.

Tables 1 and 2 show the performance of the ML methods using the IEC 60870-5-104 dataset and, more specifically, the TCP/IP network flow statistics and the IEC 60 870-5-104 payload flow statistics, respectively. Six different flow timeouts (15, 30, 60, 90, 120, and 180 s) are evaluated for both statistics, and those that provide

Table 4: Evaluation results of the proposed IDPS - CIC IoT dataset 2022 - NFStream

AI Models	Accuracy	TPR	FPR	F1-Score
Naive Bayes	0.9700	0.9700	0.0150	0.9701
SVM Linear	0.9581	0.9581	0.0209	0.9583
SVM RBF	0.9879	0.9879	0.0060	0.9879
Decision Trees	0.9988	0.9988	0.0006	0.9988
Random Forest	0.9999	0.9999	0.0000	0.9999
XGBoost	0.9998	0.9998	0.0001	0.9998
Adaboost	0.9106	0.9106	0.0447	0.9112
Logistic Regression	0.9620	0.9620	0.0190	0.9621
Quadradic Discriminant Analysis	0.5530	0.5530	0.2235	0.5051
DNN	0.9985	0.9985	0.0007	0.9985

Table 5: Datasets details

Dataset	Parser	Timeframe	# Columns	# Rows (train)	# Rows (test)
IEC 60870-5-104	cicflow	15	84	10968	4692
IEC 60870-5-104	cicflow	30	84	7980	3420
IEC 60870-5-104	cicflow	60	84	5904	2520
IEC 60870-5-104	cicflow	90	84	5088	2172
IEC 60870-5-104	cicflow	120	84	4800	2028
IEC 60870-5-104	cicflow	180	84	3588	1536
IEC 60870-5-104	custom	15	112	10968	4692
IEC 60870-5-104	custom	30	112	7980	3420
IEC 60870-5-104	custom	60	112	5904	2520
IEC 60870-5-104	custom	90	112	5088	2172
IEC 60870-5-104	custom	120	112	4800	2028
IEC 60870-5-104	custom	180	112	3588	1536
CIC-IoT-Dataset-2022	cicflow	-	84	29814	12999
CIC-IoT-Dataset-2022	nfstream	-	40	25533	10845

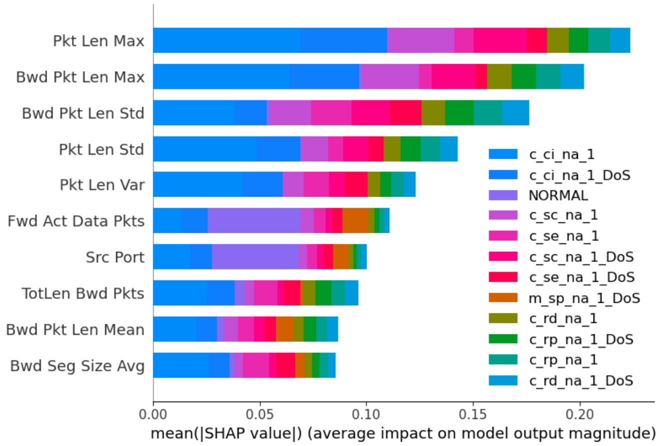

Figure 2: IEC 60 870-5-104 - CICFlow - SHAP Summary Plot

the optimal detection performance are chosen. In the first case, the best performance is achieved when the flow timeout is 180s, while in the second case, the best performance is achieved when the flow timeout is equal to 120s. In both cases, the best-performing method is Random Forest, with XGBoost and DT following closely.

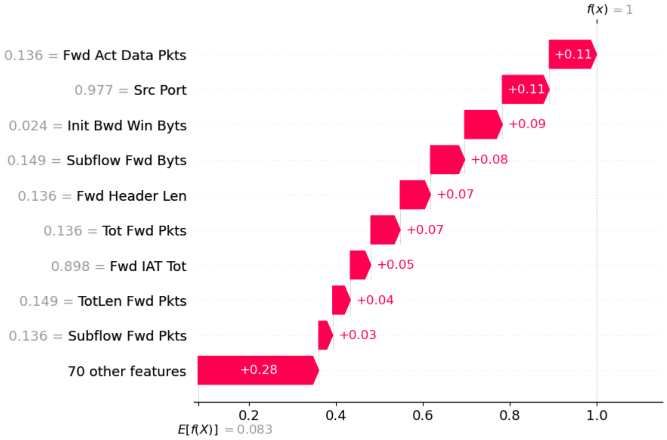

Figure 3: IEC 60 870-5-104 - CICFlow - SHAP Waterfall Plot

Tables 3 and 4 summarise the evaluation results using the `CIC IoT Dataset 2022` for both `CiCFlowMeter` and `NFStream`. In the

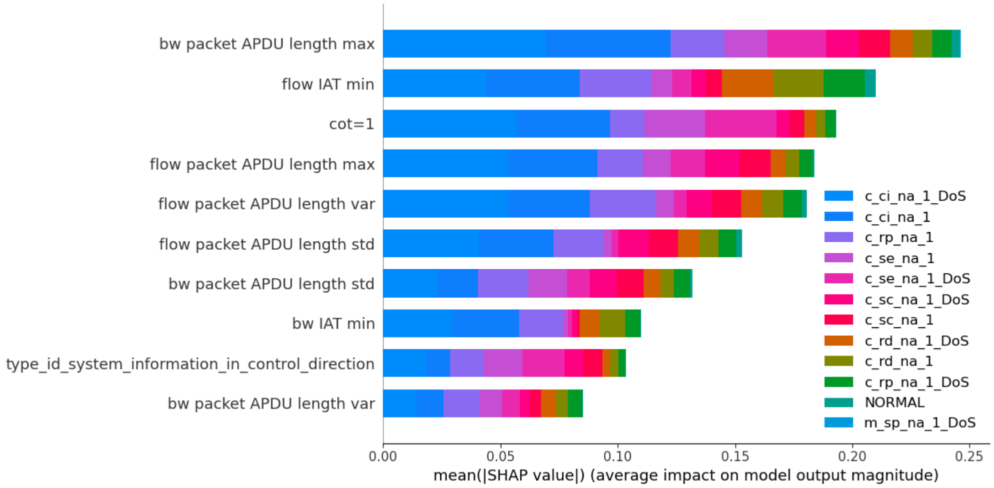

Figure 4: IEC 60 870-5-104 - Custom - SHAP Summary Plot

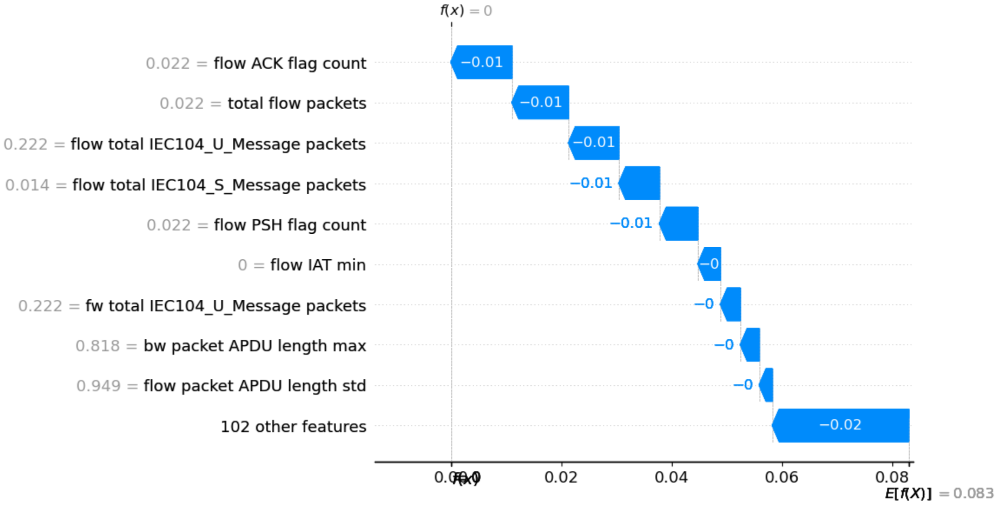

Figure 5: IEC 60 870-5-104 - Custom - SHAP Waterfall Plot

first case, XGBoost achieves the highest F1-score, while RF, DT, and DNN are very close. In the second case, RF performs the best detection performance, while XGBoost, DNN, and DT have similar performance.

Regarding the explainability functions through the SHAP method, we generate summary and waterfall plots for the best-performing model for each dataset with respect to the intrusion detection task. Figures 2, 4, 6, and 8 show the SHAP summary which depicts the feature importance based on the SHAP values. The top 10 features are listed top-down with decreasing importance. Each bar's length shows the mean absolute SHAP value that represents the average absolute impact of the feature on the final prediction. In this multi-class classification task, the bars are stacked and show the values for each one of the output classes separately.

Figures 3, 5, 7, and 9 show the explanation of a single prediction given a specific class as a waterfall plot. These plots contain the feature values on the y-axis and arrows, which show the feature contribution (positive or negative) to the prediction. More specifically, they show how this contribution moves the value from the expected output (based on the background data distribution) to the final model output for this prediction. The features appear in descending order based on the magnitude of their SHAP values. $f(x)$ is the model predicted probability value and $E[f(x)]$ is the base (expected) value. The sum of all SHAP values will be equal to $E[f(x)] - f(x)$.

5 CONCLUSIONS

The evolution of cyberthreats has witnessed a relentless progression over time. As technology enablers have advanced, cyberattackers have adapted their strategies and techniques to exploit vulnerabilities and compromise systems. Hence, the role of IDS is crucial. It is

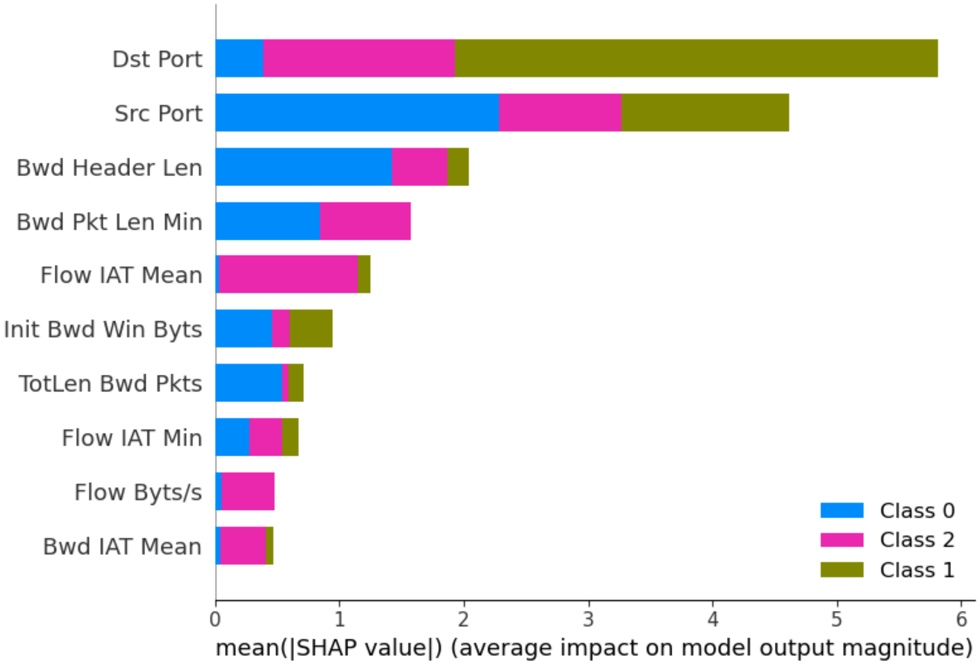

Figure 6: CIC IoT dataset 2022 - CICFlow - SHAP Summary Plot

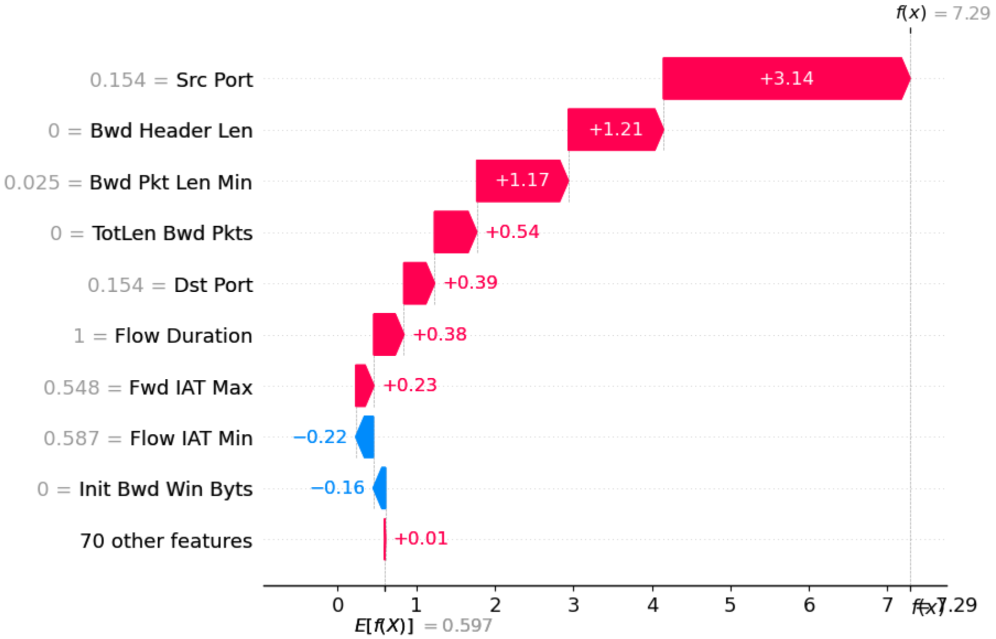

Figure 7: CIC IoT dataset 2022 - CICFlow - SHAP Waterfall Plot

evident that AI can be used to detect potential cyber-attacks and unknown anomalies; nevertheless, AI-powered IDS are still characterised by false alarms and explainability issues. Therefore, their continuous improvement is necessary. In this paper, we introduce an AI-powered IDS for the IoT, including XAI functions. According to the evaluation results, the proposed IDPS can effectively detect malicious activities against IoT and IEC 60870-5-104 IIoT environments. Finally, the SHAP-based XAI functions show the feature

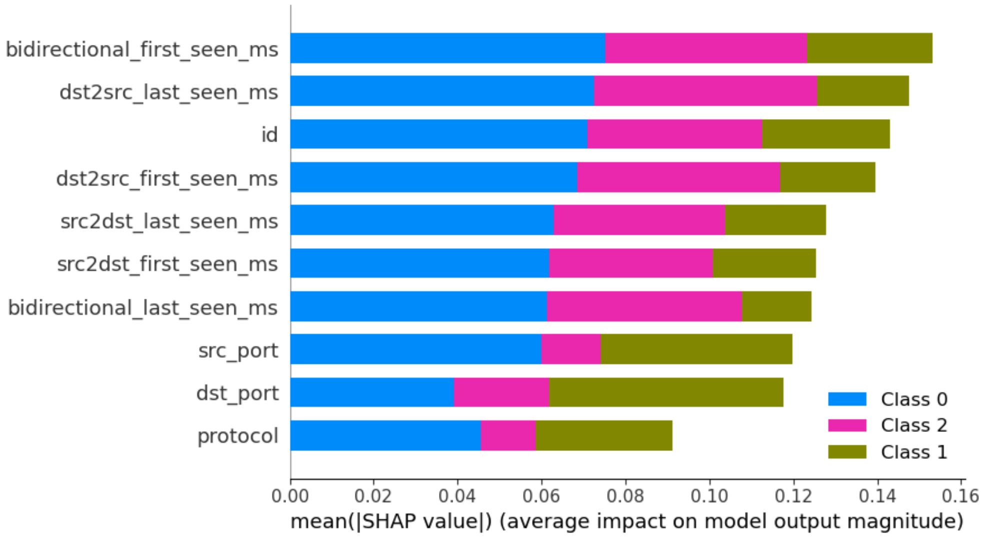

Figure 8: CIC IoT dataset 2022 - NFStream - SHAP Summary Plot

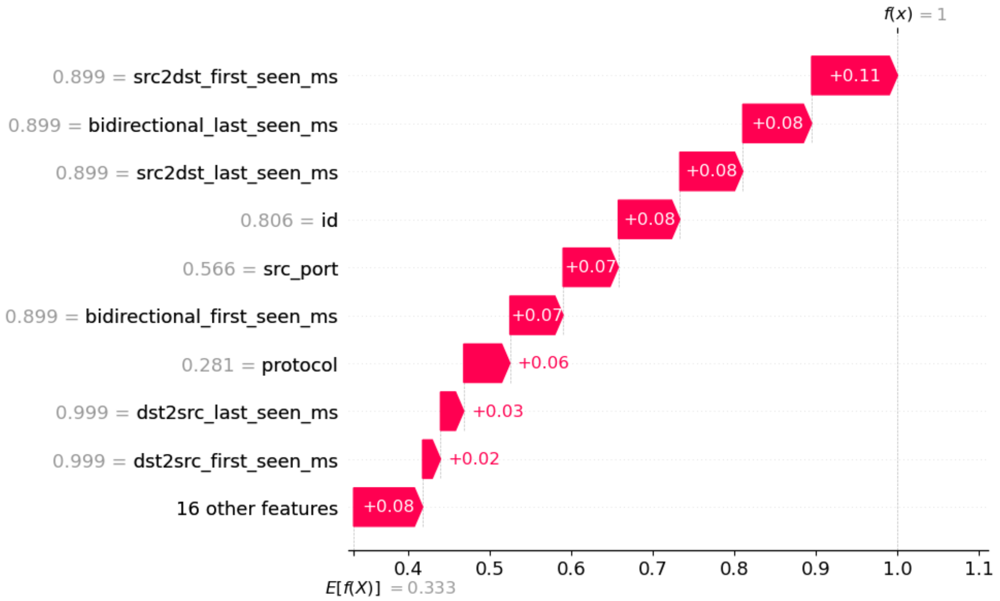

Figure 9: CIC IoT dataset 2022 - NFStream - SHAP Waterfall Plot

importance for each decision, thus allowing the security administrator and cybersecurity analysts to understand decision-making better and trust the proposed AI-powered IDS.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101070214 (TRUSTEE). Disclaimer: Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them.

REFERENCES

[1] Pieter Barnard, Nicola Marchetti, and Luiz A. DaSilva. 2022. Robust Network Intrusion Detection Through Explainable Artificial Intelligence (XAI). *IEEE Networking Letters* 4, 3 (2022), 167–171. https://doi.org/10.1109/LNET.2022.3186589

[2] Sajjad Dadkhah, Hassan Mahdikhani, Priscilla Kyei Danso, Alireza Zohourian, Kevin Anh Truong, and Ali A Ghorbani. 2022. Towards the development of a realistic multidimensional IoT profiling dataset. In *2022 19th Annual International Conference on Privacy, Security & Trust (PST)*. IEEE, Fredericton, NB, Canada, 1–11. https://doi.org/10.1109/PST55820.2022.9851966

[3] Panagiotis I Radoglou Grammatikis, Panagiotis G Sarigiannidis, and Ioannis D Moscholios. 2019. Securing the Internet of Things: Challenges, threats and solutions. *Internet of Things* 5 (2019), 41–70.

[4] Swetha Hariharan, RR Rejimol Robinson, Rendhir R Prasad, Ciza Thomas, and N Balakrishnan. 2022. XAI for intrusion detection system: comparing explanations based on global and local scope. *Journal of Computer Virology and Hacking Techniques* 19 (2022), 1–23.

[5] Scott Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. http://arxiv.org/abs/1705.07874

[6] Shraddha Mane and Dattaraj Rao. 2021. Explaining Network Intrusion Detection System Using Explainable AI Framework. https://doi.org/10.48550/ARXIV.2103.07110

[7] Daniel L Marino, Chathurika S Wickramasinghe, and Milos Manic. 2018. An adversarial approach for explainable ai in intrusion detection systems. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, Washington, DC, USA, 3237–3243.

[8] Shruti Patil, Vijayakumar Varadarajan, Siddiqui Mohd Mazhar, Abdulwodood Sahibzada, Nihal Ahmed, Onkar Sinha, Satish Kumar, Kailash Shaw, and Ketan Kotecha. 2022. Explainable Artificial Intelligence for Intrusion Detection System. *Electronics* 11, 19 (2022), 3079. https://doi.org/10.3390/electronics11193079

[9] Panagiotis Radoglou-Grammatikis, Konstantinos Rompolos, Panagiotis Sarigiannidis, Vasileios Argyriou, Thomas Lagkas, Antonios Sarigiannidis, Sotirios Goudos, and Shaohua Wan. 2022. Modeling, Detecting, and Mitigating Threats Against Industrial Healthcare Systems: A Combined Software Defined Networking and Reinforcement Learning Approach. *IEEE Transactions on Industrial Informatics* 18, 3 (March 2022), 2041–2052. https://doi.org/10.1109/TII.2021.3093905

[10] Panagiotis Radoglou-Grammatikis, Panagiotis Sarigiannidis, Georgios Efstathopoulos, Thomas Lagkas, George Fragulis, and Antonios Sarigiannidis. 2021. A self-learning approach for detecting intrusions in healthcare systems. In *ICC 2021-IEEE International Conference on Communications*. IEEE, Montreal, QC, Canada, 1–6.

[11] Maonan Wang, Kangfeng Zheng, Yanqing Yang, and Xiujuan Wang. 2020. An Explainable Machine Learning Framework for Intrusion Detection Systems. *IEEE Access* 8 (2020), 73127–73141. https://doi.org/10.1109/ACCESS.2020.2988359

[12] Tahmina Zebin, Shahadate Rezvy, and Yuan Luo. 2022. An Explainable AI-Based Intrusion Detection System for DNS Over HTTPS (DoH) Attacks. *IEEE Transactions on Information Forensics and Security* 17 (2022), 2339–2349. https://doi.org/10.1109/TIFS.2022.3183390