applied sciences

MDPI

Article

Post-Hoc Categorization Based on Explainable AI and Reinforcement Learning for Improved Intrusion Detection

Xavier Larriva-Novo *, Luis Pérez Miguel, Victor A. Villagra *, Manuel Álvarez-Campana, Carmen Sanchez-Zas and Óscar Jover

Departamento de Ingeniería en Sistemas Telemáticos (DIT), ETSI Telecomunicación, Universidad Politécnica de Madrid (UPM), Avda. Complutense 30, 28040 Madrid, Spain; luis.pmiguel@upm.es (L.P.M.); manuel.alvarez-campana@upm.es (M.Á.-C.); carmen.szas@upm.es (C.S.-Z.); oscar.jwalsh@upm.es (Ó.J.)
* Correspondence: xavier.larriva.novo@upm.es (X.L.-N.); victor.villagra@upm.es (V.A.V.)

Abstract: The massive usage of Internet services nowadays has led to a drastic increase in cyberattacks, including sophisticated techniques, so that Intrusion Detection Systems (IDSs) need to use AP technologies to enhance their effectiveness. However, this has resulted in a lack of interpretability and explainability from different applications that use AI predictions, making it hard to understand by cybersecurity operators why decisions were made. To address this, the concept of Explainable AI (XAI) has been introduced to make the AI's decisions more understandable at both global and local levels. This not only boosts confidence in the AI but also aids in identifying different attributes commonly used in cyberattacks for the exploitation of flaws or vulnerabilities. This study proposes two developments: first, the creation and evaluation of machine learning models for an IDS with the objective to use Reinforcement Learning (RL) to classify malicious network traffic, and second, the development of a methodology to extract multi-level explanations from the RL model to identify, detect, and understand how different attributes affect uncertain types of attack categories.

Keywords: Reinforcement Learning; IDS; UNSW-NB15; cybersecurity; XAI; SHAP

Citation: Larriva-Novo, X.; Pérez Miguel, L.; Villagra, V.A.; Álvarez-Campana, M.; Sanchez-Zas, C.; Jover, Ó. Post-Hoc Categorization Based on Explainable AI and Reinforcement Learning for Improved Intrusion Detection. *Appl. Sci.* **2024**, *14*, 11511. https://doi.org/10.3390/app142411511

Academic Editor: Douglas O'Shaughnessy

Received: 22 October 2024
Revised: 26 November 2024
Accepted: 28 November 2024
Published: 10 December 2024

1. Introduction

The objective of creating robust Intrusion Detection Systems (IDSs) has led these types of applications to adopt technologies such as Machine Learning (ML), often exploring the use of unconventional models for traffic classification tasks [1]. Those models, based on Deep Neural Networks (DNNs) [2], have become a popular and general solution due to their generally high performance and the possibilities for customization. However, these advantages make a model fairly complex, which has led to these models and their derivatives being categorized as "black boxes", since determining the relationship or logic between input data and results is not possible without further development [3].

There is currently a great interest in making these types of models more interpretable, by implementing techniques of what is called eXplanatory Artificial Intelligence (XAI) [4]. In the context of an IDS, the application of these types of techniques can be useful for a fast, effective and targeted response, as well as for the purposes of model tuning and correction.

Taking this into account, the problem that first arises in this research work is to explore how an unconventional model would be integrated as an Anomaly Intrusion Detection System (aIDS), based on attack categorization. A possible model is one based on Reinforcement Learning (RL) that, unlike a traditional supervised one, is based on modeling the environment with which an agent (the algorithm) interacts and learns, and applies changes following a reward system. Second, what XAI techniques can be applied in order to carry out a subsequent analysis of its operation, taking into account that this family of algorithms falls within the so-called "black box" category.

This research addresses the limitations posed by SHapley Additive exPlanations (SHAP) in Intrusion Detection Systems (IDSs) based on anomaly detection, specifically

overcoming the restrictions that SHAP imposes on deterministic models, as detailed in Section 2. Rather than providing a full framework implementation at this stage, this work sets potential research objectives, encouraging further exploration and development of the proposed IDS framework's individual components. The study focuses on IDSs driven by network traffic analysis, aiming to enhance anomaly detection in cybersecurity environments.

Furthermore, this research proposes using the benchmark dataset UNSW-NB15 [5] to test an algorithm capable of identifying different types of malicious traffic. The specific algorithm is a R.

Reinforcement Learning (RL) model utilizing Proximal Policy Optimization (PPO). The goal is to understand how the algorithm makes its decisions through the application of SHapley Additive exPlanations (SHAP) using a Python library of the same name. Section 4 further outlines the problem statement and proposed methodology, defining the ML algorithm and presenting the model evaluations to achieve a multiclass classification with SHAP explanations as the primary objective.

Additionally, this research emphasizes feature-level explanations, enabling IT administrators to identify relevant features within potential threats and gain clearer insights into the attacker's path. By interpreting which ranges of feature values might influence classifications, system administrators are better prepared to create specific filters or rules for advanced security tools, such as Extended Detection and Response (XDR) or Security Information and Event Management (SIEM) systems, facilitating a rapid, attribute-focused response to incidents.

This article introduces the background and related work to this research work in Sections 2 and 3. Section 4 explains the problem statement and the proposed methodology, defining the methodology ML algorithm presented in this paper. These sections also include the evaluation of the different models acquiring a multiclass classification with the main objective on SHAP explanations. Finally, Section 5 includes the discussion of the main conclusions and future lines for this research work.

2. Related Work

As far as we know, there are currently no related works involving the application of XAI techniques to explain the reasoning of an RL model of classification over IDS data. Regarding other models used to classify attacks from data captured by an IDS, there are several proposals such as the following: in [6], different models for intrusion detection in an Industrial Internet of Things (IIoT) environment are explored, including tree-based models such as Decision Tree (DT) and Random Forest (RF); and Deep Neural Networks (DNN), such as Multilayer Perceptron (MLP) along with a feature selection technique using XGBoost (XGB), another tree model. The study presented in [7] focuses similarly on tree models, adding the selection of features based on a genetic algorithm and achieving different results by focusing on different DNN applications, with a reduction of the classification to five classes of the possible ten classes (nine attacks, one normal traffic) considered on the UNSW-NB15 dataset.

Most of the proposals that make use of RL for an IDS use it for another purpose, such as the creation of honeypots for attack deflection [8], hyper-parameter control [9], or a multi-level multi-agent architecture [10]. The integration framework EI-XIDS [11] integrates a combination of diverse models and explainability methodologies, addressing the challenge of both the black-box nature of AI approaches to an IDS and the variance of interpretability objectives in different life-stages of an attack. It employs RL, but as a means of selection between different explainable models according to the detected scenario. We have not found any work that proposes or solves the application of explanations for an RL model in the context of threat detection.

Additional researches introduce the application of XAI in the context of IDS, aiming to explore the explainability of systems [4]. Further research presented in [12] shows an architecture structured around three core components: best-model selection, selected

model classification, and model explanation using the SHAP method. The best-model selection component focuses on identifying the most suitable machine learning models for the task. Several models are trained using IoT-specific IDS datasets, including IoTID20, NF-BoTIoT-v2, and NF-ToN-IoT-v2, which feature binary-class and multiclass outputs. Once the best-performing model is identified, it undergoes classification training using the previously mentioned datasets.

Finally, the selected model's classifications are evaluated and explained using the SHAP method. By understanding the rationale behind the model's decisions, stakeholders can gain deeper insights into potential security threats within the IoT environment.

The results presented in the paper demonstrate effective detection rates and accuracy across multiple datasets, enhancing the performance of existing methods. Notably, the use of ensemble tree models, particularly Decision Trees (DT) and Random Forests (RF), achieves 100% accuracy and Area Under the Curve (AUC) in several scenarios.

The SPIP (S: Shapley Additive exPlanations, P: Permutation Feature Importance, I: Individual Conditional Expectation, P: Partial Dependence Plot) framework, proposed by Keshk et al. [13], aims to assess explainable Deep Learning algorithms for IDSs in IoT environments. This framework generates both local and global explanations by combining various XAI and DL methods. SPIP extracts a tailored set of input features that outperform the original feature set across three of the most widely used benchmark datasets: NSL-KDD, UNSW-NB15, and ToN_IoT. First, binary classifiers are constructed to detect attacks without specifically identifying the class of the attacks. XAI methods (SHAP and PDP) are applied to analyze the most critical features across three datasets: NSL-KDD, UNSW-NB15, and ToN_IoT. Next, one-vs-all classifiers are built to detect each class of attack separately. This approach allows for the creation of dedicated models for individual attack classes. Once the results from the fitted one-vs-all classifiers are obtained, the proposed framework generates both local and global explanations for the detection of specific attack classes. Lastly, a target-vs-normal classifier is developed to classify between a targeted attack class and normal traffic. The research work published in this article confirms a strong correlation between input features and attack classes, demonstrating the validity of classifier predictions.

Barnard et al. [14] proposes a framework comprised of a two-stage pipeline for network intrusion detection. In the first stage, data for the NSL-KDD dataset are collected and processed into a feature vector. Next, an XGBoost model is trained for binary classification, distinguishing between normal and malicious traffic. This stage serves as the basis for preliminary intrusion detection. Next, the SHAP framework is employed to provide explanations for the model's decisions, focusing on the contribution of each feature to the classification outcome. In the second stage of the pipeline, the explanations generated in the previous stage are utilized to train an auto-encoder, forming the basis of an anomaly-based Network Intrusion Detection System (NIDS). This auto-encoder consists of two key components: a DNN encoder and a DNN decoder. The encoder compresses the previously generated explanations into a lower-dimensional representation, while the decoder reconstructs the original input from this compressed representation. By analyzing the reconstruction error of the auto-encoder, changes in behavior patterns can be identified, which are indicative of anomalous behavior. The experiments conducted on the NSL-KDD dataset demonstrate the framework's capability to detect new attacks. Moreover, the framework's overall performance is found to be comparable to numerous state-of-the-art research works in cybersecurity.

Hariharan et al. [15] focus on developing a classifier for network intrusion that combines high prediction with an explainer to interpret the classifier's predictions. They evaluate global XAI methods such as Permutation Importance (PI) and SHAP, along with local XAI methods like SHAP, LIME (Local Interpretable Model-agnostic Explanations), and CIU (Contextual Importance Utility), to interpret predictions made by Random Forest (RF), XGBoost, and LightGBM (LGBoost) models on both the Kaggle IDS and NSL-KDD datasets. The research mentioned above evaluates the robustness of explanation methods in terms of accuracy, recall, and precision, particularly focusing on the top 15 features

identified by PI and SHAP on the NSL-KDD dataset. The obtained results achieve values of accuracy up to 70% to 75% in model predictions based on these features. Local explanation methods were also examined for their consistency and stability.

The proposal in [16] aims to increase the explainability of decisions made by DL Intrusion Detection Systems for IoT networks. The framework consists of three main components: a DNN-based IDS specifically designed for IoT networks, built using deep neural networks (DNNs); and three XAI techniques: RuleFit, LIME, and SHAP, generating local, global, and feature importance-based explanations, respectively. In this work, two public network security datasets are considered: NSL-KDD and UNSW-NB15. The results from the experiment show that the proposed XAI framework achieves an accuracy of 88% and a precision of 96% on both datasets. The experimental results confirm that the XAI framework outperforms most state-of-the-art works in terms of accuracy and F1-score on both datasets.

The work by Mahbooba et al. [17] moves away from post hoc explainable methods like SHAP and LIME, creating explanations based on entropy-based measures like information gain for selecting those features important in the prediction process. Additionally, the inheritance structure of a DT allows the easy creation of decision rules over those same features, which easily provide a useful interpretation of the behavior of the trained model. In Table 1, our work is compared with the main contributions of some of the related studies mentioned above.

Table 1. Comparison of related works and explainable intrusion detection techniques. F1-score added when same dataset is used.

Reference	Year	Datasets	XAI Technique	XAI Scope	XAI Use	Accuracy
[12]	2022	IoTID20, NF-BoTIoT-v2, NF-ToN-IoT-v2	SHAP	Local, Global	Interpretability	-
[13]	2023	NSL-KDD, UNSW-NB15, ToN_IoT	SHAP, PFI, ICE, PDP	Local, Global	Interpretability	0.864
[14]	2022	NSL-KDD	SHAP	Global	Improving model performance	-
[15]	2023	NSL-KDD	PFI, SHAP, Lime, CIU	Local, Global	XAI techniques comparison	-
[16]	2022	NSL-KDD, UNSW-NB15	SHAP, RuleFit, LIME	Local, Global	Improve IDS trust	0.99
[17]	2021	KDDcup'99	Information Gain, Decision Trees	Local, Global	Enhance trust in IDS	
Proposed solution	2024	UNSW-NB15	SHAP	Local, Global, Ranged Feature relevance	Improve IDS performance, propose semi-automatic rules	0.70

In a recent study [3], the applications of XAI methods to IDS were classified into black- and white-box methods. White-box models are designed with explainability in mind and do not require an additional layer of analysis to offer an interpretation of the results, such as decision trees or clustering algorithms. However, this intrinsic explainability comes at the cost of higher complexity in the design, and in general, less accuracy [18]. Black-box methods are nontransparent in their decision but post hoc explainable models allow for an explanation of what pushed the model to classify a sample.

On a similar note to our proposal, the XAI-IDS framework defines the advantages that an integration of XAI techniques into the detection models could bring to Network IDS [19].

They propose a benchmark of seven black-box models, not including RL. Alongside the benchmark, a comprehensive list of both local and global explanations are defined to be included in an integrated framework.

Our work follows the trend of these state-of-the-art XAI methodologies, addressing the implications and design necessary for the use of an RL model, an architecture that has been proposed as useful on explainable IDSs [11], but one where SHAP, or other XAI approaches, have never been directly applied.

Furthermore, we propose another level of explanation based on feature value ranges, building on the idea of inferring specific decisions made by a model over a single feature [16,17], where the common individual and global explanations are presented. These feature-range explanations allow for the proposed creation of fixed detection rules, something akin to the rules that can be discerned from the inherent structure of a DT model [17]. This could help to bring the insight allowed by these prediction models into the security configuration of a system.

3. Dataset

The NSL-KDD dataset, a widely used and benchmark dataset for the solutions mentioned above, was proposed as an updated version of KDDcup'99, aiming to solve the redundancy and unbalanced issues of the later one, which was shown to highly affect the performance of detection systems [20]. However, both of these datasets are over a decade old and lack representation of recent and more complex attacks, especially in comparison with the UNSW-NB15 dataset [21]. This dataset is a comprehensive and fairly recent dataset created by the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) [5]. The same organization has published newer datasets such as the ToN-IoT [22], and the newer versions NF-BoTIoT-v2 and NF-ToN-IoT-v2. However, the network activities recorded on those sets are specific to the domain of IoT.

In its original format, UNSW-NB15 collects around 2.5 million network registries containing 49 features categorized into five groups: flow features, basic features, content features, time features, and additional generated features. Each registry is classified as either a common or an attack package, this being further categorized into one of nine possible attack groups: analysis, backdoor, DoS, exploits, fuzzers, generic, reconnaissance, shellcode, and worms. A reduced format collection of 175,341 training and 82,332 test entries is available, which reduces the recorded features to those related to IP address and source/destination ports. This is the set that has been used in this work following the instructions in [2].

4. Proposal

This research article proposes a simple architecture for the computation of explicability in two levels: by the global feature importance, and by the ranges of their values. The objective is allowing the explanation of the decision-making process of a classification model so that the protection of specially valuable systems can be kept on a human-in-the-loop approach. The use of "black-box" models on IDSs can bring a reliable level of protection but can as well leave an organization completely in the blind towards the actors behind an intrusion and their most vulnerable points of entry. In currently deployed systems, these models have been previously studied and evaluated, but this provides little insight into their inner work, just integrating the attack predictions into the system. In our proposed architecture, both global and local explanations of the model behavior could be integrated for a better performance and a rule-based configuration of the system by the administrator user.

The integration of explanations and interpretation on the detection performed by an IDS can allow a greater level of interaction between the predictions of the AI models used by that system, and the configuration and response performed by the operators of the network. After an initial training of a classification model over a known dataset, global explanations and the derived class feature importance can be reused to further fine-tune a

model, and the explanations in a by-the-feature method can be used for the development of automatic filtering to be applied to the network IDS as rule-based controls. An abstract proposal for such an application is presented in Figure 1.

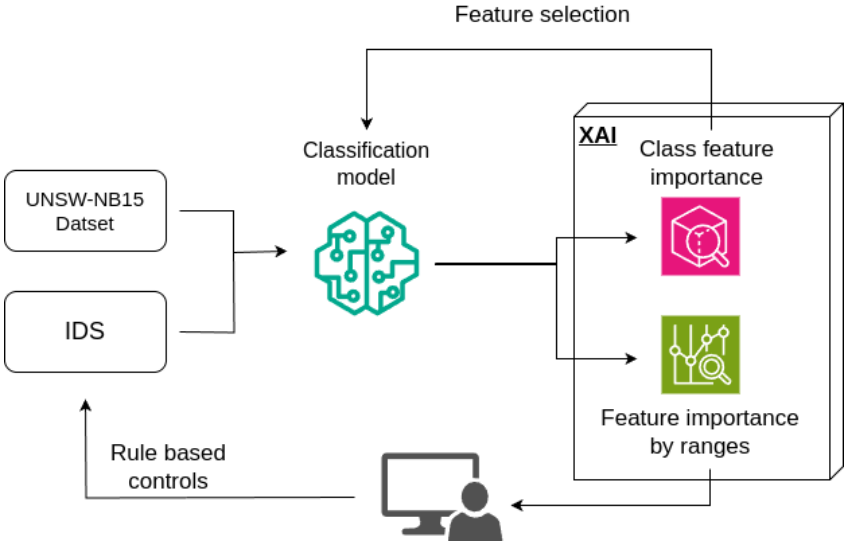

Figure 1. Proposed architecture for the use of global and feature-specific explanations.

This work was developed using the UNSW-NB15 dataset to evaluate this system for the classification and further explanation of malicious traffic. In detail, this research article evaluates a reinforced learning algorithm for multiclass classification based on Proximal Policy Optimization (PPO), using SHapley Additive exPlanations (SHAP) as an XAI technique for the post hoc explanation of the model.

The choice of both the model and the XAI technique is not arbitrary. Based on cooperative game theory, SHAP values provide a mathematically rigorous method for assigning an output contribution to each input feature in a model, allowing both a local explanation (of a single prediction) and aglobal understanding (of model capabilities) [23]. As has been presented, this has led SHAP to be the standard de facto method used for AI explanation. However, choosing SHAP has limited the choice of the RL algorithm, since the SHAP library explainer to be used, KernelExplainer, requires a prediction function that corresponds to a probability distribution. Considering that some of the RL algorithms have a fully deterministic prediction function, this has limited the choice to two within the chosen framework: PPO or A2C, from which the first has finally been chosen.

The development presented in this work follows the following steps:

- Dataset Preparation: Establishment of a criterion for selecting dataset characteristics and processing of the dataset.
- Modeling and Training: Designing of the environment according to the RL philosophy, and machine learning algorithm-oriented anomaly Intrusion Detection Systems.
- Modeling Comparison: Developing a model performance comparison.
- Application of Reinforcement XAI Model: Exploration and application of XAI techniques to the RL model with the best F1-score metric.

4.1. Dataset Preparation

For this work, the dataset UNSW-NB15 was selected using as a target the label for binary classification between intrusion or common traffic, since the aim is to classify only the malicious traffic. Also in this research work, all records with a normal value in attack_cat were discarded in the multiclass target column.

Previous to the evaluation of a set of models over this task, three different versions of the dataset used were created, based on three levels of feature selection. We have used the metric Mean Decrease in Impurity (MDI), which arises from the extension of the concept of

Gini impurity from decision trees in Random Forest models. The perceived importance of each characteristic was obtained by training 100 iterations of a Random Forest model, using the default Scikit-learn implementation. To ensure the accuracy in the MDI (Mean Decrease in Impurity) values, the average feature importance was calculated across all iterations.

Over this, three different threshold values have been applied: 0.03, 0.02, and 0.01. Selecting only those features scoring above the threshold on MDI values to be considered when training and testing the model, three subsets were constructed to be used as datasets. In Table 2, the features selected for each subset are included. In this table, all features on the subset with a threshold of 0.03 are also included on the subsequent sets, and their MDI values will also be over the lower thresholds.

Due to the UNSW-NB15 dataset having a fairly unbalanced class layout, an additional copy of each dataset was created using the oversampling technique SMOTE [24], leading to a total of six sets to evaluate models on. For each of these, five instances of a 75/25 split were created in training and testing, to ensure no variance between results. Moreover, MinMaxScaler was applied to resize the numerical features into a $[-1, 1]$ range and apply one-hot-encoding over the nominal ones.

Table 2. Description of features included in each dataset. Each set also considers the features of the higher thresholds.

Threshold	Total Number of Features	Selected Features
0.03	10	sbytes, service, smean, ct_srv_dst, ct_srv_src, ct_src_dport_ltm, ct_dst_src_ltm, ct_dst_sport_ltm, ct_dst_ltm, dbytes
0.02	14	sload, proto, sttl, dmean
0.01	28	ct_src_ltm, rate, dur, sinpkt, sjit, dload, dloss, spkts, dinpkt, synack, dpkts, djit, sloss, tcprtt

4.2. Modeling and Training

This section explores the design and training of different machine learning models oriented toward Anomaly Intrusion Detection Systems. selecting four different versions of the observed models in the related works: Decision Trees (DT), random Forest (RF), XGBoost (XGB), and Multi-layer Perceptron (MLP); as well as the proposed Reinforcement Learning model. For each architecture, six models have been trained over each of the reduced feature datasets presented above both with and without balancing.

Public implementations have been used when available, opting for the default configuration varying instead the dataset preparation as stated above. The Scikit-learn implementations were used for the first two, the XGBoost library was used for the algorithm with the same name, and Keras was used for MLP.

For MLP, a network architecture of two dense layers of 64 units was used, which is the default configuration of the neural network used in the model. Furthermore, EarlyStopping was applied with the following hyper-parameters: monitor = 'val_loss', patience = 10, and restore_best_weights = True, ensuring that the model would stop training if the validation loss did not improve for 10 consecutive epochs and the best-performing model was retained. While the training was set to a maximum of 100 epochs, the model typically executed around 50 to 60 epochs before early stopping was triggered

Reinforcement Learning Model

Due to the philosophy behind reinforcement learning models, it is necessary to first define the environment in which the agent was trained. The configuration of the observation space and action space was considered as follows:

- **Observation Space** (s): of class `Box` with shape (x), where x is the number of features of the corresponding training partition. In this way, the agent only observes one record

at a time. It is also defined that the values provided are of type `float` within the limits $[-1, 1]$.

- **Action Space** (a): of class `Discrete` with nine possible values, each corresponding to a type of attack.

For the design of the environment logic, it has been necessary to consider the following points:

- An RL algorithm essentially seeks the solution to a problem modeled, where a state transition depends on the current state and the action taken [1].
- A subset of records from the original dataset is used, where none of the elements have a temporal relationship with the previous one.
- The choice of an action (classification of a record) has no effect on the next observation or its features.

With this in mind, the environment $R(\tau)$ is designed such that an episode (τ) consists of only one state–action pair, so that:

$$\tau = (s_0, a_0)$$
$$R(\tau) = r_{t=0}$$

Otherwise, the RL algorithm would attempt to learn correspondences in transitions from one record to another, slowing down-training or, in the worst case, reducing the model performance by adding a component that could be considered noise.

The choice of algorithm was limited by three factors:

1. Availability in Stable-Baselines3: The library includes six algorithms (A2C, DDPG, DQN, PPO, SAC, and TD3).
2. Compatibility with defined observation and action spaces: Given the limitations of discrete action spaces and `Box` observation spaces, valid algorithms are reduced to A2C, DQN, and PPO.
3. Valid policy for XAI techniques: As discussed, algorithms in the Q-learning family, like DQN, use a deterministic policy, which is incompatible with the `KernelExplainer` that requires stochastic model outputs. This limits the choice to PPO or A2C.

To accelerate training, the vectorized environment functionality of Stable-Baselines3 [25] has been used, allowing the parallelization of independent environments, enabling the training of an agent in n environments per step. This significantly speeds up training, with the drawback of increased computational cost. A total of eight parallel environments were created, which nearly reaches the virtual machine's RAM limit (30 GB) for the highest combination of features and records, with a threshold of 0.01 using SMOTE.

Additionally, eight parallel environments have been created, which nearly reaches the RAM limit (30 GB) of the virtual machine, based on the highest feature and record combination using a threshold of 0.01 and SMOTE. This is the maximum allowed before encountering memory constraints.

Bearing in mind the possibility of inserting noise during training due to the fact that RL algorithms seek to establish relationships between the different states (in our case the states are the network data) and that the used registers do not follow a temporal relationship, the following environment has been designed, where each episode is only composed of a state–action or step pair. This behavior is presented in Algorithm 1.

4.3. Model Comparison

4.3.1. Reinforcement Learning

The results obtained with RL are the lowest of the algorithms studied. Furthermore, there is an increase in precision and recall when no data-balancing techniques are applied, while the accuracy and F1-score metrics decrease. When analyzing the results based on the threshold used, an increase in performance is observed as the threshold is reduced.

4.3.2. Decision Tree

The results obtained by the Decision Tree model indicate that the accuracy and F1-score values exhibit minimal variation when balancing techniques such as SMOTE are applied. However, a noticeable increase in accuracy is observed when dataset analysis techniques, such as feature encoding and selection, are implemented. Further analysis of the algorithm's performance as a function of the selected threshold shows that recall improves with higher thresholds, which corresponds to a greater number of features being included in the model. This suggests that as more features are considered, the model becomes better at identifying positive instances, particularly in imbalanced datasets.

Algorithm 1 Reinforcement learning environment and algorithm

INITIALIZATION
$x\_train, y\_train \leftarrow$ Training data
$Index \leftarrow$ random number $[0, \text{len}(x\_train)]$
NEXT_STATE $\leftarrow$ Next state function
ACTION_SPACE $\leftarrow$ Set of possible actions
return Current state
$STEP \leftarrow$ An agent step corresponds to an episode
$end\_episode \leftarrow$ False
Take action a from ACTION_SPACE in state s
Observe reward r and next state s'
if Episode termination condition is met **then**
    $end\_episode \leftarrow$ True
end if
if End of training data **then**
    Reset index to 0
    return Observation, Reward, end_episode
end if
REBOOT {Runs during initialization and when *end_episode* is False}
return Next state

4.3.3. Random Forest

Analyzing the performance of the Random Forest algorithm, the accuracy values remain stable; however, a marked decrease in accuracy is observed when data balancing techniques, such as SMOTE, are applied. Similar to the behavior seen in Decision Tree algorithms, recall values improve when data balancing is used, highlighting the model's increased ability to identify minority class instances. It is also noteworthy that the F1-score metrics remain consistent across different variations of the Random Forest algorithm, regardless of the balancing technique applied. When comparing results based on the set threshold, both the recall and F1-score values tend to increase as the threshold rises, which implies that including a greater number of features enhances the model's ability to capture positive instances while maintaining precision.

4.3.4. XGBoost

When comparing the results of the XGBoost algorithm with the previous models, a general increase in both the accuracy and precision metrics is observed across the different combinations. The results follow a similar pattern to the other models, where higher recall and lower precision are evident when data balancing techniques, such as SMOTE, are applied. Furthermore, the accuracy and F1-score metrics remain relatively consistent across all tests, with only slight decreases in accuracy when SMOTE techniques are not used. This suggests that while XGBoost benefits from data balancing for recall, it maintains strong performance across a range of metrics, showing robustness in various configurations.

4.3.5. MLP

The MLP results are lower than those obtained with the tree-based algorithms. Regarding the application of balancing techniques, the MLP model presents results that maintain the trend of previous models. If we analyze the results according to the established thresholds, we can see a notable decrease in accuracy, precision, recall, and F1-score for high-threshold values. For lower-threshold values, the results stabilize, except for precision, where a significant difference is observed between the intermediate threshold and the lower threshold.

4.3.6. Best Results

Table 3 introduces the results of the best-performing model for each of the training dataset variants in each of the model algorithms used, using F1-score as a the evaluation comparison. The F1-score, accuracy, and recall metrics use a macro average to give equal weight to all classes. The standard deviations of all models have been included, although the iterations over five different random selections of training data give almost no variance to the results.

Table 3. Performance comparison of different models. Format: mean ± std.

Model	Balance	Threshold	Accuracy	Precision	Recall	F1
DT	No	0.03	0.80 ± 0.01	0.69 ± 0.00	0.57 ± 0.00	0.80 ± 0.00
RF	No	0.03	0.81 ± 0.00	0.73 ± 0.01	0.57 ± 0.00	0.60 ± 0.01
XGB	No	0.03	0.82 ± 0.00	0.80 ± 0.01	0.60 ± 0.01	0.62 ± 0.00
MLP	Yes	0.01	0.72 ± 0.01	0.49 ± 0.01	0.61 ± 0.01	0.49 ± 0.01
RL	Yes	0.01	0.70 ± 0.01	0.48 ± 0.01	0.60 ± 0.01	0.47 ± 0.01

There is a clear difference in performance in favor of classical machine learning models such as DT and RF in terms of accuracy; instead MLP and RL require more resources to maximize their metrics.

5. Application of Reinforcement XAI Model

Over the best-performing RL model, according to the criteria of the previous threshold, a methodology for interpretability was applied. The models trained on a selection of features with an MDI over the threshold of 0.01, and balanced with SMOTE, were used. The SHAP methodology and framework were employed, creating a KernelExplainer with the RL model's action policy and a subset of 100 synthetic summary records constructed via K-means clustering.

The individual SHAP values of a model prediction help us understand how specific feature values influenced the model's decision to classify an instance into a particular class. From these values, we can extract two key insights. First, the absolute value of a SHAP score indicates the importance of the feature value in the final prediction. Second, the sign of the SHAP value reveals whether the feature contributed positively or negatively to the classification decision. By examining the SHAP values of a specific prediction, we can identify which features were critical indicators of an intrusion. In Figure 2, the features that contributed to classifying a sample as Reconnaissance are illustrated. These features, aggregated from their one-hot-encoding into nominal values, include only the top 10 based on their overall absolute importance.

These explanations bring insight into the performance of one single prediction, but can be highly dependent on the specific values of the selected sample, and thus have little use in bringing useful replicable knowledge for future instances of a same class. It is of a higher interest to compute a global class specific to feature importance, including both the absolute and direction value of how a feature is generally considered by a prediction model.

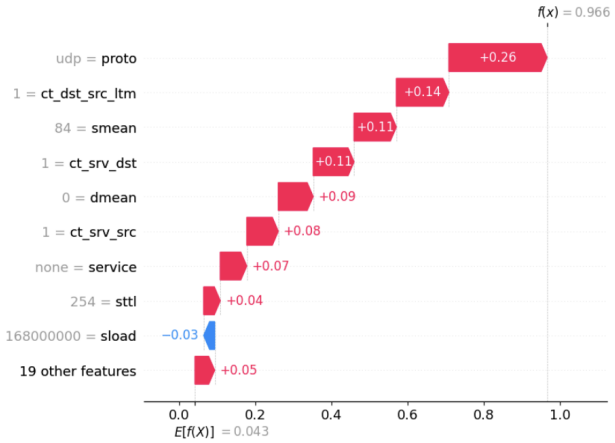

Figure 2. Individual explanation for the prediction of a single sample of the "Reconnaissance" class.

5.1. Class-Specific Feature Importance

The first step for a global explanation involves using SHAP values to compute the relative importance of the features with respect to each of the attack classes, enabling the creation of class-specific feature explanations, which can be particularly useful in the context of an IDS (Intrusion Detection System). To generate these class-level explanations, SHAP values are computed over a large number of random samples from the testing partition, totaling 5000 samples. By analyzing the distribution of SHAP values for a specific class, the most relevant features for recognizing each type of intrusion can be interpreted.

A high positive SHAP value for a feature indicates a strong contribution towards predicting a particular class, while a negative SHAP value signifies a decreasing influence on that prediction. Figure 3 shows a summary of the importance of a subset of features on the classification of the Analysis class. The selected subset corresponds to the ten features with the highest sum of absolute SHAP values.

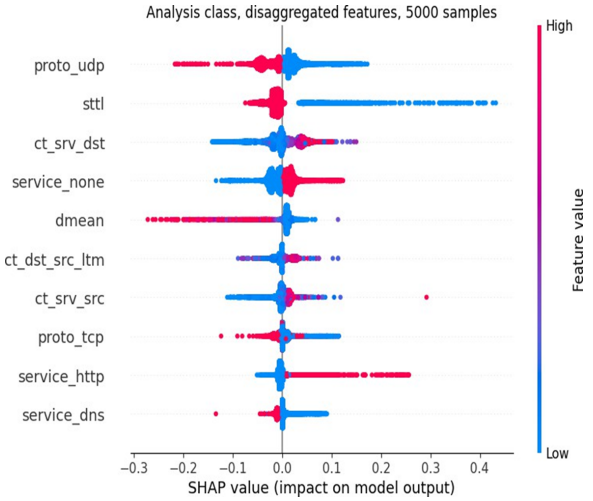

Figure 3. Feature importance interpretation for "Analysis" class.

In this figure, color is used to indicate the value of a feature relative to its own typical ranges. It must be considered that some of these features have been separated by one-hot-encoding, meaning that for proto, service and state, a high value of proto_udp indicates a positive (1) instance of the UDP protocol being used. This image allows us to easily draw some conclusions for the Analysis class, it can be inferred that UDP messages with low values of sttl (source to destination time to live) are indicative of a malicious behavior of this class.

In Figure 4, the same explanation visualization has been included for the rest of the classes. This figure can allow a similar inference over each of the attack classifications included on the dataset UNSW-NB15.

The development of rule-based controls from these global explanations require further individual studies applied over each class, either manually, or by inferring the feature importance by value range following the method described in the next section. However, some evidence can be found just by observing this overview. The selection of protocol seems to be enough of a differentiator to detect DoS, backdoor, fuzzers, and shellcode attacks. Additionally, a low value of sttl, the source to the destination time to live, is indicative of an exploit, while a low mean packet size from the source indicates a possible reconnaissance packet.

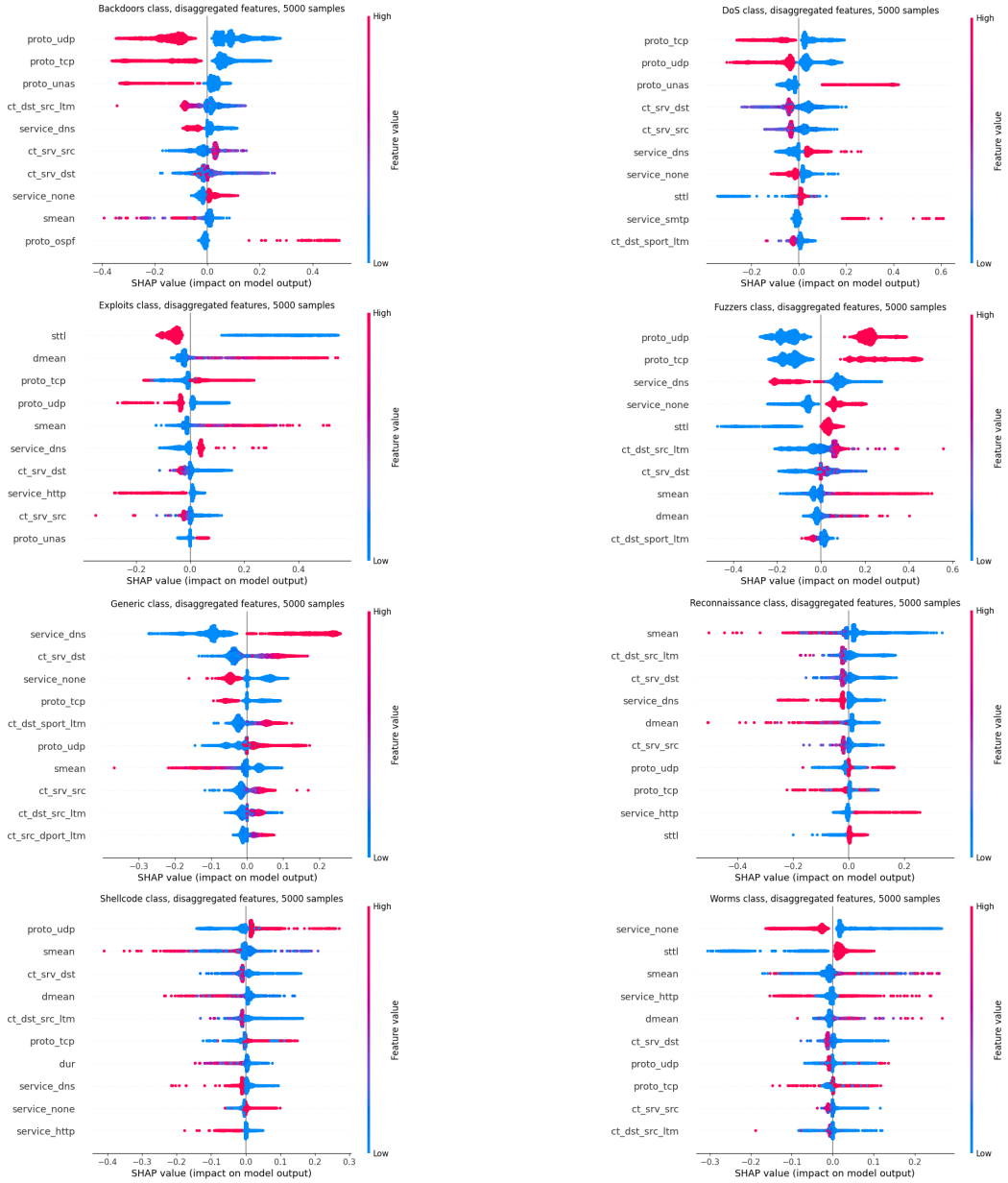

Figure 4. Feature importance interpretation with SHAP values for top 10 relevant features per each class.

5.2. Feature Importance by Value Ranges

These values alone might give information about the importance of a feature in comparison with the rest. However, with the exception of the nominal one-hot-encoded features, the "high" or "low" levels of numerical features might prove less intuitive. In

the context of an IDS, an expert would require greater precision to be able to make use of this explanation. The following Algorithm 2 computes the mean SHAP value for a feature divided by ranges, which can impact a specific range of values for a numerical feature has on a classification. This method uses the previous iterations over 5000 random samples to divide each feature's possible values on the desired number of intervals, computing the average SHAP value of all samples whose feature value lies inside said intervals.

Algorithm 2 Obtaining mean SHAP values based on ranges of a feature for a class

$f \leftarrow$ Characteristic to be evaluated
$c \leftarrow$ Class to be evaluated
$SHAP\_values \leftarrow$ SHAP values resulting from the samples used
$n_i \leftarrow$ Number of intervals into which to divide f values based on the training dataset
$i_p \leftarrow$ An n_i-length array with the values of f after processing (MinMax) corresponding to the limits of the intervals
$i_o \leftarrow$ Array of n_i length with original f values corresponding to the limits of the intervals
$f\_values \leftarrow$ Values of f in samples used for calculating SHAP values, i.e., after processing
$Empty \leftarrow$ Empty Array
$x\_ticks \leftarrow$ Empty Array
for i in $(n_i - 1)$ **do**
    A mask is created according to $i_p[i] \leq f\_values < i_p[i+1]$. If it's the last iteration, the upper limit is included
    Add to x_ticks the evaluated interval with the original values in String format: "$\geq i_o[i]\& < i_o[i+1]$". If it's the last iteration, the upper limit is included
    The mask is applied to the $SHAP\_values$ of class c, obtaining the SHAP values of all the characteristics of the samples that comply with the mask. The values corresponding to the Objective Feature f.
    The average of the values obtained is calculated and added to the array res
end for
return res, x_ticks

After applying this methodology to the numerical values used in the explanation for the Analysis class with four intervals per feature, Figure 5 illustrates the effectiveness of this algorithm when applied to the feature Ct, srv, dst.

This feature quantifies the number of connections sharing the same service and destination address within the last 100 recorded connections.

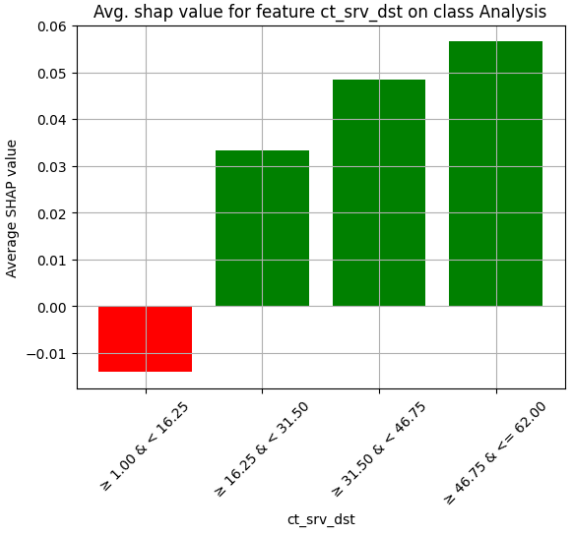

Figure 5. Feature relevance by ranges as shown by its average SHAP value for the "Analysis" class.

Ranges whose mean SHAP value is positive can be observed, showing relevance for a positive model classification. Over 15 close-by connections with one identical address for the service and destination might be indicative of an Analysis attack. This information might prove useful for the construction of specific controls and security measures disallowing connections when the proper flags are raised. A list showing the specific values of relevance for the top features as shown in Figure 3 is included in Table 4. This shows the values our trained model inferred as indicators of an attack of this class.

Table 4. Analysis attributes.

Feature	Type	Values
Proto	Nominal	From UDP / TCP
Sttl	Numerical	<63.75
Ct_srv_dst	Numerical	≥16.25
Service	Nominal	Yes None/HTTP No DNS
Dmean	Numerical	<362.75
Ct_dst_src_ltm	Numerical	≥17.00 & <49.00
Ct_srv_src	Numerical	≥16.50

It is necessary to mention that the accuracy in numerical values is subject to the number of intervals used and, in addition, not in all cases the values for the nominal characteristics are as easily discernible as in this example. Finally, a global explanation of the model is shown in Figure 6, corresponding to the impact of characteristics across all classes, where the dominance of nominal characteristics is appreciable.

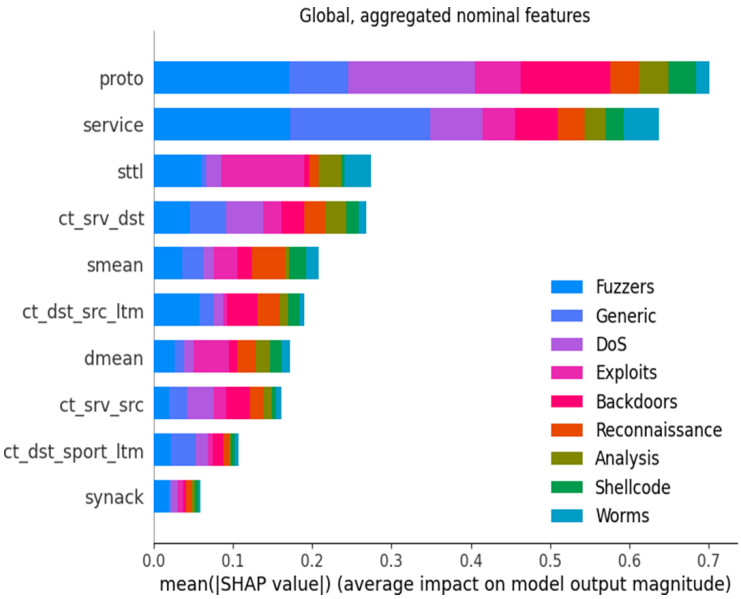

Figure 6. Global explanation with the 10 characteristics with the most impact on the model. Nominal characteristics aggregated.

6. Conclusions, Discussion, and Future Work

Given the methodology used in this work, it can be determined that the use of RL as a classifier is not optimal. Compared to other solutions, it has a clearly lower performance. In this regard, the model with the best metrics and the one that is recommended is XGBoost. However, it is not ruled out that a more in-depth study of the design of the environment, the optimization of parameters, the evaluation of other algorithms, or the application of explanations for performance improvement (as was presented in some related works) may result in a model with superior results.

Regarding XAI, it has been seen that the use of the default SHAP is useful to give basic notions about the behavior of the model, but if more precise information is required, such as in the context of an IDS, additional functionalities like the ones proposed here need to be developed. Even so, the proposed methodology has its limitations, namely mainly the absence of a unified process for numeric and nominal columns when constructing the ranges. It is hoped that in the future this limitation can be alleviated by automating the distinction between column types and applying the methodology to all characteristics (in this work it has only been applied to those present in the graphical explanation), in addition to possible improvements, such as a process of searching for valid ranges in the case where they are not found in the first place. Concerning other works, the main use of XAI has been focused on its application within the context where the model is deployed, rather than improving its performance.

The explainable model is the key component of this proposal in order to obtain the evaluation of the attributes by class analysis in order to validate the detection of possible attacks. The validation and interpretation that this model offers is fundamental for a proper use of automated detection on intrusion detection systems, even more so if a dynamic integration of possible controls and courses of action is to be considered in order to defend a system against the detected intrusion. This component can be updated and improved the most using real tools such as systems like Zeek (formerly Bro-IDS), which can obtain real-world information from a network point by classifying the various characteristics necessary for the system's operation.

Furthermore, the explainable model can be explored using algorithms and modules based on SPARK [4] technology for data streaming to provide real-time explanations calculating the SHAP values based on Spark script, thus improving the performance of the system. Finally, to improve the system, two options are worth considering: (1) implementing an anomaly alarm system with real-time user notifications; (2) focusing on the existing machine learning models by improving the data-cleaning process. This data cleaning would specifically target boosting the True Positive Rate (TPR) metric in multiclass classification, ultimately reducing false alarms.

Additionally, training these models with a combination of diverse datasets would enhance their ability to identify a wider range of attacks and anomalous traffic, leading to a more robust and secure Intrusion Detection System (IDS).

Moreover, this same explainability method could be applied to further classifications over malicious data. Models trained to recognize intrusion on a system from their logs, like those used on SIEMs, could also benefit from further explaining the indicators that pushed a model to classify some behavior as critical.

Further research could explore the unique challenges of applying XAI techniques in domains such as the Internet of Things (IoT) [26], where explanations need to account for and adapt to highly heterogeneous environments. Similarly, in Intelligent Connected Vehicles (ITS) [27], the focus could be on tailoring explanations to end users who are not technical administrators but are still interested in understanding potential attacks [28].

Similarly, explainability could be applied to models of app fingerprinting, where the analysis of network traffic is used—often without the user's knowledge—to identify the apps and functionalities being utilized on a device [29,30]. Another potential application lies in optimization schemes for more secure communication, particularly in IIoT environments [31]. Both areas are closely related to cybersecurity but fall outside the scope of intrusion detection.

The decisions made in these realms of cyber-defense systems need to be both immediate and verifiable. A system capable of conclusively explaining the reasons behind a model's decision could lead the admin in charge of a system to perform a faster recovery and implement a better prevention to further repeated intrusions. This research work represents a previous version of a future system that should be able to identify anomalous behavior on network traffic developed by potential insider attacks, where the output value will be part of the situational awareness of a modeling risk scenario, considering the system presented in [32].

As discussed in Section 2, different explainable techniques have been employed in Intrusion Detection Systems (IDS) to enhance their interpretability and performance. Table 1 compares the proposed approach with related works, focusing on key aspects such as the datasets used, whether the data were balanced, the XAI (Explainable Artificial Intelligence) techniques employed, the scope of explanation (local or global), and the primary use of these techniques. The comparison highlights the different approaches used for explainability, including SHAP, PFI, ICE, and LIME, and illustrates how the proposed model offers an advancement by incorporating SMOTE balancing and the introduction of semi-automatic rule proposals for the IDS improvement.

Author Contributions: Conceptualization, X.L.-N. and L.P.M.; methodology, X.L.-N. and L.P.M.; software, X.L.-N. and L.P.M.; validation, X.L.-N., L.P.M. and C.S.-Z.; formal analysis, X.L.-N.; investigation, L.P.M.; resources, X.L.-N., V.A.V. and M.Á.-C.; data curation, L.P.M.; writing—original draft preparation, X.L.-N. and L.P.M.; writing—review and editing, C.S.-Z., Ó.J., V.A.V., M.Á.-C., X.L.-N. and L.P.M.; visualization, X.L.-N. and L.P.M.; supervision, X.L.-N. and M.Á.-C.; project administration, M.Á.-C. and V.A.V.; funding acquisition, X.L.-N. and M.Á.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been partially by the National Institute of Cybersecurity from Spain (INCIBE), through the national promotional program "Proyectos Estratégicos de Ciberseguridad en España", which falls under the Recovery Plan, Transformation and Resilience campaign, funded with Next Generation EU funds.

Data Availability Statement: All tex data is available from the references mentioned in the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yuan, S.; Wu, X. Deep learning for insider threat detection: Review, challenges and opportunities. *Comput. Secur.* **2021**, *104*, 102221. [CrossRef]
2. Larriva-Novo, X.A.; Vega-Barbas, M.; Villagrá, V.A.; Rodrigo, M.S. Evaluation of cybersecurity data set characteristics for their applicability to neural networks algorithms detecting cybersecurity anomalies. *IEEE Access* **2020**, *8*, 9005–9014. [CrossRef]
3. Neupane, S.; Ables, J.; Anderson, W.; Mittal, S.; Rahimi, S.; Banicescu, I.; Seale, M. Explainable Intrusion Detection Systems (X-IDS): A Survey of Current Methods, Challenges, and Opportunities. *IEEE Access* **2022**, *10*, 112392–112415. [CrossRef]
4. Larriva-Novo, X.; Sánchez-Zas, C.; Villagrá, V.A.; Marín-Lopez, A.; Berrocal, J. Leveraging Explainable Artificial Intelligence in Real-Time Cyberattack Identification: Intrusion Detection System Approach. *Appl. Sci.* **2023**, *13*, 8587. [CrossRef]
5. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, 10–12 November 2015; pp. 1–6. [CrossRef]
6. Kasongo, S.M. An advanced intrusion detection system for IIoT based on GA and tree based algorithms. *IEEE Access* **2021**, *9*, 113199–113212. [CrossRef]
7. Rathod, N.A.; Gupta, T.; Sharma, N.V.; Sharma, S. Model Comparison and Multiclass Implementation Analysis on the UNSW NB15 Dataset. In Proceedings of the 2021 International Conference on Computational Performance Evaluation (ComPE), Shillong, India, 1–3 December 2021; pp. 549–555. [CrossRef]
8. Veluchamy, S.; Kathavarayan, R.S. Deep reinforcement learning for building honeypots against runtime DoS attack. *Int. J. Intell. Syst.* **2022**, *37*, 3981–4007. [CrossRef]
9. Han, H.; Kim, H.; Kim, Y. An efficient hyperparameter control method for a network intrusion detection system based on proximal policy optimization. *Symmetry* **2022**, *14*, 161. [CrossRef]
10. Sethi, K.; Madhav, Y.V.; Kumar, R.; Bera, P. Attention based multi-agent intrusion detection systems using reinforcement learning. *J. Inf. Secur. Appl.* **2021**, *61*, 102923. [CrossRef]
11. Xu, Y.; Li, C.; Zhang, K.; Xia, H.; Tu, B. EI-XIDS: An explainable intrusion detection system based on integration framework. In Proceedings of the 2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Tianjin, China, 8–10 May 2024; pp. 2680–2685. [CrossRef]
12. Le, T.T.H.; Kim, H.; Kang, H.; Kim, H. Classification and explanation for intrusion detection system based on ensemble trees and SHAP method. *Sensors* **2022**, *22*, 1154. [CrossRef]
13. Keshk, M.; Koroniotis, N.; Pham, N.; Moustafa, N.; Turnbull, B.; Zomaya, A.Y. An explainable deep learning-enabled intrusion detection framework in IoT networks. *Inf. Sci.* **2023**, *639*, 119000. [CrossRef]
14. Barnard, P.; Marchetti, N.; DaSilva, L.A. Robust Network Intrusion Detection Through Explainable Artificial Intelligence (XAI). *IEEE Netw. Lett.* **2022**, *4*, 167–171. [CrossRef]

15. Hariharan, S.; Rejimol Robinson, R.; Prasad, R.R.; Thomas, C.; Balakrishnan, N. XAI for intrusion detection system: Comparing explanations based on global and local scope. *J. Comput. Virol. Hacking Tech.* **2023**, *19*, 217–239. [CrossRef]
16. Abou El Houda, Z.; Brik, B.; Khoukhi, L. "Why should i trust your ids?": An explainable deep learning framework for intrusion detection systems in internet of things networks. *IEEE Open J. Commun. Soc.* **2022**, *3*, 1164–1176. [CrossRef]
17. 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*, 6634811. [CrossRef]
18. Islam, S.R.; Eberle, W.; Ghafoor, S.K.; Ahmed, M. Explainable artificial intelligence approaches: A survey. *arXiv* **2021**, arXiv:2101.09429.
19. Arreche, O.; Guntur, T.; Abdallah, M. XAI-IDS: Toward Proposing an Explainable Artificial Intelligence Framework for Enhancing Network Intrusion Detection Systems. *Appl. Sci.* **2024**, *14*, 4170. [CrossRef]
20. Revathi, S.; Malathi, A. A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. *Int. J. Eng. Res. Technol.* **2013**, *2*, 1848–1853.
21. Moustafa, N.; Slay, J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Inf. Secur. J. Glob. Perspect.* **2016**, *25*, 18–31. [CrossRef]
22. Moustafa, N. A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. *Sustain. Cities Soc.* **2021**, *72*, 102994. [CrossRef]
23. Lundberg, S.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. *arXiv* **2017**, arXiv:1705.07874. [CrossRef]
24. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
25. Raffin, A.; Hill, A.; Gleave, A.; Kanervisto, A.; Ernestus, M.; Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. *J. Mach. Learn. Res.* **2021**, *22*, 1–8.
26. Moustafa, N.; Koroniotis, N.; Keshk, M.; Zomaya, A.Y.; Tari, Z. Explainable Intrusion Detection for Cyber Defences in the Internet of Things: Opportunities and Solutions. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1775–1807. [CrossRef]
27. Nwakanma, C.I.; Ahakonye, L.A.C.; Njoku, J.N.; Odirichukwu, J.C.; Okolie, S.A.; Uzondu, C.; Ndubuisi Nweke, C.C.; Kim, D.S. Explainable Artificial Intelligence (XAI) for Intrusion Detection and Mitigation in Intelligent Connected Vehicles: A Review. *Appl. Sci.* **2023**, *13*, 1252. [CrossRef]
28. Bataineh, A.S.; Zulkernine, M.; Abusitta, A.; Halabi, T. Detecting Poisoning Attacks in Collaborative IDSs of Vehicular Networks Using XAI and Shapley Value. *ACM J. Auton. Transp. Syst.* 2024, *accepted*. [CrossRef]
29. Li, J.; Wu, S.; Zhou, H.; Luo, X.; Wang, T.; Liu, Y.; Ma, X. Packet-level open-world app fingerprinting on wireless traffic. In Proceedings of the 2022 Network and Distributed System Security Symposium (NDSS'22), San Diego, CA, USA, 24–28 April 2022.
30. Li, J.; Zhou, H.; Wu, S.; Luo, X.; Wang, T.; Zhan, X.; Ma, X. FOAP: Fine-Grained Open-World android app fingerprinting. In Proceedings of the 31st USENIX Security Symposium (USENIX Security 22), Boston, MA, USA, 10–12 August 2022; pp. 1579–1596.
31. Zhu, H.; Huang, Z.; Lam, C.T.; Wu, Q.; Yang, B.; Ng, B.K. A Space Shift Keying-Based Optimization Scheme for Secure Communication in IIoT. *IEEE Syst. J.* **2023**, *17*, 5261–5271. [CrossRef]
32. Sánchez-Zas, C.; Larriva-Novo, X.; Villagrá, V.A.; Rivera, D.; Marín-Lopez, A. A methodology for ontology-based interoperability of dynamic risk assessment frameworks in IoT environments. *Internet Things* **2024**, *27*, 101267. [CrossRef]