Cybersecurity

RESEARCH

Open Access

Intrusion detection in the internet of things using convolutional neural networks: an explainable AI approach

Fatemeh Ebrahimi[1], Reza Javidan[1*], Reza Akbari[1] and Yasin Hosseini[1]

Abstract

Intrusion Detection Systems (IDSs) with a Machine Learning (ML) technique have shown efficacy in securing Internet of Things (IoT) networks in recent years. As cyber threats continue to evolve, IDS have become increasingly reliant on advanced ML and deep learning (DL) techniques to improve detection accuracy. However, the growing complexity of these models often makes it challenging for security analysts to interpret the reasoning behind specific alerts. While extensive research has been conducted on IDS using ML and DL methods, the issue of interpretability remains largely unaddressed. One of the interpretable methods in machine learning is to use model-agnostic interpretation tools that can be applied to any supervised machine learning model. To address this issue, a new hybrid model composed of a lightweight one-dimensional convolutional Neural Network (1D-CNN) is proposed with the interpretation ability of the results in which, resource-constrained IoT devices can execute the proposed model. In the first phase, the SHapley Additive exPlanations (SHAP) technique is used for feature selection to detect the most important features. These features can be considered for redesigning the model by using a smaller set of features and reducing the computation and complexity of the model, leading to the creation of a lighter deep network. After the prediction of the proposed model, to interpret and explain the results and analyze the influential factors in predictions, Agnostic methods are employed both globally(SHAP) and locally(SHAP, LIME) to clarify the reasons for the predictions. Experimental results using the TON-IoT dataset showed accuracy, precision, recall, and F1-score criteria to 0.995, 0.9949, 0.9947, and 0.9947, respectively. Therefore, besides accurately predicting attacks in the area of IoT with high precision and lightweight models, the proposed method increases transparency to assist cybersecurity personnel in gaining a better understanding of IDS judgments.

Keywords Intrusion Detection System, 1D-CNN, EXplainable Artificial Intelligence, Model Interpretation, Model-Agnostic, LIME, SHAP, Internet of Things

Introduction

Significant advancements in telecommunication networks and the birth of the Internet of Things (IoT) concept have resulted from incredible growth in electronic services and applications (Elrawy et al. 2018). IoT systems have various security vulnerabilities and IoT platforms can lead to numerous security lapses and much damage. Developing an IoT intrusion detection system (IDS) should be able to analyze data packets, produce real-time responses, analyze data packets at different IoT network layers with varying stacks of protocol, and be compatible with various IoT environment technologies (Gendreau and M. Moorman", 2016). An IDS for IoT environments needs to be able to process large amounts of data quickly, with low computing power, and in difficult conditions. Recent advancements in lightweight Machine Learning (ML) architectures present a promising solution to these

*Correspondence:
Reza Javidan
javidan@sutech.ac.ir
[1] Computer Engineering and IT Department, Shiraz University of Technology, Shiraz, Iran

challenges. By prioritizing efficiency without sacrificing detection accuracy, these models are designed to operate effectively within the limited computational and memory resources typical of IoT devices. This is particularly crucial for IoT deployments, where devices often rely on battery power and must perform real-time threat detection to respond promptly to potential intrusions. By improving the operational efficiency of IDS, lightweight models can significantly enhance the overall security posture of IoT ecosystems, ensuring that even low-power devices can contribute to a robust defense strategy. However, the integration of lightweight ML models into IDS also necessitates a focus on explainability.

As cyber threats continue to evolve, IDS has become increasingly reliant on advanced ML and deep learning (DL) techniques to improve detection accuracy. However, the growing complexity of these models often makes it challenging for security analysts to interpret the reasoning behind specific alerts. While extensive research has been conducted on IDS using ML and DL methods, the issue of interpretability remains largely unaddressed. Intrusion detection models are frequently treated as "black boxes," posing difficulties in understanding how predictions are generated. For example, security analysts operating within an IDS framework are tasked with examining IDS alarms for various purposes, including threat mitigation, information gathering, and analysis (Nguyen et al. 2019). The inability to interpret alerts generated by IDS complicates analysis, which in turn hinders decision-making. Transparency remains a persistent challenge in intrusion detection. Cybersecurity professionals increasingly depend on recommendations from an AI-enhanced IDS to inform their decisions (Wang et al. 2020). Consequently, the model's predictions must be clear and understandable. An IDS model might erroneously classify zero-day attacks as routine, potentially leading to a system compromise. The first step in debugging and system detection is determining why certain samples are misclassified. Providing a thorough explanation for these misclassifications is essential for identifying effective strategies to prevent future attacks. Interpretable machine learning models like linear models and decision trees promote transparency and accountability in algorithmic decision-making. Another option is to use model-agnostic interpretation tools that can be applied to any supervised machine learning model Explainable Artificial Intelligence (XAI) techniques aim to provide insights into model behavior, allowing security analysts to interpret detection results and make informed decisions.

The integration of XAI in IDS is essential for enhancing trust and transparency. However, to the best of our knowledge, there are only a few studies that combine XAI techniques with deep learning models like 1D-CNN for intrusion detection. This work bridges that gap by integrating 1D-CNN with XAI techniques to achieve two key goals: (1) delivering accurate intrusion detection and (2) providing clear explanations for model predictions, thereby improving interpretability and encouraging adoption in real-world security systems. One effective approach to enhancing interpretability is the use of model-agnostic interpretation tools, which can be applied to any supervised ML model. To address this, we propose a new hybrid model that combines a lightweight one-dimensional convolutional neural network (1D-CNN) with robust interpretability features. The proposed model is optimized for resource-constrained IoT devices, enabling efficient execution while maintaining transparency in its decision-making process.

Contribution: The following is a summary of this paper's contributions:

- For use in resource-constrained Internet of Things networks, a lightweight model based on a low computational complexity one Dimensional Convolutional Neural Networks1D-CNN) deep learning algorithm is suggested.
- The key features of the network traffic flow for detecting IoT attacks are identified and extracted using an explainable and justifiable method called SHAP, which not only reduces model complexity but also enhances performance.
- A variety of explanation techniques are comprehensively utilized to discern the overall structure of the model and each traffic in global and also local contexts. The interpretation of the model in a global context employs the SHAP technique, while the interpretability of a specific sample for model prediction involves local interpretability using SHAP and LIME techniques.
- Using the recently generated real-world TON-IoT dataset, the proposed approach's performance is evaluated in the context of IoT networks. Based on our best knowledge, there are a few works that have been conducted in the past for the detection and interpretation of the predictions. Numerous criteria, including F1-score, recall, accuracy, and precision are included in this evaluation. The method achieved a high precision in comparison to the previous works.

The paper's remaining sections are arranged as follows. Sect. "Related work" reviews related works. Sect. "The proposed model" provides specifics on the proposed model. The dataset and assessment of the suggested model are described in Sect. "Performance evaluation". Lastly, a summary of the results and plans for further research are provided in Sect. "Conclusion".

Related work

While IDS has seen significant advancements through the machine and deep learning techniques, the interpretability of these systems remains largely underexplored. Despite the increasing scientific interest and contributions in the field of XAI, its application to IDS, particularly in combination with deep learning methods, is still limited. The lack of interpretability in IDS models poses challenges for understanding model before the decisions made by these systems. Here, we provide a summary of key studies in both IDS actions, which is critical for practical deployment, as security experts need to trust and justify and XAI domains, analyzing their contributions, limitations, and the extent to which they address the challenge of model transparency. This discussion highlights the existing gaps and motivates our work on integrating 1D-CNN with XAI to improve both detection accuracy and interpretability.

Min and Raw (Mane and Rao 2021) discussed the difficulty of explaining IDS in computer networks. They stated that when any ML model is used in production, three types of users or consumers come into play. First, a data scientist evaluates the ML model after training. The end user or customer, who wants to know the rationale behind the model's forecast, is the third party; the analyst, on the other hand, makes the ultimate decision based on the model's predictions. They initially used deep learning to build IDS. They utilized a deep neural network whose activation function was a Rectified Linear Unit (RELU). Then they created an XAI framework to maximize their DL-based IDS's transparency. They created an XAI framework to maximize their DL-based IDS's transparency. In addition to creating a DL-based IDS, the authors validated other XAI methods, including ProtoDash, SHAP, LIME, and the Contrastive Explanations method, using the NSL-KDD dataset. This study relies on the NSL-KDD dataset. This dataset has known limitations, such as being outdated and not fully representing modern network traffic patterns. As a result, the findings may not generalize well to real-world scenarios or newer datasets. This paper emphasizes local explanations but may also highlight the importance of global explanations that provide insights into the overall model behavior, which can be crucial for understanding systemic issues in model performance. They did not use more complex algorithms and preprocessing techniques such as data balancing to enhance accuracy.

Wong et al. developed an XAI architecture that employs the SHAP method to augment transparency and provide further elucidation to any IDS system (Wang et al. 2020). To increase interpretability, the writers combined local and global explanations. Additionally, they constructed multiclass and one-vs-all classifiers and contrasted how each classifier was interpreted. They claimed that security professionals may optimize the architecture of intrusion detection systems by comparing how two different classifiers interpreted identical attacks. Then, the improved structure can improve IDS's credibility with human operators. In their research, fully connected networks with ReLU activation have been used. They presented a method based on SHAP to calculate the correlations between attack types and feature values. They have used the NSL-KDD dataset, which is an improved version of the older KDD'99 dataset. More diverse datasets can provide a better evaluation of the framework's effectiveness in various scenarios. They have only utilized the SHAP method for explainability and more complex models have not been used to enhance accuracy and compare performance.

In addition to machine learning-based intrusion detection systems, Wali and Khan created another XAI framework to address adversarial attacks (Wali and Khan 2021). The authors used the SHAP technique to add global explanations to their model after initially developing a random forest classifier. Using the CICIDS dataset and the Hop Skip Jump attack, the framework's performance was assessed. They have used the NSL-KDD dataset like other researchers, but they have only evaluated three categories of attacks. Additionally, they have only utilized local explanations and have not provided a comprehensive interpretation of the results.

Amarasinghe and Manic attempted to increase user confidence by maximizing their transparency, in Deep Learning-based IDSs (Amarasinghe and Manic 2018). This paper describes a procedure for informing customers about DNN-IDS predictions both before and during deployment. The user receives input features associated with each intrusion after the DNN-IDS has been trained and before deployment. The user can see each classifier's factors that contributed during deployment. They claim that this understanding improves DNN-IDS transparency and aids users in comprehending the reasoning behind DNN-IDS judgments. They have also only used the NSL-KDD dataset and have refrained from using other models that would have higher accuracy, they have only utilized Global explanation with the LPR technique.

Khan et al. created a model by using an autoencoder that explains the model and detects attacks in industrial IoT (IIoT) networks using a CNN and Long Short-Term Memory (LSTM) network (Khan et al. 2022). The primary benefit of the architecture was its ability to detect novel and established IIoT attacks (zero-day). They used a combined LSTM and CNN model, achieving results that outperformed other methods and worked well with imbalanced data. They have only used a Global interpretation to identify the most effective features, and their

predictions have been in binary class format. They have not focused enough on the explanations provided.

Houd et al. created a new framework that interprets a DNN-based IDS for IoT networks using XAI techniques including RuleFit, LIME, and SHAP (Houda et al. 2022). Three distinct XAI methodologies have been examined to produce feature importance-based, global, and local interpretations. They used two datasets, NSL-KDD and UNSW-NB15, which are not new, and their predictions were in a binary class format. To increase accuracy, methods such as balancing or more complex models were not employed, and the explanations of the results were only for the two classes of positive and negative, without separate interpretations for each type of attack.

Ables et al. have proposed an architectural framework that consists of three stages: pre-modeling, modeling, and post-modeling Explainibility (Ables et al. 2022). Initially, a high-quality dataset's network data was retrieved and preprocessed, and the SOM model's parameters were chosen. The model is trained in the next phase. The SOM model is trained to begin the modeling stage using the high-quality dataset created during the pre-modeling phase. Ultimately, the post-modeling phase generates informative visuals that help viewers comprehend how the predictions were made. In their article, they provide both global and local reasons. Using the NSL-KDD and CIC-IDS-2017 datasets, experiments were assessed. Although this paper describes local explanations based on the distance from the best matching unit (BMU), it may not explore a broader range of techniques for generating local explanations that could enhance the interpretability of individual predictions. Their proposed method has not demonstrated high accuracy in detection, which may lead to reduced detection performance which is critical in real-world intrusion detection scenarios.

Roshan et al. have used the autoEncoder model to detect intrusion and used XAI to clarify these complex models. They have used the Shapley coefficient and correlation method to select features and made comparisons between these techniques. Their predictions have been in binary classification and only explained these two classes They used local SHAP techniques for the interpretability of the proposed method. Their focus has been on feature selection and they have used the subset of the CIC-IDS2017 dataset to test the proposed model (Roshan and Zafar 2021).

Sarhan et al. have evaluated and compared the NetFlow-based feature set with the feature set designed by the CICFlowMeter tool. Evaluation is done on three datasets (CSE-CIC-IDS2018, ToN-IoT, and BoT-IoT) using two ML classifiers Random Forest (RF) and Deep Forward-Forward (DFF). In addition, the SHAP method has been used to explain the prediction results of ML models by measuring the feature importance, and the key features affecting the prediction of the models have been identified for each data set (Sarhan et al. 2021). Their predictions have been in binary class format, and they have only used the SHAP technique to determine the top features in Global, without explaining the attack and normal classes.

Sharma et al. have used deep learning models to detect attacks in Internet of Things networks. They used a filter-based approach to select features, and then, considering limiting the number of features, they used CNN and Deep Neural Network (DNN) to detect intrusion. They also used two datasets, NSL-KDD and UNSW-NB15, which are relatively old, to evaluate their model. The accuracy achieved using their proposed model has been low in some classes. They then utilized the SHAP technique solely for local interpretability (Sharma et al. 2024).

Tabassum et al. examined the use of Decision tree (DT), RF, Adaboost, XGBoost, ANN, and Multilayer Perceptron (MLP) methods for intrusion detection on Modbus and GPS Tracker datasets. They utilized LIME and SHAP techniques to interpret the results. One of their weaknesses is the lack of use of more complex models to achieve higher accuracy and the absence of necessary explanations for the interpretation charts. Due to device limitations, they were unable to use real-time data (TABASSUM et al. 2022).

Sivamohan et al. have developed an intrusion detection system called the Explainable Artificial Intelligence Framework based on Bidirectional Short-Term Memory (BiLSTMXAI) for effective intrusion identification. They utilized the Krill Herd optimization algorithm to identify important features and employed explainable AI algorithms SHAP and LIME to enhance the interpretation of prediction results. They evaluated their model using the Honeypot and NSL-KDD datasets. Weaknesses of this paper include the lack of common interpretation plots for model explanation and the absence of result comparisons (Sivamohan and Sridhar , 2023).

Table 1 presents a summary of the research in recent years for this purpose. In the analysis conducted and the review of previous works, it is observed that limited research has been done on the explainability of IDS considering the challenges in the IoT. Given the emerging nature of the IoT and the challenges raised in this area including security, there are still many issues in this field. Therefore, research should also focus on newer and more complex attacks specific to this network. On the other hand, an IDS should extend beyond simple intrusion detection. For example, providing reasoning for identified threats. Research conducted by Min and Raw (Mane and Rao 2021), Wong et al. (Wang et al. 2020), Wali and Khan (Wali and Khan 2021), Amarasinghe and Manic

Table 1 Introduction and comparison of the methods presented in the IDS of the IoT network based on XAI

Author/year	Method	Explanation techniques	DataSet	Strengths	weakness
Min and Raw (2021)	Deep neural network with three hidden layers	LIME, SHAP, ProtoDash, Contrastive Explanations	NSL-KDD	*Review of local and Global explanations	*Failure to test the proposed model on other and newer datasets
					*Binary classification without providing specific explanations for each attack
					*No comparison with other paper
					* Not utilizing more complex algorithms to increase accuracy
Wong et al. (2020)	One-vs-All and Multiclass classifiers	SHAP	NSL-KDD	* Review of local and Global explanations	* Failure to test the proposed model on other and newer datasets
					*Using only one explanatory technique
				*Comparing explanations in binary and multiclass classification	*Not utilizing more complex algorithms to increase accuracy
					*Not using data balancing techniques to improve accuracy
Wali and Khan (2021)	Random Forest	SHAP	CICIDS	* Examining the Global explanations of several different attacks	* Review of only three main classes of attack
					*Failure to test the proposed model on other and newer datasets
					*Insufficient explanation regarding the technique used in XAI
Amarasinghe and Manic (2018)	A deep binary neural network and three types of MLPs with different depths	LPR	NSL-KDD	*Comparison between different MLP architectures and explanations	* Failure to test the proposed model on other and newer datasets
					*Not utilizing more complex algorithms to increase accuracy
					*Lack of local explanations
Khan et al. (2022)	CNN + LSTM	LIME	Real System Data	*Comparison of different learning models	*Only using SHAP in two-class classification
					*Failure to provide explanations for different types of attacks
				*Introduction of a new model for attack detection	*Insufficient focus on the explanation

Table 1 (continued)

Author/year	Method	Explanation techniques	DataSet	Strengths	weakness
Houd et al. (2022)	A deep neural network with 5 hidden layers	RuleFit, LIME, SHAP	NSL-KDD, UNSW-NB15	*Identifying influential features in two datasets	* Not using a new dataset
					*Classification of two classes and not providing specific explanations for each attack
				*Using Global and Local explanations	*Not using balancing techniques to increase accuracy on one of the datasets
Ables et al. (2022)	Self Organizing Maps (SOMs)	Unified Distance Matrix(UDM),	NSL-KDD, CIC-IDS-2017	*Use Local and Global explainability analysis	*Not using other algorithms to increase accuracy and compare with others' work
		Best Matching Unit (BMU)			*Failure to pay attention to the imbalance in the dataset
					*Failure to provide more concrete techniques such as LIME and SHAP
Roshan et al. (2021)	AutoEncoder	Shapley values	CICIDS2017	*Using Shapley values and correlation method for feature selection	*Failure to compare results with other models and work
					*Just used Binary Classification
					* Lack of sufficient explanations regarding interpretability and just use of local techniques
Sarhan et al. (2021)	Deep Feed Forward(DFF), RF	SHAP	CSE-CIC-IDS2018, BoT-IoT, ToN-IoT	*Use the Newer dataset	*Not comparing the results with other papers
					*Only providing the Shapley value and not providing interpretations and explanations of the results
				*Compare three dataset	*Just used Binary Classification and don't explain each attack
					*Failure to examine local explanations
Sharma et al. (2024)	CNN, DNN	LIME, SHAP	NSL-KDD, UNSW-NB 15	*Use the Pearson correlation method for feature selection	*Low accuracy in detecting some attacks with the proposed model
					* Not using a new dataset
					* Do not compare with other paper
					*Just Local explanation

Table 1 (continued)

Author/year	Method	Explanation techniques	DataSet	Strengths	weakness
Tabassum et al. (2022)	DT, RF, Adaboost, XGBoost, ANN, and MLP	LIME, SHAP	Modbus and GPS Tracker datasets	*Use Local and Global explainability analysis	* Not use a real-time dataset
					* Not compare with other paper
					*Lack of sufficient explanations regarding interpretability
					*Not using more complex models to achieve higher accuracy
Sivamohan et al. (2023)	BiLSTMXAI	SHAP, LIME	Honeypot and NSL-KDD	*Use Local and Global explainability	* lack of common interpretation plots
					*Failure to test the proposed model on other and newer datasets
				*Compare two dataset	*Do not compare with other paper

(Amarasinghe and Manic 2018), and Khan et al. (Khan et al. 2022) only examined a single dataset, which generally consisted of public datasets and was not necessarily specific to IoT networks.

Considering that our paper focuses on IDS in IoT networks, it is being investigated on a state of the art dataset. According to deep learning algorithms, it can accurately detect various types of attacks. In certain past works, the selection of models or insufficient preprocessing of the datasets resulted in a low level of accuracy in detecting attacks. In most of the research, the explanation techniques have not been fully elaborated on both locally and globally and on different types of attacks. This paper provides explanations alongside attack detection, in both local and global scopes, for various types of attacks to make the causes of predictions transparent for security managers. The absence of information on IDS warnings makes it difficult to analyze them, which in turn makes it difficult to make wise judgments.

The proposed model

Considering the existing gaps and limitations, this section presents a model of an IDS based on XAI in IoT networks to address this gap. At first, the types of attacks are predicted using a 1D-CNN model, and in the next phase, the predictions are interpreted and explained. By outlining the motivations and causes behind each attack, the suggested model not only correctly predicts the different kinds of attacks but also demonstrates their key features. A description of the architecture and components of the proposed method is provided below.

Model architecture

The model that is suggested is shown in Fig. 1. Preprocessing involves turning the datasets in different feature categories into numerical data so they are ready for use in learning models. Subsequently, The data is standardized to ensure that features with larger numerical magnitudes do not disproportionately influence the learning process. This step allows the model to treat all features equally during training, improving convergence and stability. After preprocessing and splitting the data into training and testing sets due to challenges posed by the unbalanced dataset, the Synthetic Minority Over-sampling Technique (SMOTE) has been employed for oversampling. The input data for the proposed 1D-CNN model is then supplied. At this stage, the model predicts various types of attacks. Then, the SHAP technique is applied to the 1D-CNN to identify the most effective features. In other words, important features are identified that will create a lighter model for the IoT network. Once again, data is provided to the proposed model with newly selected features and lower inputs, and the model parameters are adjusted to detect attacks. In the testing phase, after detection, XAI techniques are used to interpret results for better prediction understanding. This stage identifies influential features in detecting each specific attack.

Data preparation

First, the IOT network datasets were investigated to select the most related ones. The majority of intrusion detection techniques currently employed on IoT systems, as demonstrated by the work given in Thereza and Ramli (2023); Chalichalamala et al. , 2023; Nisha et al. , 2024), have concentrated on simulating assaults or on using comparatively older datasets, including NSL-KDD, KDD-99, UNSW-NB15, or CICIDS. The heterogeneity of modern IoT networks, which frequently comprise a variety of protocols, standards, and technologies, is not sufficiently reflected in these datasets.

One of the latest versions of the IoT and IIoT datasets is the TON-IOT dataset, which is designed for assessing security systems based on artificial intelligence. This dataset includes various heterogeneous data sources, such as network traffic datasets, datasets from Windows 7 and 10 operating systems, Ubuntu 14 and 18 TLS datasets, and telemetry datasets from IoT and IIoT sensors. Booij et al. have acknowledged that they examined the importance of the heterogeneity of datasets and demonstrated through numerical experiments that this heterogeneity indeed improves the learning rate of machine learning-based detection algorithms (Booij et al. 2021). Therefore, this study utilizes a state-of-the-art dataset, TON-IoT.

Proposed deep learning model for attack detection

Deep learning algorithms have been the winners of numerous contests in the machine learning and pattern recognition fields in recent years. There are three important reasons why deep learning has become important recently. First, the processing capabilities (e.g. GPU) have significantly increased. Second, the cost-effectiveness of hardware and other recent advances in ML research has improved (Karatas et al. 2018). In real-time applications on resource-constrained IoT devices, 1D-CNNs provide notable advantages over RNNs and LSTMs. Faster inference times result from their capacity to handle input data in parallel, which is crucial for applications that demand quick replies (Zhai et al. 2021). Because of their speed, efficiency, reduced resource requirements, and effective feature extraction capabilities, 1D-CNNs are preferred over RNNs, LSTMs, or hybrid models when it comes to real-time performance on IoT devices. Because of these benefits, 1D-CNNs are especially well-suited for applications that require rapid processing and deployment on

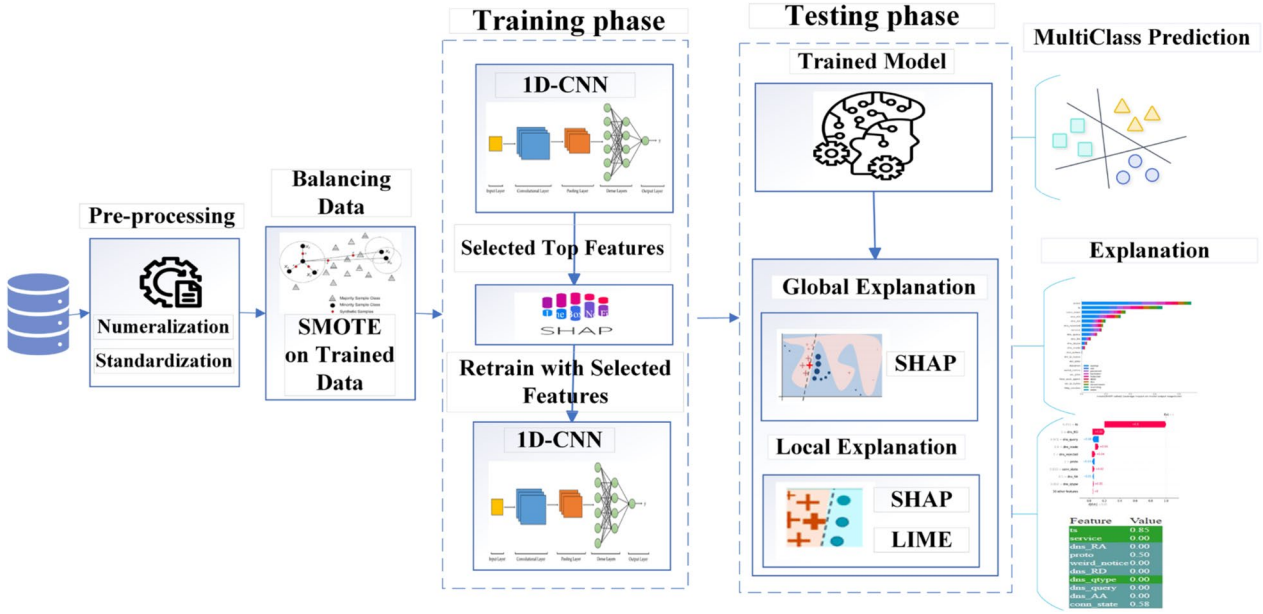


Fig. 1 Proposed Model Structure

hardware with constrained computing power. 1D-CNNs are utilized for the suggested classifiers since they are a good option for classifying network traffic (Lotfollahi et al. 2020). In this study, considering the reasons presented and previous research studies, 1D-CNN algorithms have been used to predict IoT network traffic.

The suggested 1D-CNN model is shown in Fig. 2. Network data is reshaped by the system to make it compatible with 1D-CNN models. The reshaping system provides inputs to the input layer. A 1D-CNN's main building blocks are its convolution layer blocks, that has three blocks of convolution layers. Our goal is to utilize three convolutional layers to represent features at different levels of abstraction. The initial layers learn low-level features, while the subsequent layers capture more complex patterns and relationships within the data. A Max_Pooling layer has been added after the convolutional layers to help reduce the sampling of the feature maps, decreasing their dimensions and ensuring that the most important features are preserved.

On the other hand, CNN frequently has an overfitting issue, necessitating test set parameter fine-tuning. A reduction layer eliminates certain neurons from the training set, lowering the likelihood of overfitting. A regular dropdown does not regularise the activations when feature maps show substantial interrelatedness between successive frames. Instead of dropping individual units, we employed a spatial dropout layer that drops the entire feature maps. Finally, the flattening layer is fully connected to a dense layer. The early stopping strategy is

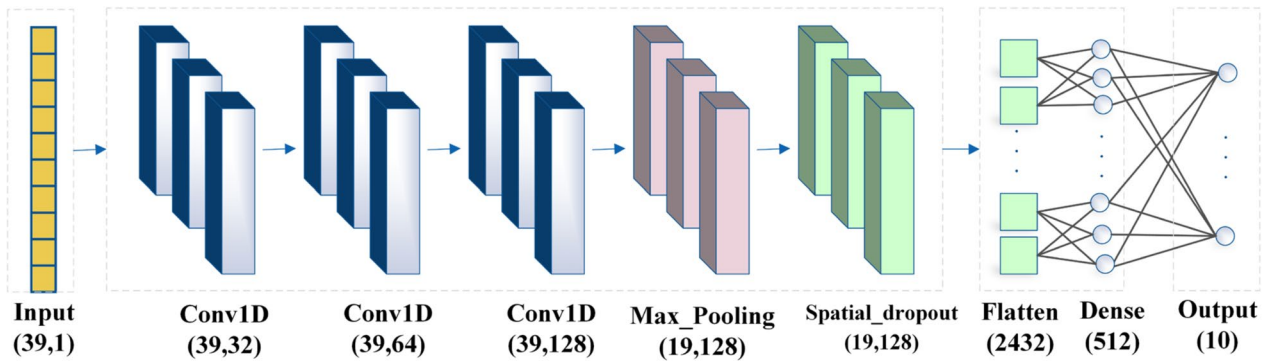


Fig. 2 Proposed 1D-CNN model architecture

also incorporated into the suggested model, which halts the training process to prevent overfitting if the validation set's loss function value remains constant for some rounds and it can save computational resources by avoiding unnecessary training time once the model starts to overfit.

39 neurons compose the input layer of the network in the first phase, which is the total amount of features (excluding "src_ip" and "dest_ip") in the input layer. The output layer, which is the final layer of the model, has an equal number of neurons as the classes that contain the various attack types seen in the dataset as well as normal traffic. Table 2 shows the different parameters of the proposed 1D-CNN model. The settings of the CNN parameters were based on experimental tests and the settings that gave better performance were used.

Feature selection using the global eXplainable method

This paper uses the SHAP technique to identify the features that have had the greatest impact, which is more justifiable. By applying the SHAP technique on the TON-IoT dataset and 1D-CNN models, the best features have been selected. Using the SHAP technique, the Shapley value of each feature, which represents its contribution to the detection of specific traffic, has been determined. Therefore, in this section, the maximum amount of Shapley has been used to select the most effective feature. Using the proposed 1D-CNN architecture and the top 12 features that have shown a positive impact on detection. Selected features include ts, conn_state, proto, dns_qtype, dns_rejected, dns_AA, service, dns_RA, dns_query, dns_RD, dns_rcode, weird_notice. In the next stage, only the selected features identified are used as input features to the proposed model. feature analysis helps the designer select a subset of features and create a model with lower inputs that can be deployed on IoT nodes as a lightweight model with lower computations.

Table 2 Parameter adjustments for the suggested 1D-CNN model

Layer(type)	Parameter of each layer
CNN1D	Filters = 32
	Kernel size = 5
	Padding = 'same'
	Input shape = (39,1)
CNN1D (2)	Filters = 64
	Kernel size = 5
	Padding = 'same'
CNN1D (3)	Filters = 128
	Kernel size = 5
	Padding = 'same'
Maxpooling1D	–
SpatialDropout	0.5
Flatten	–
Dense	512 neuron
Dense	10 neuron

Global eXplanation for the proposed model

Interpretability helps stakeholders understand, which aspect of the model influences the overall predictions. It informs stakeholders about the features the model utilizes for decision-making. Since identifying attacks and their important features is generally crucial for security managers to better identify and detect important attacks, a global explanation of SHAP has also been used to identify the overall model and attacks. This provides a better understanding of the attacks and their influential features. SHAP is an open-source algorithm used to address the accuracy vs. explainability dilemma. The SHAP framework explains how each feature contributes to the output of the model using the idea of Shapley values, a technique from cooperative game theory. The values are a unified measure of feature importance and are used to interpret predictions from machine learning models. SHAP values tell us about the importance of a feature and the direction of the relationship (positive or negative).

The Shapley value ϕ_i for a feature i is computed as Eq. 1:

$$\phi_i = \sum_{S \subseteq F\{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{SU\{i\}}\left(x_{SU(i)}\right) - f_S(x_s) \right] \tag{1}$$

where: F represents the set of all features, while S is a subset of features excluding feature *i*. Within the formula 1, When feature i is present in feature set S, the model's prediction is denoted by the term $f_{SU\{i\}}\left(x_{SU(i)}\right)$. When only the features in subset S are considered, the prediction is denoted by the term $f_S\left(x_s\right)$. Depending on the size of subset S, the fraction indicates the relative weight of the various contributions (Lundberg and S, Lee 2017).

SHAP has become increasingly popular due to the SHAP open source package that was developed:

- A high-speed exact algorithm for tree ensemble methods (called "TreeExplainer").
- A high-speed approximation algorithm for deep learning models (called "DeepExplainer").
- Several model-agnostic algorithms to estimate Shapley values for any model (including "KernelExplainer" and "PermutationExplainer").

Local explanation for each prediction

Focusing solely on global descriptions carries the risk of losing valuable information. For a specific prediction, features that may not be significant on a global scale are not necessarily important, while features that may seem insignificant globally could be very important (Neupane et al. 2022). Therefore, this paper utilizes local interpretability to explain and interpret each prediction individually, identifying the specific features that contribute to each prediction and attack, and striving to detect misclassifications. In addition to employing local interpretability using LIME, this paper also explores different techniques, including SHAP, in a local context for interpreting and explaining DL models. The system architecture of our suggested model is shown in Fig. 7, which also incorporates the LIME model into the planned machine learning pipeline to increase the model's explainability. Among the local explanation methods, the LIME algorithm makes use of a surrogate model to approximate the black box model's predictions. LIME interprets individual predictions using a local surrogate model rather than training a global surrogate model. To explain the predictions at another place based on the simple model, the LIME model simulates the behavior of the complicated model at one location. LIME explicitly tries to model the local neighborhood of any prediction by focusing on a narrow enough decision surface, even simple linear models can provide good approximations of black box model behavior. LIME supports image, text, and Tableau data types. In LIME, interpretable models frequently use decision trees or linear regression that are trained with small perturbations (e.g., adding random noise, deleting specific phrases, and obscuring portions of an image) (Zafar and N. Khan", , 2021). With this method, one data instance's attribute values are changed, and the impact on the output is monitored. It serves as an "explainer" for the predictions made from every data sample (Neupane et al. 2022). Equation 2 is utilized to acquire the description generated by LIME (Lundberg and S, Lee 2017).

$$\xi(x) = \arg\min_{g \varepsilon G} \{ L(f, g, w^x) + \Omega(g) \tag{2}$$

where: g is the explanatory model for sample x. G designates a class of models, including decision trees and linear models, that may be interpreted. f represents the original model, w^x specified weighting factor between the sampled and original data, and L represents the loss function and indicates the measure used to determine the proximity of the predictive explanations to the predictions of the original model. complexity of model g is indicated by $\Omega(g)$.

To implement these techniques, Python programming language is used along with the SHAP and LIME libraries, utilizing functions such as KernelExplainer and LimeTabularExplainer locally, and the SHAP library with the DeepExplainer function globally. To interpret predictions of different attacks, it is necessary to first feed prepared data into learning models such as 1D-CNN, then utilize XAI techniques to provide interpretable explanations for understanding different types of attacks.

Performance evaluation

In this section, we provide the dataset and assess how well the suggested approach works. It is important to note that this paper was written in Python on Google Collaboratory using a T4 GPU and 13 GB of RAM.

Dataset

TON-IoT contains 44 features and encompasses various types of attacks as well as normal traffic, as seen in Table 3 (Booij et al. 2021). The distribution of data in each category of attacks is shown in Table 4. The dataset is randomly divided into three parts: 60% of samples are used as training sets, 20% for validation, and 20% for testing. The attacks found in the TON_IoT network dataset are divided into 9 categories, which is shown in Table 4.

Class unbalance processing

Unbalanced data in machine learning refers to datasets, where the class distribution is skewed. There are three main approaches to addressing the issue of class imbalance: the data-level approach, the cost-sensitive approach, and the algorithm-level approach. A key advantage of the data level approach is its generality since it can be applied to any classifier**.** While algorithm-level techniques and advancements in multi-class classification could be effective in dealing with class imbalance (Demirkıran et al. 2022; Coscia et al. 2024), they can also be complementary to data-level approaches. While several techniques also address the class imbalance in data level generative adversarial networks (GAN) (Zhai et al. 2021), Less Important Components for Imbalanced Multiclass Classification (LICIC) (Vincenzo et al. , 2018), SMOTE, and other algorithms, the SMOTE has a wider range of detection and more established results in various domains, including intrusion detection, and has a strong base of empirical studies demonstrating its effectiveness (Zhai et al. 2021).

For example, GAN can generate new samples that reflect the distribution of minority classes, but training them can be more complex and require fine-tuning the model to avoid state collapse and other issues. For certain applications, SMOTE may provide sufficient improvements without the overhead of GAN training. Of course, it should be mentioned that there are disadvantages to the smote method, such as overfitting or amplifying the

Table 3 Features of the TON_IoT dataset

ID	Feature	Type
Service profile: connection activity		
1	Ts	Time
2	src_ip	String
3	src_port	Number
4	dst_ip	String
5	dst_port	Number
6	Proto	String
7	Service	String
8	Duration	Number
9	src_byte	Number
10	dst_byte	Number
11	conn_state	String
12	missed-bytes	Number
Service profile: statistical activity		
13	src_pkts	Number
14	src_ip_bytes	Number
15	dst_pkts	Number
16	dst_ip_bytes	Number
Service profile: DNS activity		
17	dns_query	string
18	dns_qclass	Number
19	dns_qtype	Number
20	dns_rcode	Number
21	dn_AA	bool
22	dns_RD	bool
23	dns_RA	bool
24	dns_rejected	Bool
Service profile: SSL activity		
25	ssl_version	String
26	ssl_cipher	String
27	ssl_resumed	Bool
28	ssl_established	Bool
29	ssl_subject	String
30	ssl_issuer	String
Service profile: HTTP activity		
31	http_trans_depth	Number
32	http_method	String
33	http_uri	String
34	http_version	String
35	http_request_body_len	Number
36	http_response_body_len	Number
37	http_status_code	Number
38	http_user_agent	Number
39	http_orig_mime_types	String
40	http_resp_mime_types	String
Service profile: violation activity		
41	weird_name	string
42	weird_addl	string
43	weird_notice	bool
service profile: violation activity		
44	Label	Number
45	Type	String

Table 4 Distribution of attacks in the TON_IoT dataset

Training dataset		Testing dataset	
Class	**Distribution**	**Class**	**Distribution**
Normal	192,157	Normal	59,873
Scanning	12,711	Scanning	4035
XSS	12,729	XSS	3976
Dos	12,814	Dos	4051
DDoS	12,658	DDoS	4142
Backdoor	12,742	Backdoor	4085
Injection	12,872	Injection	3952
MITM	649	MITM	218
Password Cracking	12,867	Password Cracking	3963
Ransomware	12,868	Ransomware	3914

noise or overlap of classes, which can be discounted considering its advantages. Considering the advantages of the SMOTE method, it has been used in this paper, and techniques such as Tomek Links have been used to avoid the mentioned problems. Tomek Links help in identifying and removing instances that may be noisy and can be used to identify and remove instances from the majority class that are near instances of the minority class (Peng and Y.J. Park", , 2021). After applying Tomek Links to the dataset, a very small amount of data was detected as noise and this number was also removed from the dataset.

Encoding features

Since the TON_IoT network dataset consists of various types of features including numerical, binary, and nominal features, it is essential to convert nominal features into numerical form to be compatible with many ML algorithms (Guo et al. 2023). In this research, an ordinal encoding method has been used to convert string values into numerical values.

Remove features

IoT devices often operate in dynamic environments where IP addresses frequently change due to Dynamic Host Configuration Protocol (DHCP) assignments. This variability means that IP-based analysis may yield misleading insights. Intrusion detection in IoT contexts is

often more effective when it focuses on device behavior patterns rather than their identities (e.g., IP addresses) (Zohourian et al. 2024). On the other hand, malicious hackers can initiate attacks from computers belonging to authorized users, indicating that the features "src_ip" and "dest_ip" are ineffective for intrusion detection. Including unnecessary features such as IP addresses can lead to more complex models that memorize the training data instead of generalizing to unseen data. This increases the risk of overfitting, where the model performs well on the training data but poorly on the test or validation datasets. The two features "src_ip" and "dest_ip" have been removed from the dataset as they do not influence the target variables.

Imputation of missing values

Missing values are common in large databases. An accurate handling of these values is necessary to produce a valuable analysis. Zero values are substituted for the missing values in the proposed model's numerical characteristics. The '-' values are substituted with the ordinal encoding method for the categorized values.

Data standardization

Numerical features typically have comparatively distinct scales. When this happens, a classifier using raw data produces results that are skewed toward particular features. In this study, standard deviation normalization (also known as Standard Scaler) is employed. StandardScaler uses the normalizing principle to change each feature's distribution so that its mean is equal to zero and its standard deviation is equal to one. By ensuring that every feature is on the same scale, this procedure stops any one characteristic from controlling the learning process because of its greater size. Equation 3 shows the standardization function.

$$X = \frac{x - \mu}{\sigma} \tag{3}$$

where x stands for the standardized feature value, μ for the feature mean, σ for the standard deviation, and x for the original feature value.

Experimental results and discussion

The results of analyzing TON_IoT data and detecting traffic types using 1D-CNN will be presented in the next section. After that, SHAP global explanation techniques are applied and the results are expressed, and finally, Various local description techniques such as SHAP and LIME are presented.

Experimental evaluation using deep learning models

At first, the traffic type was detected using 1D-CNN deep learning algorithms. The model is assessed using the TON-IoT test data for multi-class classification. After training the model, we evaluate its performance with unseen data. The performance evaluation metrics used here are as follows:

Accuracy and Loss: Figs. 3 and 4 show the loss and accuracy of the model before and after feature selection using SHAP.

Precision, recall, and F1-Score: Table 5 compares the TON_IoT evaluation metrics for the deep learning models before and after using SHAP for feature selection. In these criteria, the proposed model has reached 100% precision in detecting Scanning, DoS, Injection, Back-Door, and MITM attacks. As seen from the comparison of the results before and after using SHAP for feature

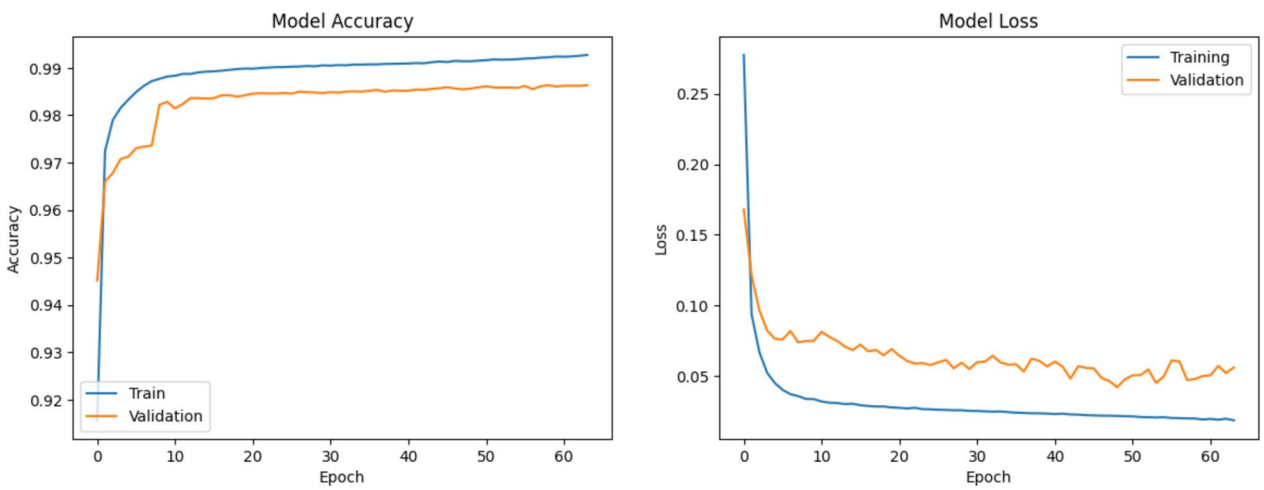


Fig. 3 Accuracy and loss of the proposed 1D-CNN model with all features

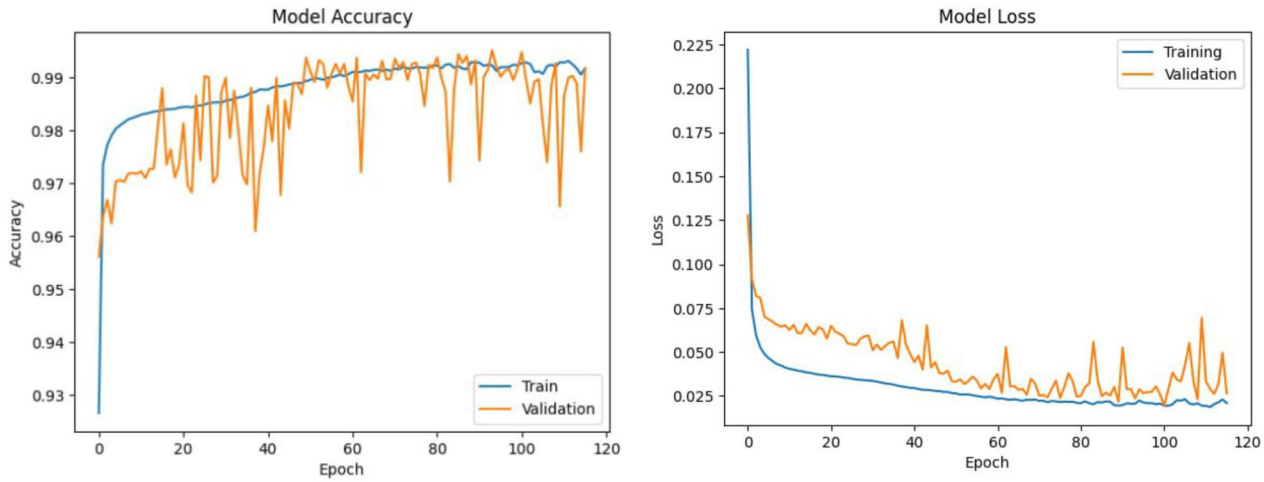


Fig. 4 Accuracy and loss of the proposed 1D-CNN model with selected features(SHAP)

selection in Table 5, after selecting 12 important features and retraining, the model has improved all evaluation metrics. An accuracy of 99.5% indicates that the model correctly classifies 99.5% of the instances. A precision of 99.49% means that of all instances predicted as positive, 99.49% are true positives, suggesting that the model has a good level of correctness in its positive predictions.

A recall of 99.47% shows that the model identifies 99.47% of the actual positive cases, reflecting strong detection capability. An F1-Score of 99.47% highlights a balanced performance between precision and recall, making it a suitable model for contexts where false positives and false negatives have significant implications, such as in intrusion detection systems. Both models have been executed ten times and the average values of the criteria are shown in Table 6. To determine the significant difference between these results, a confidence interval with $\alpha = 5\%$ has been calculated. As can be seen in Table 7, there is a significant difference between the results with 95% accuracy. And definitely, the model has been predicted with higher accuracy after selecting the feature.

Additionally, an important point to note is that by implementing SHAP, feature selection and retraining the model with fewer inputs lead to a model with fewer parameters and reduced computational complexity, ultimately resulting in a lighter model and increased accuracy. This is essential for its implementation in resource-constrained IoT networks. Three main factors are typically considered when assessing the complexity of a 1D-CNN: the number of layers, the size of the layers, and the total amount of network parameters. We can sum together the complexity of each network layer to determine the total complexity of a 1D-CNN. Both the total number of parameters and the number of parameters used in each layer are shown in Table 8. The model's overall number of parameters dropped from 1,302,410 to 409,354 and its size reduced from 4.97 to 1.72 MB after applying SHAP to the 1D-CNN, as demonstrated, without adversely affecting the outcome.

Other popular metrics for evaluating the complexity of statistical models are the Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC). By weighing the trade-off between model simplicity and goodness of fit, they aid in model selection. AIC is a metric that penalizes the amount of parameters in a statistical model while assessing how well the model fits the observed data. The AIC formula is presented in Eq. 4. Similar to AIC, BIC imposes a larger penalty on models with more parameters, Eq. 5 shows that. The proposed models were also examined in Table 9 based on the criteria of AIC and BIC.

$$AIC = 2K - 2\ln(L) \tag{4}$$

$$BIC = \ln(n)K - 2\ln(L) \tag{5}$$

where:

k = number of parameters in the model.

L = likelihood of the model.

n = number of observations (samples).

As can be seen from Table 9, the values of AIC and BIC in the case where feature selection is used are lower and indicate less complexity of the model.

Comparison of the results with other models

In the studies that were conducted, the number of credible papers that utilized the TON-IoT dataset in this field was very limited. Some of these have focused on

Table 5 Evaluation metrics of SMOTE + 1D-CNN, SMOTE + SHAP + 1D-CNN model on TON-IoT dataset

SMOTE + 1D-CNN				
	Accuracy	**Precision**	**Recall**	**F1-Score**
Normal	0.9798	0.9983	0.98	0.989
Scanning	1	1	1	1
Dos	0.9997	1	1	0.9999
Injection	0.9997	0.9997	1	0.9997
DDOS	0.999	1	0.999	0.9995
Password	0.9873	0.9528	0.987	0.9698
XSS	0.9949	0.988	0.995	0.9915
Ransomware	0.9925	0.8009	0.993	0.8865
BackDoor	1	1	1	1
MITM	1	1	1	1
Overall Accuracy	**0.9858**			
Macro AVG		0.974	0.995	0.9836
Weighted AVG		**0.9879**	**0.986**	**0.9863**
SMOTE + SHAP + 1D-CNN				
	Accuracy	**Precision**	**Recall**	**F1-Score**
Normal	0.9938	0.9981	0.9938	0.996
Scanning	1	1	1	1
Dos	1	1	1	1
Injection	0.9972	1	0.9972	0.9986
DDOS	0.9995	0.9957	0.9995	0.9976
Password	0.9931	0.9296	0.9932	0.9604
XSS	0.9823	0.9844	0.9824	0.9834
Ransomware	0.9969	0.9995	0.9969	0.9982
BackDoor	1	1	1	1
MITM	1	1	1	1
Overall Accuracy	**0.995**			
Macro AVG		0.9907	0.9963	0.9934
Weighted AVG		**0.9949**	**0.9947**	**0.9947**

The bold values indicate the final results from the proposed model

Table 6 The average value of the criteria

	AVG accuracy	**AVG precision**	**AVG recall**	**AVG F1-score**
1D-CNN	0.984	0.98612	0.984	0.9846
SHAP + 1D-CNN	0.9946	0.9947	0.9946	0.9946

binary classification, which cannot be compared with our research. Alsaedi et al.utilized several well-known ML methods such as RF, Logistic Regression (LR), Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), k-nearest Neighbour (KNN), Support Vector Machine (SVM), Naïve Bayes (NB), and LSTM for binary and multiclass intrusion detection. They evaluated the models using the TON_IoT dataset (Alsaedi et al.

Table 7 The confidence Interval of the criteria

	Accuracy	**Precision**	**Recall**	**F1-Score**
CI of 1D-CNN	(0.9832,0.9849)	(0.9853,0.9868)	(0.9832,0.9848)	(0.9838,0.9854)
CI of SHAP + 1D-CNN	(0.9941,0.9950)	(0.9943,0.9952)	(0.9941,0.9950)	(0.9941,0.9951)

Table 8 Comparison of the parameters of two models layer by layer

Layers	1D-CNN	SHAP + 1D-CNN
Reshape Layer:	Output Shape: (None, 39, 1)	Output Shape: (None, 12, 1)
	Number of Parameters: 0	Number of Parameters: 0
Conv1D Layer:	Output Shape: (None, 39, 32)	Output Shape: (None, 12, 32)
	Number of Parameters: 192	Number of Parameters: 64
Conv1D Layer:	Output Shape: (None, 39, 64)	Output Shape: (None, 12, 64)
	Number of Parameters: 10,304	Number of Parameters: 2112
Conv1D Layer:	Output Shape: (None, 39, 128)	Output Shape: (None, 12, 128)
	Number of Parameters: 41,088	Number of Parameters: 8320
MaxPooling1D Layer:	Output Shape: (None, 19, 128)	Output Shape: (None, 6, 128)
	Number of Parameters: 0	Number of Parameters: 0
SpatialDropout1D Layer:	Output Shape: (None, 19, 128)	Output Shape: (None, 6, 128)
	Number of Parameters: 0	Number of Parameters: 0
Flatten Layer:	Output Shape: (None, 2432)	Output Shape: (None, 768)
	Number of Parameters: 0	Number of Parameters: 0
Dense Layer:	Output Shape: (None, 512)	Output Shape: (None, 512)
	Number of Parameters: 1,245,696	Number of Parameters: 393,728
Dense Layer:	Output Shape: (None, 10)	Output Shape: (None, 10)
	Number of Parameters: 5,130	Number of Parameters: 5,130
Total Parameter	1,302,410	409,354
Size	4.97 MB	1.72 MB

Table 9 Comparison of the complexity of two models

	AIC	BIC
1D-CNN	2,604,820.069	17,055,086.31
SHAP + 1D-CNN	900,884.039	5,898,584.50

2020). The results obtained from CART have been better compared to other methods, achieving an accuracy of 0.77 across all criteria, which shows a significant gap from the results attained in our research. In a paper presented by Gad et al., several ML techniques, including LR, NB, RF, DT, SVM, KNN, AdaBoost, and XGBoost, were employed for attack detection. The TON-IoT dataset was used to evaluate and compare the methods. They utilized the Chi2 technique for feature selection and the SMOTE method for data balancing. The best results were achieved when no changes were made to the data and the XGBoost method was utilized (Gad et al. 2021). To compare with the method presented in this paper, this approach has been implemented, we attained an accuracy of 0.9748, 0.975, 0.9748, and 0.974 for precision, recall, and F1 score, respectively. Zhong Cao et al. have used two datasets of ToN-IoT consisting of the Network and the IoT device dataset (Cao et al. , 2024). They also employed SMOTE and Tomek approaches for dataset balance. They used sine and cosine component cyclic encoding for temporal features. They have evaluated their approach with each of the datasets and the combined dataset. They have used various models such as RF, CART, KNN, LSTM, DNN, and CNN. The best evaluation results on the Network dataset are presented in Table 10. As observed, our proposed model has achieved improved performance compared to theirs. Of course, the dataset was also evaluated with the basic models such as SVM and MLP to provide a more robust comparison, as shown in Table 10.

Another metric that can be calculated is Timing Performance. It describes how long it takes a model to complete particular tasks, such as training and inference. Table 11 demonstrates the performance of various models in terms of accuracy and time consumption. The rows represent a specific study or paper that proposes a model for intrusion detection. The table includes accuracy and time consumption results for our proposed model on TON-IoT. Each dataset has its accuracy values and corresponding training and inference times specified. As can be seen, most of these papers lack time performance analysis. However, in the paper presented by Hnamte et al. (Hnamte and J. Hussain", , 2023), there are time performance values that can be compared. For a fair comparison, all conditions including the environment in which the model is trained and the total number of training data, must be equal to ensure a just comparison. It is worth mention that the environment in which they executed their model is several times more powerful than

Table 10 Comparison table on the Ton-IOT dataset

Author	Classifier	Accuracy	Precision	Recall	F1-Score
Alsaedi et al. (2020)	CART	0.77	0.77	0.77	0.75
Gad et al. (2021)	XGBoost	0.98	0.98	0.98	0.98
Cao et al.(2024)	CNN + S + T	0.9887	0.9936	0.9918	0.9918
	SMOTE + MLP	0.96	0.97	0.96	0.96
	SMOTE + SVM	0.91	0.92	0.91	0.91
Proposed Model	SHAP + SMOTE + 1D-CNN	0.995	0.9949	0.9947	0.9947

Table 11 Result in comparison with time consumption

Author	Model	Dataset	Accuracy	Precision	F-score	Recall	Training Time	Inference Time
Hnamte et al. (Hnamte and J. Hussain", , 2023)	DCNN	ISCX2012, DDoS (Imbalance) (Kaggle), DDoS (Balance) (Kaggle), CICIDS2017, CICIDS2018	0.9979,0.9999, 1,0.9996,1	0.9979,0.9999 1,0.9996,1	0.9978,0.9999, 1, 0.9996,1	0.9978,0.9999, 1, 0.9996,1	26 s,10 s, 17 s, 40 s,15 s	19.50 s,7.11 s, 11.79 s, 29.36 s,9.91 s
Sharma et al. (Thereza and Ramli 2023)	2D CNN model	NSW-Nbnew, NSL-KDDnew	0.81,0.99	*	*	*	*	*
Patil et al.(Patil et al. , 2022)	Voting classifier	CICIDS-2017	0.9625	0.89	0.89	0.89	*	*
Mahbooba et al.(Mahbooba et al. 2021)	DT	KDD	*	1	*	1	*	*
Arslan et al. (Bisharat et al. 2014)	1DCNN	Edge-IIoTset cybersecurity dataset	0.999	0.988	0.9878	0.9879	*	*
Proposed Model	**SHAP + 1D-CNN**	**TON-IoT**	**0.995**	**0.9949**	**0.9949**	**0.9947**	**2220 s**	**0.004489 s**

* Not available

The bold values indicate the results of the proposed model

the execution environment in this paper. We reported the training time based on the entire training data and the inference time for a single data point.

Using XAI for model interpretation

Global eXplanation

One of the main objectives of this paper was to explain and interpret IDS predictions. In this step, after modeling and identifying the type of traffic, the SHAP technique was used to obtain global explanations of the predictions. This section discusses the results and known features of each type of traffic. SHAP Summary plot for the TON-IoT dataset corresponds to the 1D-CNN algorithms, represented by Fig. 5. The features are listed on the Y-axis according to their importance, while the X-axis shows the average value of SHAP. The length of the bar tells us how much influence the feature has on the prediction. 'ts' is the most important feature for all classes. Table 12 lists the order of influential features in detecting normal traffic and various attacks using the SHAP technique. These features are sorted based on their Shapley values.

Local eXplanation

In addition to general explanations for the proposed model, LIME and SHAP are utilized in this paper to provide a local explanation for each specific sample. To conduct a more thorough examination, samples from various attacks have been selected for detailed analysis using

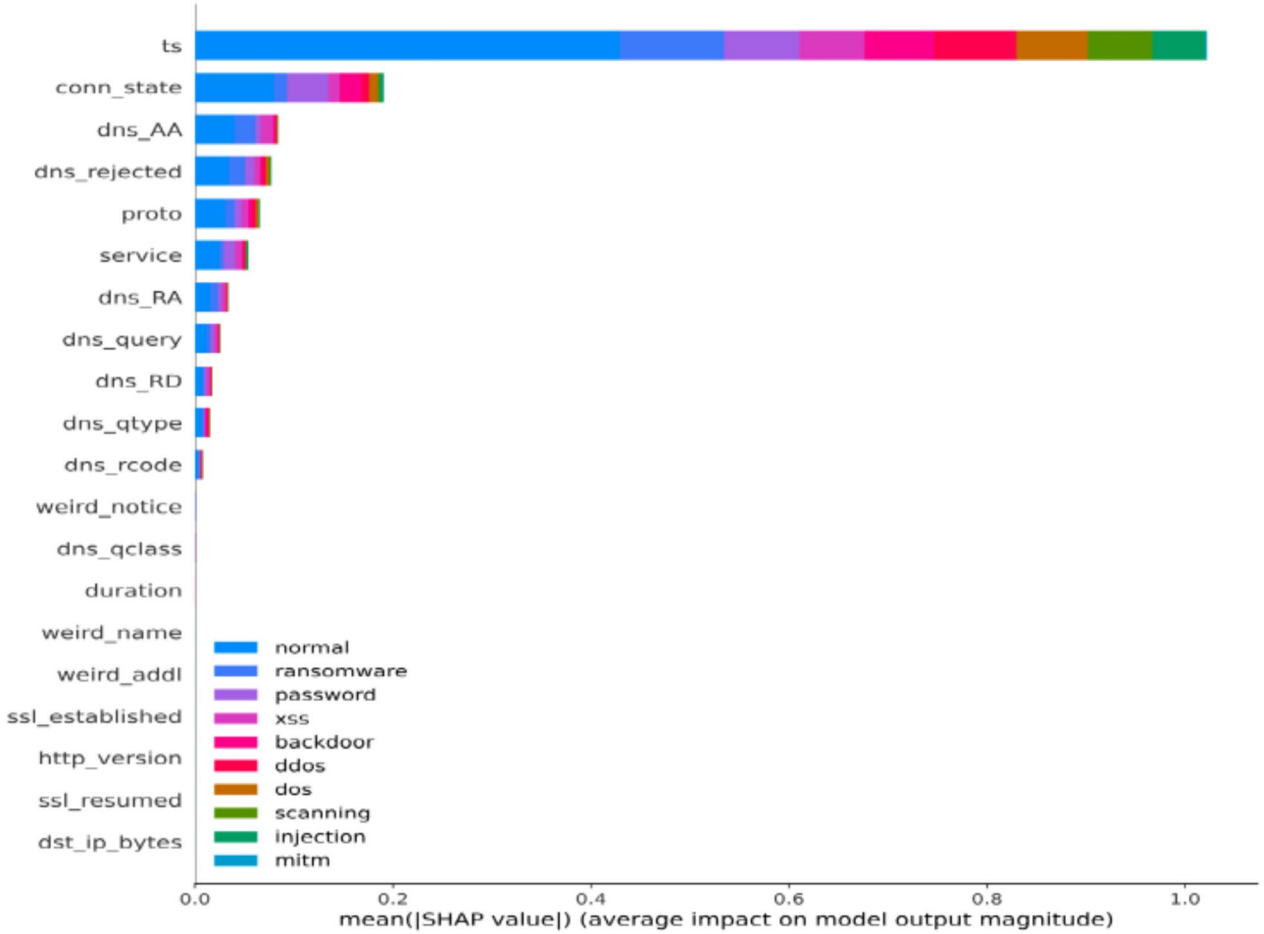
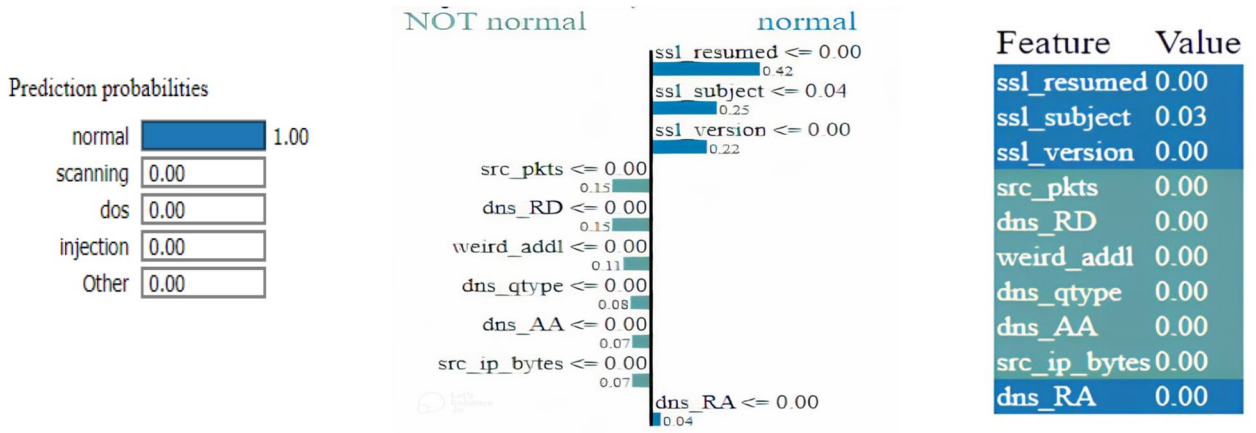


Fig. 5 Summary plot of the SHAP algorithm on the 1D-CNN model

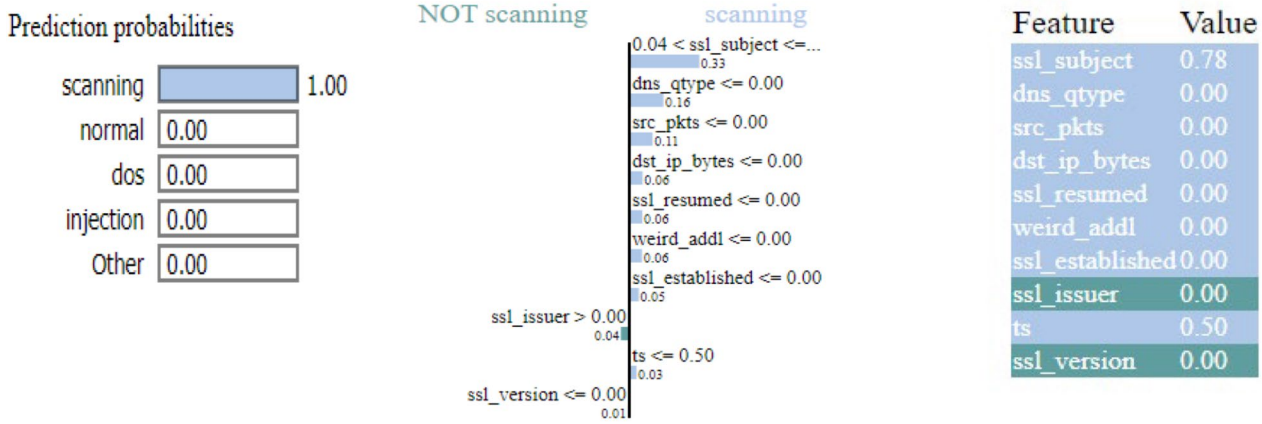
Table 12 Order of influential features in various types of traffic by the proposed model using the SHAP technique

Backdoor	DDOS	DOS	Injection	Mitm	Normal	Password	Ransomware	Scanning	XSS
ts	ts	ts	ts	ts	ts	ts	ts	ts	ts
conn_state	conn_state	conn_state	conn_state	conn_state	conn_state	conn_state	dns_AA	conn_state	dns_AA
proto	Proto	dns_rejected	service	dns_RA	dns_AA	service	dns_rejected	dns_rejected	conn_state
dns_qtype	dns_rejected	proto	dns_rejected	service	dns_rejected	dns_rejected	conn_state	Proto	service
dns_rejected	Service	dns_RA	Proto	proto	proto	proto	proto	service	proto
dns_AA	dns_AA	dns_AA	dns_query	dns_query	service	dns_AA	dns_RA	dns_AA	dns_rejected
service	dns_rcorde	dns_qtype	dns_rcorde	dns_RD	dns_RA	dns_RA	service	dns_qtype	dns_RA
dns_RA	dns_query	dns_RD	dns_RD	dns_AA	dns_query	dns_query	dns_query	dns_rcode	dns_query
dns_query	dns_RD	service	dns_qtype	dns_rejected	dns_RD	dns_RD	dns_qtype	dns_query	dns_RD
dns_RD	dns_qtype	dns_query	dns_AA	dns_qtype	dns_qtype	dns_rcode	dns_RD	dns_RA	dns_qtype
dns_rcode	dns_RA	dns_rcorde	dns_RA	dns_rcorde	dns_rcorde	dns_qtype	dns_rcode	dns_RD	dns_rcode

different plots. As seen in Fig. 6, we took two samples and predicted each class. The likelihood of each class is displayed on the left side of the figure, While the importance of features is shown in the middle bar chart and the top ten features and their respective values are displayed on the right. Green indicates the feature's negative influence, whereas blue indicates its good impact. Figure 6a shows a sample of normal traffic as observed in the left graph,



(a). A Normal sample that was correctly identified



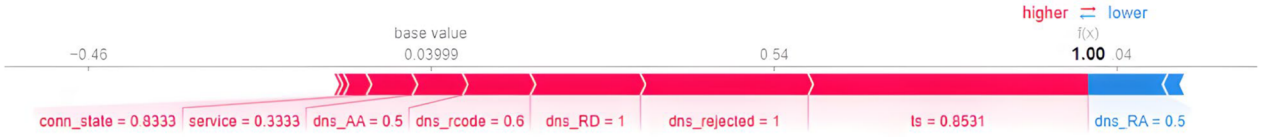
(b). A Scanning sample that was correctly identified

Fig. 6 LIME explanations for instances accurately predicted by the model

which has been identified as normal traffic with 100% accuracy and displays the top ten features in the middle.

The features ssl_resumed, ssl_version, and dns_RA have values ≤ 0.00 and ssl_subject ≤ 0.04, respectively, and the weights applied are 0.42, 0.22, 0.04, and 0.25 to predict the sample as Normal. The features src_pkts, dns_RD, weird_addl, dns_qtype, dns_AA, and src_ip_bytes have values ≤ 0.00 to predict a sample as Not Normal, and the weights assigned are 0.15, 0.15, 0.11, 0.08, 0.07, and 0.07, respectively. In the right chart, the value of each feature is determined based on its weight in the middle chart, leading to the final values for these features, which are 0.0075 for normal traffic and 0.00 for Not normal traffic. Therefore, this traffic is identified as normal. Similarly, as seen in Fig. 6b, for another example categorized as a scanning attack, the model is recognized as correct with a probability of 100%.

The Shapley value, which illustrates how features affect the model's predictions, is computed using SHAP. We computed the shape values for a specific instance that we chose. The local plot of the DDoS instance, which displays the contribution of each feature to the prediction, is displayed in Fig. 7. The figure indicates that the features that have a positive influence on the predictions are displayed in red, while the features that have a negative impact are displayed in blue. As seen in Fig. 7a, the features "ts," "dns_rejected," "dns_RD," "dns_rcode," "dns_AA," "service," and "conn_state" have helpful contributions to the prediction value, while "dns_RA" has negative effects for the specific record chosen from the dataset The anticipated class is DDoS since the overall positive contribution exceeds the whole negative contribution. Figure 7b is another example of a selected normal class, and the model has



(a). A DDoS sample that was correctly identified

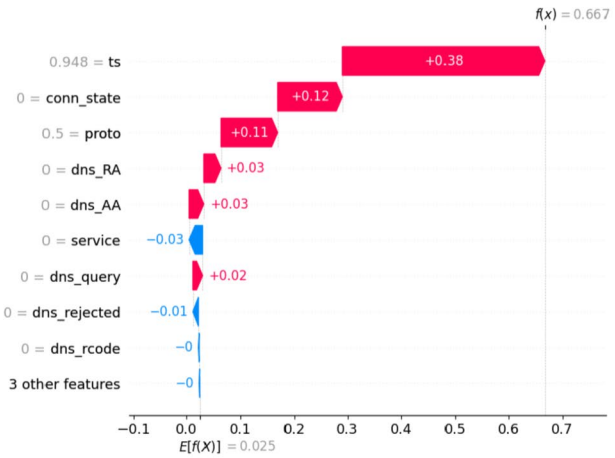
(b). A Normal sample that was correctly identified

Fig. 7 Force plot explanations for two samples correctly predicted by the model

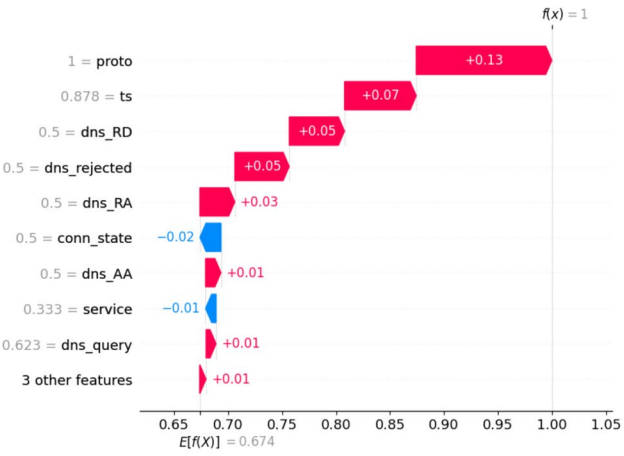
correctly identified it as normal, with observable influential features.

A waterfall plot in XAI is another visualization tool used for understanding the contribution of individual features to a specific instance's prediction. Each bar in the plot shows the contribution of a feature. The bars in red show that the feature positively influences the prediction. The bars in blue show that the feature negatively influences the prediction. In Fig. 8a, ts, conn_state, proto, dns_RA, dns_AA, and dns_query are features with a positive impact, and service, dns_rejected, and dns_rcode are features with a negative impact on predictions. Figure 8b shows the waterfall plot for a normal sample.

As can be seen from the interpretation of predictions Providing feedback on the model's predictions is a way for human operators to contribute to continuous improvement. The identification and understanding of attack patterns require interpretable machine learning in IDS. Identifying common attributes of attacks is possible for analysts by analyzing the features used in decisions. Security professionals trust systems more when they are transparent about their predictions, which encourages them to rely on and act on the model's output. This can lead to quicker responses to identified threats. The model's understanding of attack patterns over time is made more clear through this interaction. Security



(a). A Ransomware sample that was correctly identified



(b). A Normal sample that was correctly identified

Fig. 8 SHAP waterfall plot explanations for two samples correctly predicted by the model

teams can quickly devise appropriate incident response strategies when interpretable models identify patterns that indicate potential threats. Clearness in data processing and decision-making algorithms is required by regulatory standards in many industries. These requirements can be satisfied by interpretable models that provide explanationable outputs. By interpreting, biases in decision-making processes can be mitigated, and the IDS cannot unfairly target specific traffic patterns without justification.

Conclusion

One of the goals of XAI is to make end users aware of the reasoning behind decisions or recommendations generated by an AI system. Nowadays, IDS do not provide any information about their decisions. One of the main goals of this paper has been the interpretability of IDS. In this paper, a hybrid model for IDS detection and interpretation was presented, focusing on the deep learning model and XAI. A deep learning architecture based on 1D-CNN has been proposed for IoT networks, taking into account their resource limitations, to identify the type of traffic. In the first step, the SHAP algorithm was used to select features, which not only reduced the complexity of the model but also improved its performance. Additionally, global explanation methods such as SHAP and local explanation methods like LIME and SHAP have been employed to interpret deep learning models, aiding in understanding the reasons behind the decisions made by IDS. The model presented in this paper has the potential to improve IDS transparency. As a result, professionals in cybersecurity are better able to distinguish between different types of attacks and make wiser decisions. One limitation of the proposed model is its scalability, as it may face challenges in large-scale environments, where the volume of data and the number of devices can be significant. Additionally, the performance of the proposed approach in heterogeneous IoT environments, which may have different communication protocols, needs to be thoroughly evaluated. Future work could focus on optimizing it for real-time deployment, utilizing techniques that reduce inference time so that the system can respond appropriately to threats as they occur. Furthermore, by employing federated learning and creating decentralized model training without data sharing, privacy and security can be enhanced, allowing the proposed model to be generalized across various environments. Of course, the proposed model can also be evaluated on newer datasets and ensure that the explanations provided by the models are appropriate or satisfy the intended requirements when providing explanations for the models.

Author contributions
The first author, Fatemeh Ebrahimi, implemented the complete code and wrote the draft of the paper. The second author, Reza Javidan, supervised, approved the idea, and made corrections to the paper and the correspondence. The third author, Reza Akbari, gave some advice and initial ideas. The fourth author, Yasin Hoseini, implemented some parts of the initial code.

Funding
There is no funding for this research.

Data availability
No specific dataset is generated for this paper.

Declarations

Conflict of interest
The authors declare that there are no competing interests.

Received: 12 September 2024 Accepted: 22 January 2025
Published online: 05 September 2025

References

Ables J, Kirby T, Anderson W, Mittal S, Rahimi S, Banicescu I, Seale M (2022) Creating an explainable intrusion detection system using self organizing maps ArXiv abs/2207.07465

Alsaedi A, Moustafa N, Tari Z, Mahmood A, Anwar A (2020) TON_IoT telemetry dataset: a new generation dataset of IoT and IIoT for data-driven intrusion detection systems. IEEE Access 8:165130–165150

Amarasinghe K, Manic M, (2018) Improving user trust on deep neural networks based intrusion detection systems, In: Proc 44th Annu Conf IEEE Ind Electron Soc (IECON), pp. 3262–3268

Bisharat A, Mubeen M, Bilal M, Abbasi S, (2014) 1D-CNN-IDS: 1D CNN-based intrusion detection system for IIoT, https://doi.org/10.48550/arXiv.2409.08529

Booij TM, Chiscop I, Meeuwissen E, Moustafa N, den Hartog FTH (2021) ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. IEEE Internet Things J 9:1–1

Cao Z, Zhao A, Shang W et al (2024) Using the ToN-IoT dataset to develop a new intrusion detection system for industrial IoT devices. Multimed Tools Appl. https://doi.org/10.1007/s11042-024-19695-7

Chalichalamala S, Govindan N, Kasarapu R, (2023) A comprehensive analysis of intrusion detection in internet of things (IoT), https://doi.org/10.1109/aikiie60097.2023.10390177

Coscia A, Iannacone A, Maci A, Stamerra A (2024) SINNER: a reward-sensitive algorithm for imbalanced malware classification using neural networks with experience replay. Information 15(8):425

Demirkıran F, Çayır A, Ünal U, Dağ H (2022) An ensemble of pre-trained transformer models for imbalanced multiclass malware classification. Comput Secur 121:102846

Elrawy M, Awad A, Hamed H (2018) Intrusion detection systems for IoT-based smart environments: a survey. J Cloud Comp. https://doi.org/10.1186/s13677-018-0123-6

Gad G, Abdallah N, Ahmed T (2021) Intrusion detection system using machine learning for vehicular Ad Hoc networks based on ToN-IoT dataset. IEEE Access. https://doi.org/10.1109/ACCESS.2021.3120626

Gendreau AA, Moorman M, (2016) Survey of intrusion detection systems towards an end to end secure internet of things, In: 2016 IEEE 4th International Conference on Future Internet of Things and Cloud(FiCloud). IEEE, Vienna. pp. 84–90

Guo G, Pan X, Liu H, Lie F, (2023) An IoT Intrusion detection system based on TON IoT network dataset, IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC): pp. 0333–0338

Hnamte V, Hussain J (2023) Dependable intrusion detection system using deep convolutional neural network: a novel framework and performance evaluation approach. Telematics Inform Rep. 11:100077. https://doi.org/10.1016/j.teler.2023.100077

Houda ZAE, Brik B, Khoukhi L (2022) Why should i trust your IDS ?": an explainable deep learning framework for intrusion detection systems in internet of things networks. IEEE Open J Commun Society 3:1164–1176
Karatas G, Demir O, Sahingoz OK, (2018) Deep Learning in Intrusion Detection Systems. In: 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), pp. 113–116
Khan IA, Moustafa N, Pi D, Sallam KM, Zomaya AY, Li B (2022) A new explainable deep learning framework for cyber threat discovery in industrial IoT networks. IEEE Internet Things J 9(13):11604–11613
Lotfollahi M, Jafari Siavoshani M, Shirali Hossein Zade R, Saberian M (2020) Deep packet: a novel approach for encrypted traffic classification using deep learning. Soft Comput 24(3):1999–2012
Lundberg SM, Lee S, (2017) A unified approach to interpreting model predictions, Adv neural inform process syst
Mahbooba B, Timilsina M, Sahal R, Serrano M (2021) Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model. Complexity. https://doi.org/10.1155/2021/6634811
Mane S, Rao D (2021) Explaining network intrusion detection system using explainable AI framework, arXiv: 2103.07110
Neupane S, Ables J, Anderson W, Mittal S (2022) Explainable intrusion detection systems (X-IDS): a survey of current methods, challenges, and opportunities. IEEE Access 10:112392–112415
Nguyen QP, Lim KW, Divakaran DM, Low KH, Gee M Ch Chan (2019) A gradient-based explainable variational autoencoder for network anomaly detection, 2019 IEEE Conference on Communications and Network Security (CNS), pp. 91–99
Nisha N, Nasib SG, Preeti G (2024) A review on machine learning based intrusion detection system for internet of things enabled environment. Int J Electric Comput Eng 14:1890–1898
Patil S, Varadarajan V, Mazhar SM, Sahibzada A, Ahmed N, Sinha O, Kumar S, Shaw K, Kotecha K (2022) Explainable artificial intelligence for intrusion detection system. Electronics 11(19):3079
Peng CY, Park YJ (2021) A new hybrid under-sampling approach to imbalanced classification problems. Appl Artificial Intell. https://doi.org/10.1080/08839514.2021.1975393
Roshan K Zafar A. (2021) Utilizing XAI technique to improve autoencoder based model for computer network anomaly detection with shapley additive explanation(SHAP), ArXiv abs/2112.08442
Sarhan M, Layeghy S, Portmann M (2021) Evaluating standard feature sets towards increased generalisability and explainability of ML-based network intrusion detection. Big Data Res. https://doi.org/10.1016/j.bdr.2022.100359
Sharma B, Sharma L, Lal C, Roy S (2024) Explainable artificial intelligence for intrusion detection in IoT networks: a deep learning based approach. Expert Syst Appl 238:121751
Sivamohan S, Sridhar S (2023) An optimized model for network intrusion detection systems in industry 4.0 using XAI based Bi-LSTM framework. Neural Comput Applic 35:11459–11475
TABASSUM S, Sabrina, Parvin N, Hossain N (2022) Network attack detection using XAI and reliability analysis. In: 2022 25th International Conference on Computer and Information Technology (ICCIT). IEEE, p. 176–181
Thereza N, Ramli K(2023) Development of Intrusion detection models for IoT networks utilizing CICIoT2023 Dataset
Vincenzo D, Impedovo D, Pirlo G (2018) LICIC: less important components for imbalanced multiclass classification. Information 9(12):317
Wali S, Khan I (2021) Explainable AI and random forest based reliable intrusion detection system, https://doi.org/10.36227/techrxiv.17169080.v1
Wang M, Zheng K, Yang Y, Wang X (2020) An explainable machine learning framework for intrusion detection systems. IEEE Access 8:73127–73141
Zafar MR, Khan N (2021) Deterministic local interpretable model-agnostic explanations for stable explainability. Mach Learn Knowledge Extract 3:525–541
Zhai J, Qi J, Zhang S (2021) Imbalanced data classification based on diverse sample generation and classifier fusion. Int J Mach Learn Cybernetics. https://doi.org/10.1007/s13042-021-01321-9
Zohourian A, Dadkhah S, Molyneaux H, Euclides HN, Ghorbani A (2024) IoT-PRIDS: Leveraging packet representations for intrusion detection in IoT networks. Computers Security 146:104034

Publisher's Note