International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

Explainable AI Framework for Intrusion Detection Systems

Sidhant Kumar[1] Sagar Choudhary[2] Anil Kumar Yadaw[3]

*1,3B.Tech Student, Department of CSE, Quantum University, Roorkee, India
2Assistant Professor, Department of CSE, Quantum University, Roorkee, India.

How to Cite this Article:

Kumar, S. & Yadaw, A. K. (2026). Explainable AI Framework for Intrusion Detection Systems. International Journal of Creative and Open Research in Engineering and Management, 02(05). https://doi.org/10.55041/ijcope.v2i5.750

https://doi.org/10.55041/ijcope.v2i5.750

Abstract.

Cybersecurity is a fast-changing field where data patterns are always shifting. Attackers develop new ways to break into digital systems and networks. Due to this ongoing threat, Intrusion Detection Systems (IDS) are vital for protecting modern cyber infrastructure. Recently, Machine Learning (ML) and Deep Learning (DL) based intrusion detection systems have shown notable improvements in spotting malicious activities and identifying cyber-attacks more accurately. Deep neural networks can learn complex patterns from large datasets, which leads to better detection compared to traditional methods. However, as these models become more accurate and complex, they also become harder to understand, making it challenging to trust and use them in real-world situations. Many deep learning models operate like “black boxes”; the reasoning behind their predictions is often unclear.

This research paper suggests an Explainable Artificial Intelligence (XAI) framework for Network Intrusion Detection Systems. The goal is to improve transparency and trust in machine learning-based security solutions. The framework combines deep neural networks with explainability techniques to give useful insights into the decision-making process of the model throughout different stages of the machine learning pipeline. The study uses the NSL-KDD dataset to assess the performance of this approach. Various XAI techniques, such as SHAP, LIME, Contrastive Explanations Method (CEM), ProtoDash, and Boolean Decision Rules via Column Generation (BRCG), are used to create explanations for the predictions made by the IDS model. These methods help identify which features have the most impact on detecting cyber-attacks and evaluate their influence on the final prediction. The results show that combining deep learning with explainable AI can improve both detection accuracy and model transparency in cybersecurity applications.

Keywords: classification, intrusion detection system, cybersecurity, explainability, SHAP scores, explainable AI, Deep neural network, SHAP, LIME, AIX 360, BRCG, CEM, ProtoDash, local explanations, global explanations, rules

1. Introduction

Recently, Machine Learning (ML) has been used in modern intrusion detection systems to correlate features, classify patterns and identify outliers (anomalies), indicating a potential attack. Network security researchers spend hours analyzing attacks and classifying them into one of several attacks, for example, port sweep, password guess, teardrop. The attack landscape is always changing however, with hackers continuously discovering new ways to attack. Therefore, such general classifications could be obsolete with the adoption of new attack strategies by hackers. Moreover, as ML systems are often seen as black boxes that lack transparency regarding how or why certain network traffic is flagged, a recently emerging area of research, explainable artificial intelligence (XAI), comprises works that address explaining the predictions of ML models. Hence, this paper also proposes an XAI framework along with the proposed intrusion detection system to assist an analyst to take the final decision.

For example, if we classify an attack as a "guess password" and an explanation indicates this was due to one hot indicator and approximately 125 source bytes. If the security analyst agrees that this explanation seems reasonable, it is easy for the security analyst to accept the alert. In the case of a new attack, when an anomalous attack is detected and an explanation provided, the analyst could decide that this is a new abnormal attack pattern and add an appropriate rule to a system such as Zeek (formerly Bro-IDS). For future work, this could be paired with a natural language generation module, generating sentences in English.

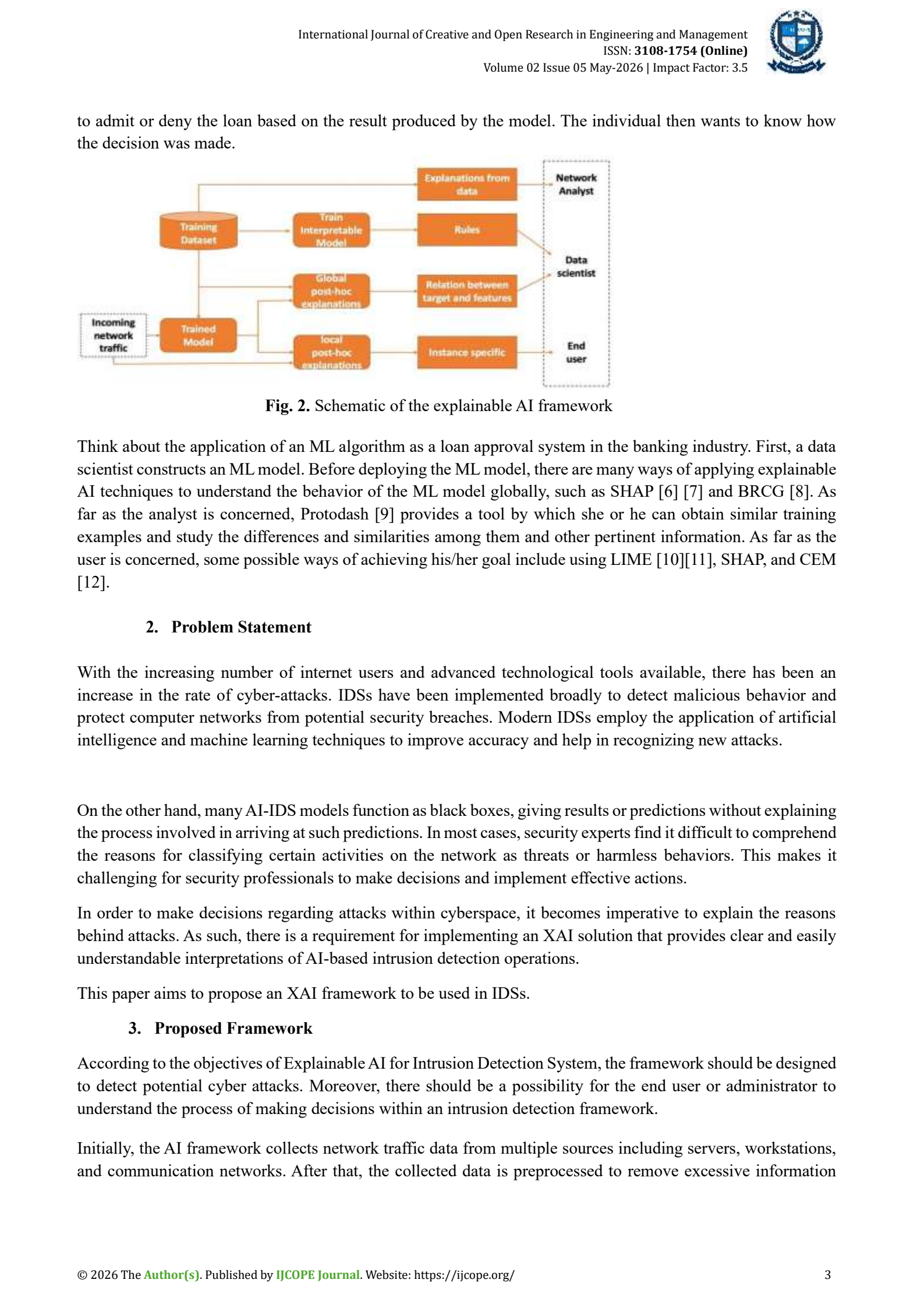
People have noticed that Deep Neural Networks (DNNs) consistently edge out other machine learning algorithms when it comes to performance.. But here's the catch: if you look at the accuracy-versus-interpretability tradeoff, you'll see a clear pattern, boosting model accuracy often chips away at how easily we can make sense of what's going on under the hood. That's not a trivial issue. In fact, this very lack of transparency is part of why DNNs rarely show up in real-world settings like bank loan approvals; folks simply don't trust decisions they can't unpack.. So, what can we do about it?.

Fig. 1. Steps involved in ML pipeline

Here are the various stages in the ML pipeline. We can obtain explanations at each of these stages using explainable AI algorithms. Explanation at the training dataset level allows us to determine if there are any biases present in the data set that may be corrected before training begins. At the trained ML algorithm level, explanations allow us to determine if there is anything wrong that the model has learned during the training process. If that is the case, then we can improve the performance of the model through tuning or reselecting new features. Instance-level explanations allow us to debug the model but also prove useful post-deployment. Therefore, at every step of the ML pipeline, these explanations make it easy to develop and deploy the ideal network intrusion detection system. Figure 2 shows how these explanations can be generated for all categories of users.

There are three main categories of users when an ML model moves into production. First, we have the data predictions. The third is the end user or customer, who wants to understand the reason behind the model's predictions.

Take for example a scenario where an algorithm is used by a financial institution to approve loans. In such a case, a data scientist first creates the model. Before using the model, its accuracy and effectiveness are tested after consultations with the subject matter expert. Thereafter, the loan officer makes the final decision of whether



to admit or deny the loan based on the result produced by the model. The individual then wants to know how the decision was made.

Fig. 2. Schematic of the explainable AI framework

Think about the application of an ML algorithm as a loan approval system in the banking industry. First, a data scientist constructs an ML model. Before deploying the ML model, there are many ways of applying explainable AI techniques to understand the behavior of the ML model globally, such as SHAP [6] [7] and BRCG [8]. As far as the analyst is concerned, Protodash [9] provides a tool by which she or he can obtain similar training examples and study the differences and similarities among them and other pertinent information. As far as the user is concerned, some possible ways of achieving his/her goal include using LIME [10][11], SHAP, and CEM [12].

2. Problem Statement

With the increasing number of internet users and advanced technological tools available, there has been an increase in the rate of cyber-attacks. IDSs have been implemented broadly to detect malicious behavior and protect computer networks from potential security breaches. Modern IDSs employ the application of artificial intelligence and machine learning techniques to improve accuracy and help in recognizing new attacks.

On the other hand, many AI-IDS models function as black boxes, giving results or predictions without explaining the process involved in arriving at such predictions. In most cases, security experts find it difficult to comprehend the reasons for classifying certain activities on the network as threats or harmless behaviors. This makes it challenging for security professionals to make decisions and implement effective actions.

In order to make decisions regarding attacks within cyberspace, it becomes imperative to explain the reasons behind attacks. As such, there is a requirement for implementing an XAI solution that provides clear and easily understandable interpretations of AI-based intrusion detection operations.

This paper aims to propose an XAI framework to be used in IDSs.

3. Proposed Framework

According to the objectives of Explainable AI for Intrusion Detection System, the framework should be designed to detect potential cyber attacks. Moreover, there should be a possibility for the end user or administrator to understand the process of making decisions within an intrusion detection framework.

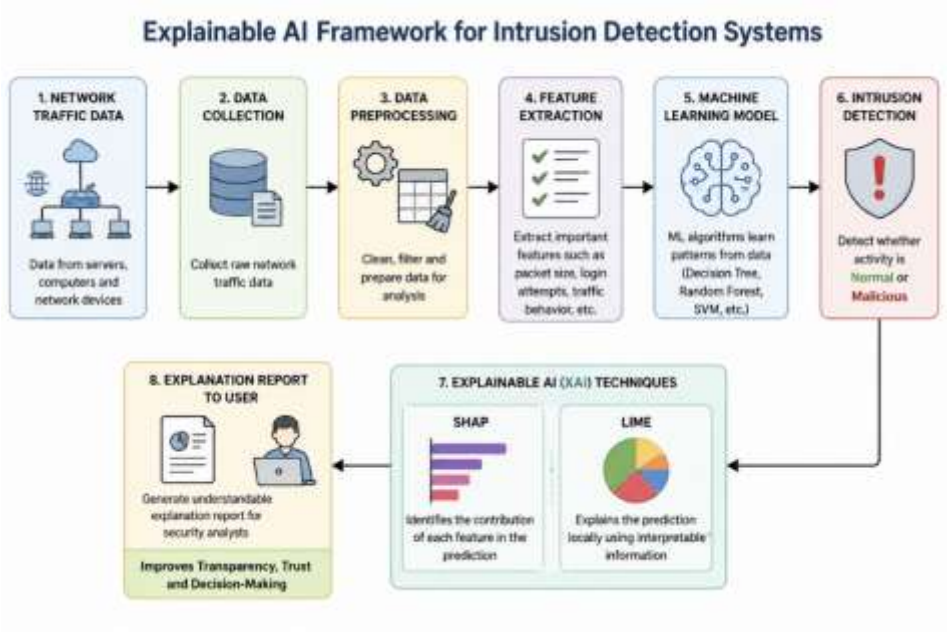
Initially, the AI framework collects network traffic data from multiple sources including servers, workstations, and communication networks. After that, the collected data is preprocessed to remove excessive information

from it and make data ready for further analysis. As a result of data preprocessing stage, there is a need to extract significant features including packet sizes, number of login attempts, network traffic pattern, and connectivity features.

Then, the data processed in the previous stage is passed to a machine learning algorithm used for intrusion detection purposes. Machine learning algorithm examines network traffic and detects abnormal activity or an attempted cyber-attack. Some machine learning algorithms can be considered as appropriate ones in this situation, namely, Decision Trees, Random Forest, or Support Vector Machines.

Next, Explainable Artificial Intelligence techniques should be applied to analyze why a certain prediction is made by the algorithm. Some XAI methods can be applied at this stage such as SHAP and LIME. They help to find out which features contributed the most to the decision and make a conclusion about its nature.

Finally, the output explanation of the intrusion detection system will be generated.



4. Methodology

The methodology used in this research describes how the Explainable AI Framework was developed to improve cyberattack detection and provide explanations for the predicted attacks. Explainable Artificial Intelligence combines machine learning methods with AI to create better cybersecurity tools.

First, the data collection process takes place. The data will be collected from various sources such as servers, workstations, routers, or even wider communication networks. Traditional cybersecurity datasets like NSL-KDD and CICIDS can be used in this stage too.

Secondly, the pre-processing step takes place. During pre-processing, extra data and duplicates are removed from the dataset to increase its quality. Moreover, missing values are eliminated and the data is converted to the format required for AI models. Pre-processing helps to increase the accuracy and efficiency of the AI.

Thirdly, features relevant to detecting cyberattacks are selected during the feature extraction phase. These could be factors like packet size, number of logins, traffic volume, protocol type, connection duration, etc. By extracting the features, AI focuses on the right information for detection.

Once feature selection takes place, machine learning algorithms can be implemented to classify the network data as normal or malicious. For example, decision tree, random forest, SVM or neural networks can be used for detecting intrusions.

Then, after detecting an intrusion, the use of XAI can begin. The purpose of XAI is to explain the model's predictions in order to make them more understandable to humans. Techniques like SHAP or LIME allow us to see which features were used in predicting a cyberattack.

Finally, the AI explains why a particular action is categorized as malicious or normal. This is achieved via the creation of a report, which explains the reasoning behind the model's predictions.

5. Explainable AI Techniques

XAI techniques are used to make the operation of AI-based intrusion detection systems more transparent and comprehensible. XAI methods help security specialists to know why a certain machine learning model classified network activities as normal and malicious. This makes XAI beneficial for building reliable, trustworthy, and effective cybersecurity infrastructure.

The problem with conventional approaches to machine learning is that predictions made with their help often do not include clear explanations, which causes confusion and distrust towards AI-powered tools. Explainable AI solves this issue and identifies the features driving the predictive decision.

The two most common Explainable AI methods are SHAP and LIME.

SHAP (SHapley Additive exPlanations)

SHAP is one of the explainability techniques helping to understand the influence of certain features on the prediction made. SHAP reveals the most impactful features that play a big role when detecting cyber threats. Examples might be peculiarities of login activity, anomalies in network traffic or different sizes of packets sent via the Internet connection. In addition to that, SHAP provides both positive and negative feature contributions to the final prediction.

LIME (Local Interpretable Model-Agnostic Explanations)

LIME is yet another technique aiming to explain certain predictions and how they were made. It produces a comprehensible interpretation around each particular prediction to show security specialists what happened. With the help of LIME, analysts learn why a certain activity was considered malicious.

To summarize, SHAP and LIME are helpful techniques that enhance the interpretability of intrusion detection systems and improve users' experience with them.

6. Advantages of the Proposed Framework

There are a number of benefits to be achieved by the Explainable AI Framework for Intrusion Detection Systems in terms of cybersecurity and protection of the network. Artificial Intelligence coupled with the use of Explainable AI brings an increase in trustworthiness, transparency, and performance to cyber attack detection.

One key benefit to this system is increased transparency. Traditional AI systems typically work as black boxes which means users cannot determine how decisions are reached. This system however can easily identify why certain decisions have been made and can offer explanations for this information, this is making the system more user friendly.

Another key benefit to this system is increased trust in the intrusion detection system. Security analysts, network administrators or whichever other users it may be, will gain a greater sense of trust in the AI model when it comes to its ability to define what network activity is considered malicious or normal.

Improved decision making is also a key benefit to security analysts. Understanding the context behind any discovered attack will lead to an increased likelihood of timely and accurate responses to these threats and reduce confusion and confusion within the network security environment.

The identification of the critical features which causes the cyber attack is another key feature of the proposed system. Explainable AI techniques such as SHAP and LIME will outline the various factors and criteria contributing to these decisions.

Increased efficiency in the time it takes to perform threat analysis is another benefit that this system can offer, as automatically generated explanations may allow security analysts to focus more on an attack rather than defining it.

Therefore this system allows a greater transparency, reliability, trust, and performance in AI-based intrusion detection systems.

7. Challenges and Limitations

While the Explainable AI Framework for Intrusion Detection Systems presents a number of benefits, there are also a number of challenges and limitations in adopting this system. The problems identified can affect system performance, speed and accuracy in the field of cyber security.

A challenge encountered with AI and Explainable AI is that it requires a substantial amount of computation and memory, which leads to high cost and large processing times, as these techniques are complex and difficult to compute. This problem can grow exponentially when processing network traffic and data sets that grow exponentially on a second-by-second basis.

There is also the issue that in a real world application of such system in a cyber security environment there will be huge volumes of data to analyze on a per-second basis and to explain in this time. This problem leads to a slow system that is unlikely to succeed in preventing a threat in time.

The quality of data provided can also influence system performance, the accuracy of a training set can impact the efficiency and detection capabilities of a machine learning system. Poor quality, out of date and skewed datasets can impact prediction rates of a given model.

The Explainable AI may result in explanations that cannot be understood by a non technical user, while increasing transparency within AI it still requires technical and security expertise.

Explainability must be balanced with performance, some very high performance models that have proven accurate (for instance, a neural network) tend to be extremely hard to explain.

This fact is a hurdle for the application of many high performance models within the framework due to limitations and the need to interpret their behavior. The existence of rapidly changing attacks means that explanations need to keep up with current threats.

However, the importance of explainable AI will only increase as systems are increasingly expected to be transparent.

8. Analysis of the Dataset

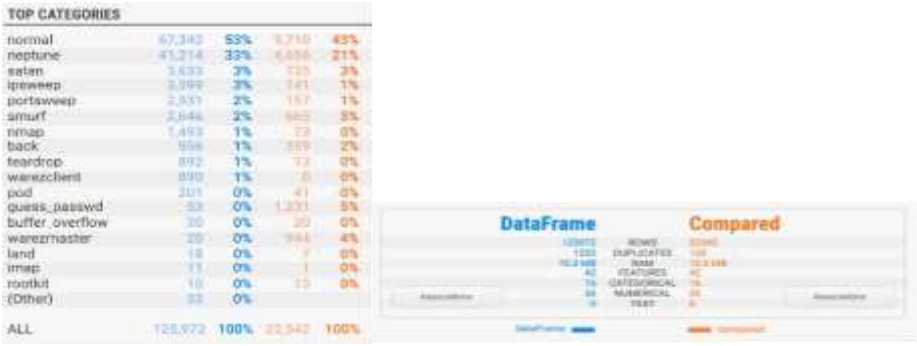


Fig. 3. Basic overview and attack distribution in NSL-KDD dataset

IDS has various data sets [13]. One such widely used data set is NSL-KDD [14], [15], in which there are various types of network intrusion attacks, and these attacks have been labeled manually after analyzing their characteristics through a network flow analysis tool. The comparison between training (DataFrame) and test (Compared) datasets has been depicted in Figure 3.

9. Literature Review

The distribution of algorithms and main techniques used for intrusion detection research is presented in Figure 1. Based on the most papers we have counted 68 of primary studies. In this research work different classification approaches were examined [24]: The majority of the experiments were run using classification approaches, which represented 81% of total experiments. The remaining approach was mainly clustering, dataset analysis techniques, prediction, estimation (3%), association, statistical analysis. They have observed that the experiments performed using public datasets represented 79% of the experiments and those using private datasets were 21%, which were compared in the studies [25].A comparison of 18 different techniques for IDS has been presented and the best 6 algorithms were the followings: Support Vector Machine, Deep Neural Network, Random Forest, Naive Bayes, Decision Tree and K- nearest neighbor. Additionally, other researchers have proposed some new and different approaches. There were boosting algorithms, ensemble combined feature selection algorithms, machine learning approaches, etc which are all Ensemble methods used to boost up the accuracy of a ML classifier applied on IDS shown in Table 1. The authors, Peddabachigari et al. [26] built intrusion detection systems using hybrid intelligent systems. They evaluated different new approaches for intrusion detection based on their research, and analyzed performance against the KDD Cup 99 benchmark intrusion dataset. Their works were focused on the data temporal correlation (DT) and sparse maximum likelihood (SVM). After that they developed hybrid DT-SVM and then they built ensemble approach that included individual methods like DT, SVM, DT-SVM. From the presented results it was obvious that DT produced better or the same accuracy as compared with the other three classes Probe, U2R and R2L. In comparison with a direct SVM, a hybrid DT-SVM has offered better or same accuracy compared with direct SVM for all classes. It's only the ensemble approach that resulted best with respect to Probe and R2L classes. The classification of Probe class was found to be 100% accurate in the ensemble approach so that with better base classifiers other classes also achieved 100% accuracy. The conclusion of the authors was to create an individual hierarchical intelligent IDS model to fully take advantages of best individual base classifiers and ensemble. Proposed Ditto: A Robust and Fair System for Machine Learning at the Edge and a scalable solver to manage the resources on this system. Heterogeneous network was defined as an area where performance robust vs attacks vs performance fairness competes on resources. They derived the class of linear problem and showed that they can theoretically guarantee simultaneous robustness and fairness of the Ditto framework and scaled solver. The proposed system has demonstrated competitively against existing customization approaches and offered better accurate and robust models than fairly and robust baseline systems. Mohseni et al. [27] described

two categories of challenges and opportunities in implementing ML safety in the wild and a three prong strategy to approach them which included: limitations of ML in open world settings, comparison to the traditional safety benchmarks and mitigation strategies for enhancing ML safety which are runtime error detection, adaptation and avoidance. These three strategies were employed to address the ML dependability goal to design safely, improve model robustness, accuracy, and increase runtime error detection. They described ML safety as the approach of addressing long-term risks of ML, especially with the focus on cases where general ML capabilities will exceed safety issues for next decade or safety issues will be more complicated.

10. Classification Model

10.1 Preprocessing

The NSL KDD data set contains 42 attributes in total: 3 are categorical, 6 are binary, 23 are discrete and 10 are continuous. We will keep all these features but transform the 3 categorical attributes. One-hot encoding will be applied to convert categorical to binary attributes. Indeed one-hot encoding is preferable to integer encoding when we are dealing with categorical values since the former does not suppose a natural ordering of attributes which may not be present. In such case, algorithm will tend to learn the false order and will be less efficient.

Therefore, we will have 122 features in the new data set, instead of 42.

10.2 Model Architecture

Dataset have two types: Normal and Attack. Any other label than "Normal" is considered as "Attack". Binary Classification Model is being trained using fully connected deep neural networks [5]. It is trained on "KDDTrain+.txt" while "KDDTest+.txt" is being used to evaluate it.
In this experiment fully connected network along with RELU activation function is being used. It has three hidden layers where there are 1024, 768 and 512 neurons. The input dimension is 122 and the output dimension is 2. The two output neurons are better than one to describe SHAP explanation. The structure of the Classifier is-[122,1024,768,512,2]. Network intrusion detection model has been implemented using Keras library.
Adam optimizer is used along with Learning Rate=0.01 and Number of Epochs=100. Dropout is also used in the network. The above mentioned classifier works fine. The results are given below..

Accuracy	Precision	Recall	F1
0.824	0.964	0.713	0.820

Table 1. Results on KDDTest+ dataset

11. Significance of the Study

This project has applied the training to the intrusion detection system using the CICIDS-2017 dataset. Proper feature selection is the most important step in intrusion detection. According to the related studies, the characteristics needed to distinguish an attack can be required, required to a certain extent, not required or only required for some attacks. The CICIDS-2017 dataset comprises of 76 features, used to train and test IDS. From these, we selected only 10 important features of the CICIDS-2017 dataset, so as to increase classification accuracy. Explanations have proven to be helpful to both experts and non-experts in making a choice between competing models of trust, to improve untrustworthy models, and in achieving deeper understanding of predictions in the text domain. Therefore, the authors have generated LIME (local interpretable model-agnostic explanations) explanations for DT, RF and SVM, after using the machine learning models.

12. Generating Explanations

The Explainable AI have generated a series of hot research topics in the data science community in recent years. As one of the hottest research topics of today's world, many tools and libraries have been generated every day to clarify the "black box" models. However, today there are no standard performance measure to compare and evaluate methods and no "best" model explaination for machine learning models. In fact, there are several methods for generating machine learning models explanation [18], whether model-specific and model-agnostic, local and global, intrinsic and post-hoc, and so on. Hence, we adopted multiple model explaination methods [19] in our research.

LIME can be utilized to explain an individual instance through a local linear approximation of the behavior of model. Regarding the decision function of model, we know it may not be simple globally, while if we are focused on model's decision function around that instance, it would be easier, since the model's decision function can be approximated by perturbations of the sample. A linear model can be easily fitted around perturbed samples, we will then obtain some insight of model locally.

The logic of using SHAP is based on game theory, which can be used for explaining locally as well as globally. There are game and players within game theory, and here our game is to mimic the result of our model while player would be features trained in our ML model. SHAP represents how each feature contributes to model's prediction. To measure feature importance, it evaluates results obtained by using each combinations of features. In case of 'n' features, SHAP will train '2n' models. While datasets would be the same, features would differ.
An open-source toolkit for interpretability and explainability of ML models and data developed by IBM researchers is known as AI Explainability 360 (AIX 360) [20]. It contains 8 different explainability algorithms. We select 3 explainability algorithms in explaining deep neural networks due to their usage at different steps in AI modeling processes.
To explain model's predictions based on training data, ProtoDash provides exemplar-based explanations of data summarization and the prediction made by the model. BRCG algorithm trains the interpretable model using a supervised algorithm to perform binary classification. It can learn Boolean rules from the data which either 'OR' of 'AND' rules or 'AND' of 'OR' rules. CEM algorithm can be used to explain the predictions locally made by the model.

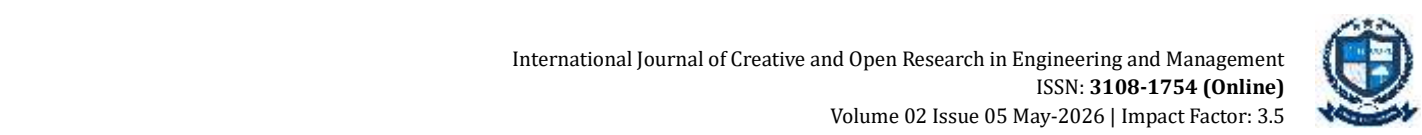
.

Algorithm	ProtoDash	Boolean Decision Rules via Column Generation (BRCG)	Contrastive Explanation Method (CEM)
Use	Explanations fr training data	Train interpretable Model	Local post-explanations

Table 2. AIX 360 algorithms

13. Results

SHAP Summary plot: It is a global explanation of the model incorporating the impacts of features with their importance. A SHAP Summary plot visualizes a Shapley value for a feature and a particular observation with a single dot. Features are plotted on the Y-axis and SHAP values on the X-axis, where the color represents the low



and high value, and features from top being the most important at the top to the bottom being the least important.

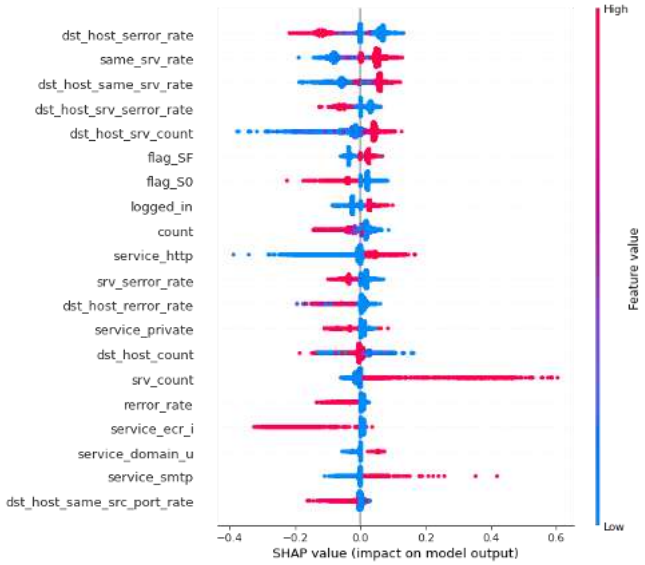


Fig. 4. SHAP summary plot – high value of 'same_srv_rate' increases the probability of attack whereas high value of 'dst_host_serror_rate' decreases the probability of attack

Global explanations based on SHAP is obtained on the whole/ part of the dataset. The local explanation based on SHAP is on one data point at a time and that explains the values of the feature that are contributing for positive decision and contributing for negative decision. Figure 5 is the example of local explanation where the probability of attack is 1.00 and features along with the value, are 'dsthostsamesrvrate', 'samesrvrate', 'service_private' and so on. Features in red color increase prediction and features in blue color decrease the prediction.

Fig. 5. SHAP force plot used for local explanations to explain a particular instance where output probability of 'attack' is 1 and shows features contributing in decision

Figure 6 shown below is the SHAP Force Plot for a bunch of points from the testing data set. The force plot for 50 points from each class (i.e., normal and 3 attack types) is shown below. This figure is actually obtained by rotating all the individual force plots of a single point (as in figure 5) at 90-degree angles and placing them side-by-side. There is clear distinction between the different attack types from the explanations provided..

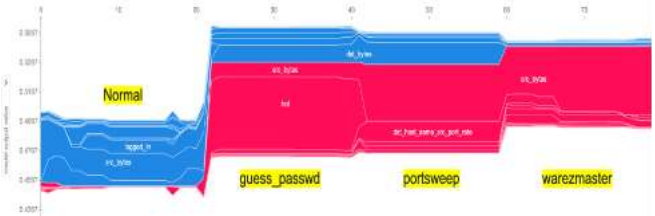
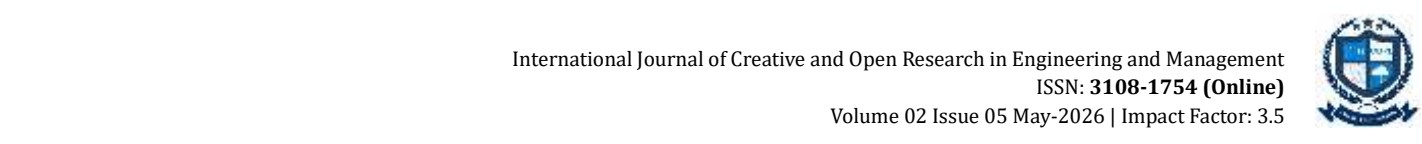


Fig. 6. SHAP force plot for 4 types of data points in NSL-KDD dataset

From the rules we see how the impacting factors vary across the data points in the dataset.
The DNN [5] is trained in such a way to predict whether the network traffic is 'normal' or 'attack'. The behavior of the model can be simplified to some rules. To generate these rules, we used BRCG and generated the following rules from the dataset.
Rule for predicting Y=1(Attack), if any rule is satisfied, else Y=0(Normal):



wrong_fragment>0.00
srcbytes<=0.00 AND dsthostdiffsrv_rate>0.01
dsthostcount<=0.04 AND protocoltypeicmp
numcompromised>0.00 AND dsthostsamesrv_rate>0.98
srvcount>0.00 AND protocoltypeicmp AND serviceurp_i NOT

The performance of the BRCG algorithm is also notable. The performance parameters are:
Accuracy for Training Data = 0.9823
Accuracy for Test Data = 0.7950 This shows that even without employing the network traffic, it is possible to achieve ~80% of accuracy.

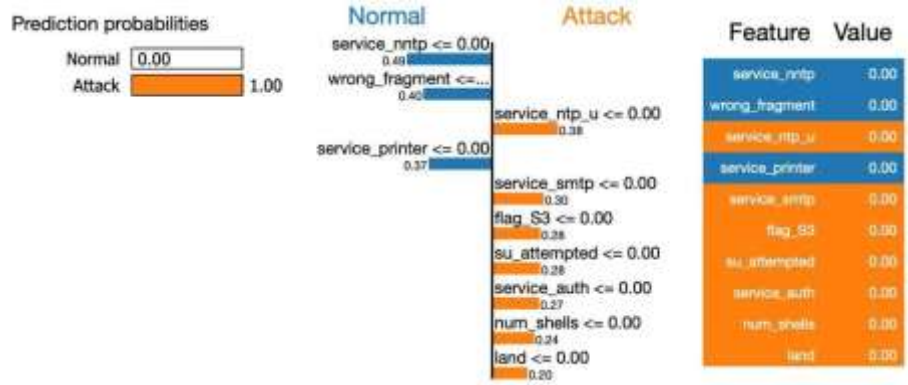


Fig. 7. Explaining individual prediction of deep learning classifier using LIME

LIME provides local explanations. Figure 7 below shows how explanations can be used to identify whether the output of the classifier would be “Normal” or “Attack”. The color scheme has been used to identify which features contribute to which output classes. The features colored in orange contribute to “attack” and those in blue color contribute to “normal” class.

	0	1	2	3	4
duration	0	0	0	0	0
src_bytes	0	7.47846e-07	0	0	0
dst_bytes	0	0	0	0	0
land	0	0	0	0	0
wrong_fragment	0	0	0	0	0
...	...	...	...	...	...
flag_S3	0	0	0	0	0
flag_SF	0	1	0	0	0
flag_SH	0	0	0	0	0
Class	Attack	Attack	Attack	Attack	Attack
Weight	0.935025	3.00021e-05	0.000150011	0.0255018	0.0392928

	0	1	2	3	4
duration	1.0	1.00	1.0	1.0	1.0
src_bytes	1.0	0.08	1.0	1.0	1.0
dst_bytes	1.0	1.00	1.0	1.0	1.0
land	1.0	1.00	1.0	1.0	1.0
wrong_fragment	1.0	1.00	1.0	1.0	1.0
...	...	...	...	...	...
flag_S1	1.0	1.00	1.0	1.0	1.0
flag_S2	1.0	1.00	1.0	1.0	1.0
flag_S3	1.0	1.00	1.0	1.0	1.0
flag_SF	1.0	0.08	1.0	1.0	1.0
flag_SH	1.0	1.00	1.0	1.0	1.0

Table 3a. Similar instances predicted as attack **3b.** Use of weights to show similarity

The end-user taking the final decision based on the prediction of the model will get to know why the model predicted in that way, if we give the same test example (taken from train dataset somehow) in the train dataset. We assumed our first test example to be an attack for which the model has predicted attack. The similar examples present in the train dataset are shown in Table 3a, the higher the weight value, the more it is similar.

Table 3b contains human readable explanations which are based on weight of the features. Table 3a and Table 3b shows the top 5 closest examples to the test example. Observing the weights we can see the 0th column is most similar, since the value is 0.93. Both of these two tables will provide great ease to the analyst to make the final decision with more accuracy.

ML models need to be completely transparent to the end-users and they must have answers to all of the queries such as Why did the model predict so, which features lead to this kind of decision, with what changes the decision can be reversed etc. All these issues can be addressed by CEM algorithm.
With respect to the one where the prediction was "normal" we analyzed, CEM provides us with a set of methods through which the decision can be modified by making minimal changes to the values of the features. CEM can also show us what minimum set of features and corresponding values have to be invariant for the prediction to be invariant with respect to the model. By analyzing the output of statistics from the CEM algorithm on a sample pool of applicants, insight on what minimum set of features is of importance.

```
Sample: 2
prediction(X) [[1.00e+00 3.15e-10]] Normal
prediction(Xpn) [[0.18 0.82]] Attack
```

	X	X_PN	(X_PN - X)
duration	3.4653e-05	0.02	0.02
hot	0	0.12	0.12
dst_host_serror_rate	0	0.03	0.03
Class	Normal	Attack	NIL

```
PP for Sample: 5
Prediction(Xpp) : Normal
Prediction probabilities for Xpp: [[0.55 0.45]]
```

	X	X_PP
num_root	0	0.02
count	0.00782779	0.03
srv_count	0.00782779	0.06
diff_srv_rate	0	0.02
dst_host_count	0.607843	0.01
Class	Normal	Normal

Table 4. Pertinent negative and pertinent positives for an instance

From Table 1 below, it is observed that when we alter the values of only three attributes viz duration, hot, dst_host_serror_rate from 0, 0, 0 to 0.02, 0.12, 0.03 respectively, while maintaining the rest feature values intact, then the class label predicted by the classifier changes from ‘normal’ to ‘attack’. From Table 2, minimum values of five features namely num_root, count, srv_count, diff_srv_rate and dst_host_count, needed to maintain the same classification prediction of the classifier, can be noted.

14. Future Scope

Future application scope of Explainable AI in Intrusion Detection System is too wide because with the increased use of digital technologies and internet in today's world, cyber-threats are exponentially increasing with the pace of time. In the coming years Explainable AI can perform well to develop more intelligent, fast and reliable security systems.

One of the most promising future advancement could be the development of real-time explainable intrusion detection system where the future frameworks could be developed to perform analysis of network traffic in real-time to provide real-time explanation about the detected cyber attack. In this manner the security analysts can respond the threats in a more effective way.

Explainable AI can be integrated with further technologies like Cloud Computing, IoT devices, Smart devices etc. It's obvious that with the increased number of cloud services and IoT devices, the associated cybersecurity risks would also increase. Future explainable AI frameworks can be developed to improve the security of the technologies by detecting and providing explanations for the unusual activities more efficiently.
There is also scope for developing deep learning models along with the Explainable AI techniques. Deep learning algorithms improve the attack detection capabilities and at the same time Explainable AI makes it easier for the user to understand the predictions that are being made by complex deep learning models thus establishing more understandable and powerful intrusion detection system.

The future work also involves reducing the computational complexity and improving the speed and efficiency of popular Explainable AI methods like SHAP, LIME so that they can work effectively with large scale intrusion detection systems.
The user-friendly and visual explanations will also be provided to the non-expert users or organizations so that they can understand what the cyber threats and decision.
In short, future of explainable AI in cyber-security looks promising as it will establish greater trust in AI systems and will aid in building an intelligent intrusion detection system.

15. Conclusion

Currently, using ML techniques for intrusion detection system is less trusted, as it is not known which and why features have been selected. The proposed framework explains by extracting features through local as well as global explanation and establishes the relation of feature to model prediction. Through explanation one can identify the pattern learned by the model, if pattern is incorrect then other features can be chosen and data set is adjusted to improve learning of model.

Network analyst will then make a final decision with help of prediction made by the model. Besides predictions, explanations are also provided, such that instances that are similar to test instance will be explained to support the decision of network analyst. To end-user, explanations enable them to understand which feature contributed in making the decision and to what extent. They can also change the values of features to manipulate the prediction of model.

Our framework thus provides an explanation at every level of ML pipeline which would suit the various user group of intrusion detection system.

References

1. C. S. W. M. M. Daniel L. Marino, "An Adversarial Approach for Explainable AI in Intrusion Detection Systems," in IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, 2018, pp. 3237-3243, doi: 10.1109/IECON.2018.8591457.
2. K. Z. Y. Y. X. W. Maonan Wang, "An Explainable Machine Learning Framework for Intrusion Detection System," IEEE Access , vol. 8, pp. 73127 - 73141, 16 April 2020.
3. D. Gunning, "Explainable Artificial Intelligence (XAI)," DARPA/I2O, November 2017.
4. [Online]. Available: https://www.darpa.mil/attachments/XAIProgramUpdate.pdf.
5. G. S. Maheshkumar Sabhnani, "KDD Feature Set Complaint Heuristic Rules for R2L Attack Detection," in Proceedings of the International Conference on Security and Management, SAM '03, June 23 - 26, 2003, Las Vegas, Nevada, USA, Volume 1
6. R. V. K. S. P. P. Rahul K. Vigneswaran, "Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security," in In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 16). IEEE., 2018.
7. S. Lundberg, "SHAP," 2020. [Online]. Available: https://github.com/slundberg/shap.
8. S.-l. l. Scott M. Lundberg, "A Unified Approach to Interpreting Model Predictions," in Advances in Neural Information Processing Systems 30 , 2017.
9. O. G. D. W. Sanjeeb Dash, "Boolean Decision Rules via Column Generation," in Advances in Neural Information Processing Systems 31, 2018.
10. A. D. G. C. C. A. Karthik S. Gurumoorthy, "Efficient Data Representation by Selecting Prototypes with Importance Weights," in International Conference on Data Mining (ICDM), 2019.

11. S. S. C. G. Marco Tulio Ribeiro, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016.
12. M. T. C. Ribeiro, "LIME," 2020. [Online]. Available: https://github.com/marcotcr/lime 12. P.-Y. C. R. L. C.-C. T. P. T. K. S. P. D. Amit Dhurandhar, "Explanations based on the Missing: Towards Contrastive Explanations with Pertinent Negatives," in Advances in Neural Information Processing Systems 31, 2018.
13. S. S. S. K. J. Santosh Kumar Sahu, "A detail analysis on intrusion detection datasets," in 2014 IEEE International Advance Computing Conference (IACC), Gurgaon, India, 21-22 Feb. 2014.
14. "NSL_KDD," 31 July 2015. [Online]. Available: https://github.com/defcom17/NSL_KDD.
15. M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorban, "A detailed analysis of the KDD CUP 99 data set," in ubmitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009
16. "SweetViz," 12 August 2020. [Online]. Available: https://github.com/fbdesignpro/sweetviz.
17. S. Loukas, "Everything you need to know about Min-Max normalization: A Python tutorial," 28 May 2020. [Online]. Available: https://towardsdatascience.com/everything-you-needtoknow-about-min-max-normalization-in-python-b79592732b79.
18. C. Molnar, Interpretable Machine Learning- A Guide for Making Black Box Models Explainable., 2020
19. V. A. a. R. K. E. B. a. P.-Y. C. a. A. D. a. M. H. a. S. C. H. a. S. H. a. Q. V. L. a. R. L. a. A. M. a. S. M. a. P. P. a. R. R. an, "One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques," September 2019.
20. "AI Explainability 360 (v0.2.0)," 2019. [Online]. Available: https://github.com/TrustedAI/AIX360.

21. Lundberg, S. Shap vs. Lime. 2019. Available online: https://github.com/slundberg/shap/issues/19 (accessed on 17 July 2022).

22. Wang, M.; Zheng, K.; Yang, Y.; Wang, X. An explainable machine learning framework for intrusion detection systems. IEEE Access 2020, 8, 73127–73141. [CrossRef]

23. Vigneswaran, R.K.; Vinayakumar, R.; Soman, K.; Poornachandran, P. Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security. In Proceedings of the 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India, 10–12 July 2018; pp. 1–6.

24. Tran, M.-Q.; Elsisi, M.; Liu, M.-K.; Vu, V.Q.; Mahmoud, K.; Darwish, M.M.F.; Abdelaziz, A.Y.; Lehtonen, M. Reliable deep learning and IoT-based monitoring system for secure computer numerical control machines against cyber-attacks with experimental verification. IEEE Access 2022, 10, 23186–23197. [CrossRef]

25. Scott, S.-l.l.; Lundberg, M. A unified approach to interpreting model predictions. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.

26. Ribeiro, M.T.C. Lime. 2020. Available online: https://github.com/marcotcr/lime (accessed on 17 July 2022).

27. Sahu, S.K.; Sarangi, S.; Jena, S.K. A detail analysis on intrusion detection datasets. In Proceedings of the 2014 IEEE International Advance Computing Conference (IACC), Gurgaon, India, 21–22 February 2014.

28. Ando, S. Interpreting Random Forests. 2019. Available online: http://blog.datadive.net/interpreting-random-forests/ (accessed on 17 July 2022)

29. Dong, B.; Wang, X. Comparison deep learning method to traditional methods using for network intrusion detection. In Proceedings of the 8th IEEE International Conference on Communication Software and Networks (ICCSN), Beijing, China, 4–6 June 2016; pp. 581–585.

30. Islam, S.R.; Eberle, W.; Bundy, S.; Ghafoor, S.K. Infusing domain knowledge in ai-based ”black box” models for better explainability with application in bankruptcy prediction. arXiv 2019, arXiv:1905.11474