International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

An Improved Explainable Artificial Intelligence for Intrusion Detection System in Cloud Environment

Rejna Azeez Nazeema*[1], Shama Kouser[2], Samar Mansour Hassen[3], Nagla Babikar[4], Mawahib Sharafeldin Adam Boush[5]

Submitted: 27/01/2024 **Revised**: 05/03/2024 **Accepted**: 13/03/2024

Abstract: Automated Anomaly Detection (AD) systems that can spot suspicious activities are perfect for cloud computing applications. Previous research has not solved the open challenge of irregularity discovery in cloud computing. The usual operations of a cloud server must be characterised, malicious anomalies must be distinguished from benign ones, and false alarms must be avoided at all costs to prevent alert fatigue. Various industrial applications showcase the growing potential of the cloud infrastructure. The Network Intrusion Detection System (NIDS) is seen as a crucial part of data transmission security, which seems to be in jeopardy. Upcoming advances in intellectual IDS have made use of Explainable Artificial Intelligence (XAI) methods. The bulk of IDS is based on either supervised or uncontrolled XAI methods. The lowered model's capacity to identify attack patterns is diminished due to the fact that NIDS relies on supervised learning using labelled data. Also, the unsupervised model can't deliver a reasonable result. Later, to enhance the performance of unsupervised learning, a high-quality feature set will be selected using an effective feature selection technique. Finally, a Machine Learning (ML) classifier based on XAI will be used for classification. The optimal generalizability capacity for data training is shown by this strategy. Analyses are also performed on the unlabeled data. In this step, we filter away the dataset's noisy and redundant samples. The anticipated technique outperforms other existing methods in terms of accuracy, according to the experimental data.

Keywords: *NIDS, Dataset, Deep Learning (DL), XAI, Intrusion Detection*

1. Introduction

When it comes to keeping people engaged with online services like social networks and web search, performance and availability are two of the most important criteria. Anomalies caused by exogenic and/or endogenic factors make it difficult to provide high availability with great performance. Due to the large number of services and metrics used by most service-oriented architectures (SOAs), automatic AD is challenging [1]. The detection of malicious behaviour on separate nodes gets increasingly challenging when the use of numerous services and apps in cloud computing environments expands dramatically due to the diversity of traffic patterns created by these applications. When it comes to datasets that only belong to one class, most of the existing approaches that have been published in the literature have serious limitations [2]. Throughout the development of the monitoring system, it uncovered best practices, implementation recommendations, and design solutions. Anomaly detectors for complicated systems might be built using them by other researchers and professionals [3].

In order to evaluate the efficacy of existing detection methods, the experiment was conducted using several application monitoring data sets. According to the results, these strategies' efficacy on different types of data is dependent on the specific data elements they aim to address. Because of this, it is difficult to predict anomalies and improve detection robustness and accuracy using existing AD techniques. To achieve these three goals, an ELBD architecture was developed that leverages state-of-the-art detection techniques that were hand-picked. With the three classic linear ensemble methods—weighted average, average, and maximum—plus a deep ensemble technique, the framework is complete. While the ELBD framework accomplishes improved detection robustness as well as accuracy, tests show that the deep ensemble method may provide the best trustworthy along with precise detection for cloud applications [4].

The NIDS is seen as a crucial part of data transmission security, which seems to be in jeopardy. More recently,

[1]*Department of Computer Science, Jazan University, College of Computer Science and Information Technology, Jazan, Kingdom of Saudi Arabia. razeez@jazanu.edu.sa*
[2]*Department of Computer Science, Jazan University, College of Computer Science and Information Technology, Jazan, Saudi Arabia. skouser@jazanu.edu.sa*
[3]*Department: Management Information Systems, Jazan University, College of Business Administration, Jazan, Saudi Arabia. shassen@jazanu.edu.sa*
[4]*Department of Computer Science, Jazan University, College of Computer Science and Information Technology, Jazan, Saudi Arabia. nbabiker@jazanu.edu.sa*
[5]*Department of Computer Science, Jazan University, College of Computer Science and Information Technology, Jazan, Saudi Arabia. mboush@jazanu.edu.*
Corresponding Author Email: author@email.com

intelligent IDS have been developed with the use of ML techniques. Supervised as well as unsupervised ML techniques are the backbone of most IDS. The decreased model's efficacy to detect attack patterns is diminished due to the fact that NIDS relies on labelled data in supervised learning. Also, the unsupervised model can't offer a suitable result. Later, to enhance the performance of unsupervised learning, an effective auto-encoder is used for feature selection, allowing for the identification of acceptable features [5].

A denial-of-service (DOS) attack is one that overwhelms the target system's resources to the point that it can't handle any more requests. The term "Remote to Local" (R2L) refers to an assault that intrudes on a local machine from a remote machine. A way for unauthorised users to get root capabilities is known as User-to-Root (U2R). Attackers using this technique can get access to a system via a regular user account, but they are really trying to exploit security holes to get root or administrator rights [6].

Cloud-based AD(CAD) is an approach that identifying anomalous cloud activity. An important component of CAD for multiclass anomaly classification is the convolutional neural network long short-term memory (CNN-LSTM). For the purpose of binary anomaly classification, the ensemble ML(EML) model is used. The presentation of CAD in binary AD as well as multiclass anomaly classification is evaluated using a high-quality dataset from UNSW. Other conventional ML models along with state-of-the-art have also been compared to CAD [7].

However, existing data stream management systems are unable to adequately examine the network streams in order to detect anomalies in real-time. The AD algorithms are useless here due to their computational complexity, high false-positive rates, and inapplicability to networks. Accordingly, this a hybrid data processing approach for network AD that uses both convolutional neural networks (CNNs) and grey wolf optimisation [8]. It proposes TopoMAD, a stochastic seq2seq model that successfully mimics the interdependence of polluted data over space and time. They use sliding windows over continuously acquired measurements to show how things change over time, and they use system topological information to arrange metrics from different parts. The geographical data is extracted using graph neural networks, while the temporal data is extracted using long short-term memory networks. If you want model to be able to withstand training on corrupted data, you should use a variational auto-encoder [9].

Improving the performance of supervised classifiers effectively relies heavily on feature detection. Feature extraction and feature selection are the two types of approaches. If you have an objective function in mind, feature subset selection may provide the best performance by removing irrelevant or unnecessary features. Feature selection approaches, according to several studies, overcome the "dimensionality curse" and outperform NIDS in terms of detection. Feature extraction-based mapping turns unique features with more dimensions into features with fewer dimensions, making new non-linear as well as linear combinations of those features. There has been recent evidence of DL technology combined with resourceful IDS for feature extraction from a variety of researchers. Extracting useful structures from raw data along with feeding them into the classifier for attack detection is done automatically using ML approaches.

The research study has made the following important contributions:

1. Begin by obtaining the dataset from the readily available web site. The NSL-KDD dataset is used in this study for the purpose of threat prediction.
2. The subsequent step is to do preprocessing and normalization to clean the dataset.

3. The next step is to examine the dataset for the most influential features; this danger might have a significant effect on the application.

3. The last step is to use the ML based classifier model to calculate the prediction accuracy. Predicting the accuracy of the classifier is therefore accomplished using the XAI classifier. Python is used to run the simulation and measure things like accuracy, precision, f1 score and recall.

Below is the outline for the remainder of the article. Using a variety of methods, Section 2 summarises previous work on the topic of IDS. Section 3 defines the proposed method, while Section 4 discourses its outcomes. The references come after Section 5, which determines the work.

2. Literature Review

Garg et al. [10] improved in exploration, exploitation, and initial population formation were introduced to the GWO and CNN learning techniques. For the model to be more practical, the dropout functionality was also revamped. These updated versions are referred to as Improved-GWO (ImGWO) as well as Improved-CNN (ImCNN), respectively. In two steps, the it successfully identifies abnormalities in the network. In the first stage, ImGWO is used for feature selection with the aim of attaining the optimal compromise between two objectives: minimising the feature set along with lowering the error rate. As a second step, ImCNN is used to classify network anomalies. The efficacy of the model is validated by means of synthetic and benchmark datasets (KDD'99 and DARPA'98).

He et al. [11] discuss into robustly model temporal and geographical relationships among contaminated data, TopoMAD employs a stochastic seq2seq approach. To capture the temporal dependence, it uses sliding windows over continuously acquired measurements and uses system topological information to arrange metrics from dissimilar components. The geographical data is extracted using graph neural networks, while the temporal data is extracted using long short-term memory networks. To make it even more resilient to training on corrupted data, construct the model using a variational auto-encoder.

Thakkar et al. [12] preventing these issues from negatively impacting cloud users should be top priority. Using intrusion detection systems, or IDSs, has become the standard method for finding cloud security risks. More and more focused on learning-based techniques for security applications since the introduction of ML. One state-of-the-art approach to cloud threat detection is DL. Unfortunately, the detection accuracy as well as false-positive rate of cloud IDSs as they are right now are rather low.

Hagemann, Tanja, and Katerina Katsarou. [13] provided a systematic review of 215 articles that may be considered representative of the last decade of this scientific advancement.The paper gives a brief summary of the specific models utilised for AD and outlines three key methodological domains: ML, DL, and statistical techniques.

Chkirbene et al. [14] created a weighted class classification algorithm to solve the problem of imbalanced data and protect the network from malicious nodes. The approach merges a supervised ML algorithm that makes use of past data from network nodes with an especially calculated best-effort iterative strategy to improve the pinpoint accuracy of seldom-discovered attacks. The ML algorithm creates the classifier that distinguishes between the investigated attacks. A private database is where these judgements are stored by the system.

NG, Bhuvaneswari Amma, and S. Selvakumar [15] applied a scaling method for AD in a fog environment using vector convolutional DL (VCDL). The scalability of the AD system depends on the communication's ability to be sent to the nodes in the fog layer for processing. This is captured by the VCDL approach, which leverages fog nodes for computation and spreads IoT traffic training. In the fog layer, the master node distributes the training parameters.

Gulenko et al. [16] find that anomalies and minor misbehaviours of the systems might occasionally precede breakdowns and go undiscovered when outages are the sole item being monitored. Data collected from all levels and components of the cloud architecture is analysed using ML approaches to detect hosts and services exhibiting anomalous behaviour. A host's usual behaviour may be simulated using several techniques, which can subsequently be used for AD during runtime. A thorough offline examination of data from anomaly injection sessions demonstrates the models' accuracy and memorability.

Shakya, Dr. Subarna, and Dr. S. Smys. [17] discuss the data collected from lower levels frequently contains unexpected values that are unusable for the application. These out-of-the-ordinary occurrences in the data are called anomalies. Intentional or unintentional attacks, broken-edge equipment (typically mobile devices, sensors, or actuators), or changes in the surrounding environment may all lead to the disclosure of unexpected data. Maintaining the competence of the network along with application is the objective of eradicating the anomalies.

Meng, Qiuhan, and Songye Zhu [18] suggested a DL framework for AD within the fog paradigm using hardware techniques, a temporal convolutional network and an autoencoder to find problems in building vibration monitoring data without any help from a person.The anomalies were discovered automatically based on the differences in restoration errors between the unique as well as rebuilt signals. The use of an adaptive threshold method that took into account the false as well as missed detections brought on by high variations in vibration signals resulted in the finest documentation presentation. Using the log-likelihood of the restoration errors, this method searched for an optimal coefficient for anomalies.

3. Proposed Methodology

To train the intrusion detection system, this work used the NSL-KDD dataset. Part of intrusion detection is picking the correct features. The NSL-KDD dataset comes with 41 features that may be used for both training and testing IDS. According to these findings, in order to enhance classification accuracy, this research selected just a subset of the NSL-KDD dataset's features.

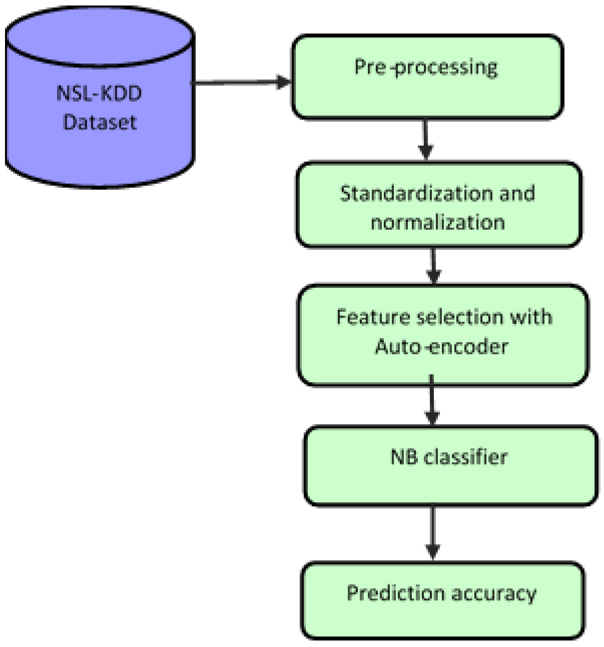

NSL-KDD Dataset
Pre-processing
Standardization and normalization
Feature selection with Auto-encoder
NB classifier
Prediction accuracy

Fig.1. Flow diagram of the proposed model

For both specialists and laypeople, explanations are a lifesaver when it comes to picking trust measurement models, fixing unreliable ones, and understanding text domain predictions. Thus, after the application of the ML model, the generated local interpretable model-agnostic explanations (LIME) observations for Decision Tree (DT), Random Forest (RF), and Logistic Regression (LR) [19]. Figure 1 shows the proposed flow diagram.

3.1. Preprocessing

The preprocessing get input from NSL-KDD dataset. By suitably modifying and scaling the whole dataset, preprocessing in ML primarily aims to speed up the testing as well as training process. An essential part of any ML workflow is data preprocessing before feeding it into an ML algorithm. The features are normalised, and outliers are removed during preprocessing. In addition to enhancing the model's interpretability and accuracy, it also helps to decrease the period as well as assets needed to train the model and avoid overfitting. Cleaning the dataset is a crucial part of preprocessing.

Data cleansing entails identifying and removing any data that is missing, erroneous, incomplete, or superfluous from the dataset. Errors, duplicates, and rows containing null values, empty cells, or values that are not numbers are removed. Especially in datasets recycled for NIDS, class imbalance is a common issue, and ML techniques presume an equitable distribution across classes. Relying on minority class detection is critical when dealing with imbalance. If your method is sensitive to the magnitudes of certain features, you may scale the data using MinMaxScaler. These models may learn better and perform better thanks to the scaling of the data. Then this preprocessed data is sent to normalization stage.

3.2. Normalization

Another typical way to preprocess data in ML is through normalisation. Scaling each column in a dataset to a uniform value is known as normalisation. Size, scope, and unit variability are common features of real-world datasets. To ensure that ML models have a consistent understanding of these features, feature scaling is necessary. The model's performance and stability throughout training are both improved by this. ML does not, however, need to normalise all datasets. When features have different ranges, it is just required [20]. The suggested model's min-max scaler, which allows normalisation, is defined as:

$$X_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (1)$$

This normalized output is then sent to feature selection stage as input.

3.3. Feature Selection

When developing a ML model, feature selection is crucial as it determines the model's efficiency. To ensure that the features do not have outsized values that might affect the outcome, scale them before moving on to the feature selection procedure. Feature scaling causes features to have a standard deviation of one and an average of zero. By removing superfluous or unneeded features from the dataset, feature selection leaves just the necessary data for building the model. The likelihood of any model being overfit are reduced as a result of this. This experiment, use an invariant feature selection approach that extracts the necessary features using an ANOVA F-Test. This approach takes a detailed look at each feature to find out how strong a link there is between the features that have labels. Recursive feature elimination (RFE) is employed to remove all the unnecessary features and retain just the required ones when we acquire the necessary ones. The features that were kept are arranged according to how important and relevant [21]. After feature selection classification is done.

3.4. Classification with LGBM

This study, primarily use LightGBM (LGBM) for ML jobs. One group of well-liked models that uses gradient boosting is LGBM. An ensemble of ineffective learners, often decision trees, is employed in gradient boosting. To put it simply, a weak learner will learn a function that is not very accurate for the task at hand. The aim of each succeeding weak learner is to fix the samples that the one before it messed up. Thus, it seems reasonable that each poor learner tends to lean slightly towards the negative gradient of a certain cost function. Even though our preferred model may provide a clear feature significance score, the way they calculate it gives more weight to features that are either continuous or have many alternative values, such as categorical features, as they are more commonly used in DT branches [22].

Hence, permutation importance is employed for feature removal. We begin by training a model on the training dataset along with an accuracy metric A on the validation dataset. This is how permutation significance works. Then, in order to get A_perm we independently permute each of the features of the validation dataset and assess the accuracy of the model on this altered validation set. The difference between the two accuracy ratings reveals the feature's significance. We decided that the F1-score would be the best measure of permutation relevance for our needs. This is a result of the AWID dataset's class-imbalance problem. Differences in accuracy will be modest due to the tiny number of invasive incidents. We severely penalise features that lead to any noticeable decline in intrusion class identification by calculating the F1-score by considering the intrusive class as the positive class [22]. The feature significance ranking formula is shown in Algorithm 1.

Algorithm 1. Feature Classification Method

```
1. Initialize datasets training as well as validation
2. Build a model using the dataset for training
3. Compute baseline F1-score on validation set, $F_{base}$
4. for each feature of the dataset do
5.     Permute the values of the feature in validation dataset
6.     Create F1-score of this validation dataset, $F_{perm}$
7.     Importance of feature: $F_{base} - F_{perm}$
8. end for
```

Following classification, the LIME model was included in the ML pipeline to enhance the model's explainability. With the help of the change in feature values of a data sample, LIME is able to correctly explain many ML methods for regression predictions. Each article's value is transformed into the predictor's contribution in this way. A local interpreter may provide their insight on each data sample. When accuracy has to be raised, sophisticated approaches like LR, the XGB ML classifier, and the LGBM classifier are often recommended. The model resolves classifier problems after using LIME. The LIME approach sheds light on a black-box ML model by allowing one to manipulate the input data samples and see the impact on predictions. By replicating the performance of the complex model in a different location using the LIME model, we may get insight into the predictions generated by the simple model in one area. You can manage Tableau, text, and image data types using LIME. These findings reveal LIME's analysis of the NSL-KDD data. This is the LIME algorithm.

- An explanation is required if there are n occurrences of interruption with little to no change in value. Using this fake data, LIME builds a local linear model centred on the changed observation.
- The predicted outcomes of the modified data are in sight.
- Determine the distance between the initial observation and each impacted observation. Calculate the degree of similarity by using the distance. Determining the most effective means of representing the revised data predictions in light of the preprocessing data is the next stage.
- A model is fitted to the perturbed data using the appropriate preprocessing data.
- The coefficients, also known as weights, of the fundamental model determine the outcomes.

4. Results

4.1. Dataset

It is easy to fix the mistakes in the KDD-99 cup dataset using the NDSL-KDD dataset, which is a new dataset composed of records extracted from the whole KDD dataset. Nevertheless, there are several issues with the dataset, such as the fact that it does not accurately portray low-footprint assaults. There are no duplicate entries in the test set, and the reduction rates are better in the NSL-KDD dataset. When compared to KDD-99, NSL-KDD contains fewer data points, making it a cheap workload option for training ML models [19].

4.2. Analysis

The classification reports that include the assessment results of the models. As well as showing the accuracy, precision recall, as well as F-1 score for each class individually, a classification report also provides an overall and weighted average of these metrics for all classes. The report also includes the level of accuracy of the predictions made using the test data as a whole. Here are the definitions:

- Accuracy = $\frac{TN+TP}{\mathrm{TN+TP+FN+FP}}$ the proportion of correct classifications to the total number.
- Precision = $\frac{\mathrm{TP}}{\mathrm{TP+FP}}$ the percentage of accurately detected positives
- Recall = $\frac{\mathrm{TP}}{\mathrm{TP+FN}}$ the proportion of true positives classified as attacks.

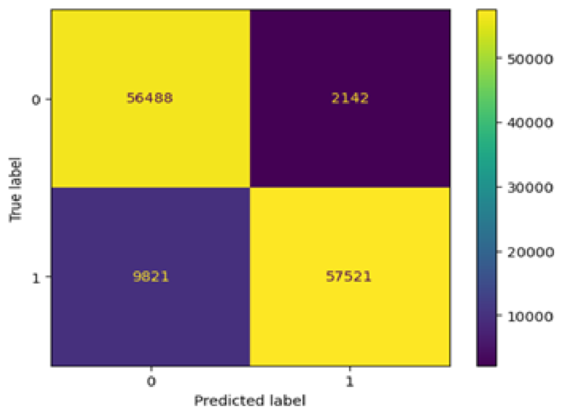
- F1 Score $= 2 * \left(\frac{\text{precision}*\text{recall}}{\text{precision}+\text{recall}}\right)$, the symmetrical average of recall and precision.

True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are the terms used here. If the F1-score is high, then the precision and recall are equally good; if it's low, then either the recall or the precision is poor. Both a high recall and precision value indicate that the model is good at identifying real positive instances. A high recall value indicates that the model can properly identify the majority of positive situations. The comparison is done with other ML algorithms like LR, DT, RF [23].

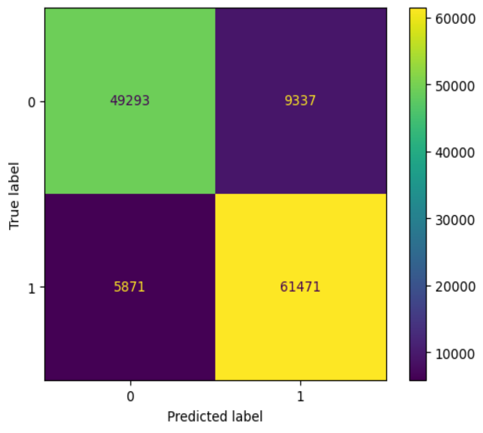
LR: One ML technique that may help you choose the best model to build a link between class variables along with features is LR. In binary class issues (with 0 and 1), the probability of belonging to the class given an observation frequently generates a number between one and zero. Nevertheless, with a few tweaks, it may be used to handle situations involving many classes [19].

Random Forest: RF is an ensemble learning classification as well as regression technique that works well for classifying data. A network of decision trees is trained during training and then utilised for class prediction. All of the trees' classes are factored into the computation, and the one with the most votes is taken as the result [19].

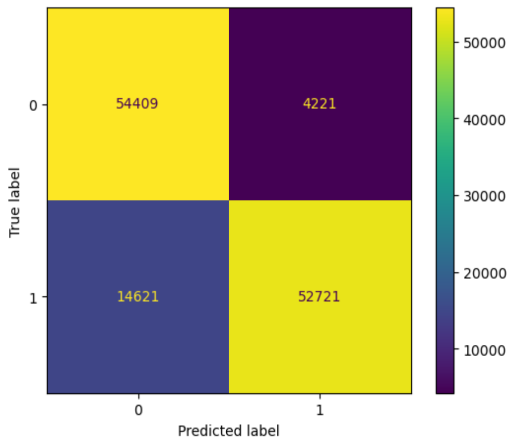
DT: There is a root node, some branches, and some leaf nodes in a decision tree. The outcome of an attribute test is stored in the branch as well as class tags, and it occurs in every internal node. At the very top of every tree is the root node. Every node in a DT stands for a feature, every link for a decision, along with every leaf for a result. Figure 2 shows the confusion matrix graph for these models [19].



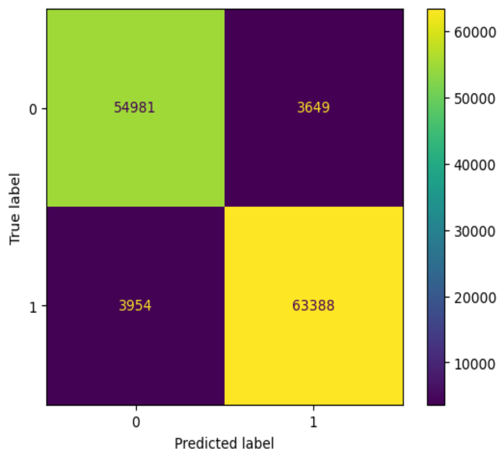
(a)



(b)



(c)



(d)

Fig.2. Confusion Matrix for RF, LR, DT and proposed LGBM

Table 1. The proposed method is compared to ML classifiers

Methods	Rec	Prec	F1 Score	Acc
RF	0.96	0.79	0.86	0.87
LR	0.87	0.91	0.89	0.88
DT	0.92	0.74	0.82	0.83
Proposed LGBM	0.95	0.94	0.94	0.94

Various approaches' performance parameters are compared in Table 1. Compared to other models like RF, DT, LR and the proposed model seems to perform better across all parameters. The proposed method is compared to training set ML classifiers. On the NSL-KDD data, the proposed LGBM method has a recall of about 0.95%. With a performance gap of 0.01%, this model beats RF, 0.08% LR and 0.03% DT. Alternatively, the proposed LGBM method has a prec of about 0.94%. With a performance gap of 0.15%, this model beats RF,0.03% LR and 0.2% DT and simultaneously the proposed LGBM method has an f1 score of about 0.94%. With a performance gap of 0.08% RF,0.05% LR and 0.12% DT. And finally, the proposed LGBM method has an acc of about 0.94%. With a performance gap of 0.07% RF,0.06% LR and 0.11% DT.

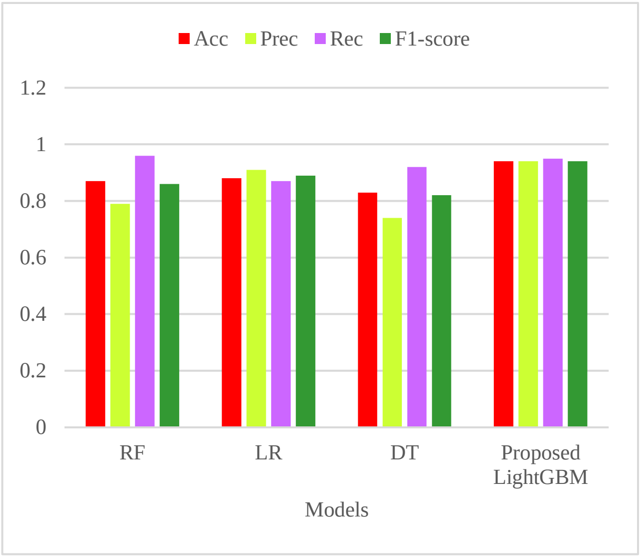
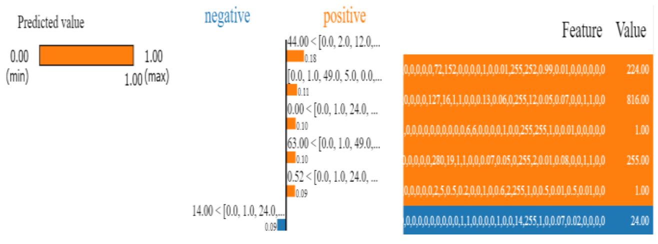
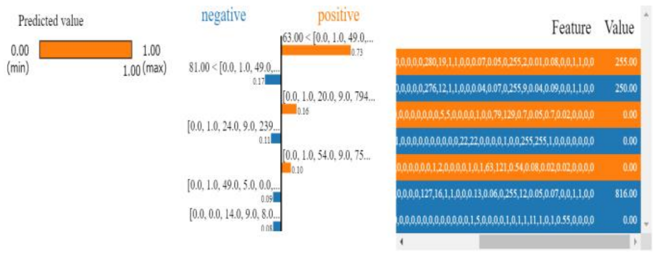


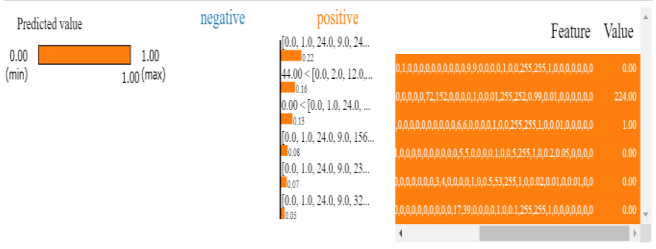
Fig.3. Comparison to the accuracy of ML classifiers in training set.



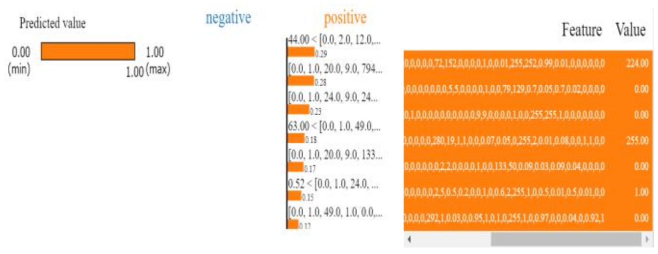
(a)



(b)



(c)



(d)

Fig.4. Lime Representation for RF, LR, DT and proposed LGBM

Figure 3 shows the proposed strategy is compared to the accuracy of ML classifiers. We have calculated various performance measures, including the prec, rec, f1 score as well as acc of the system, based on the results, which are enlisted using the LIME method with the proposed algorithm, which gives the best accuracy of 94%. Also, the LIME representations of every model are illustrated in Figure 4. Thus, it is evident that the proposed model provides improved accuracy when related to other existing classifiers.

5. Conclusion and Future Direction

In order to implement an IDS, this study suggests the LGBM ML technique. The use of a classifier allowed for the precise training of these models. This study found that trust is the most important factor in human-machine interactions. Using a modular and extendable paradigm, LIME gives a clear and simple explanation of predictions. Choosing representative models requires a firm grasp of prediction. It is useful for both system experts and others without specific training in areas such as model selection, trust assessment, repairing problematic models, and interpreting predictions. Following the training of many ML models, this training recommends using a LIME explainable framework to better understand the model's prediction. LR, LGBM, DT, and ML ensemble all contributed to improved IDS prediction accuracy, and LIME explanation graphs demonstrated how each technique performed in this regard. Applying explainability to DL-based IDS analysis might be a future extension of this work. Additionally, an app that allows users to analyse data in real-time and assess the accuracy of predictions is currently being developed.

Conflict of Interest

None

References

[1] Hochenbaum, Jordan, Owen S. Vallis, and Arun Kejariwal, "Automatic anomaly detection in the cloud via statistical learning," *arXiv preprint arXiv*:1704.07706, 2017.

[2] Garg, Sahil, Kuljeet Kaur, Shalini Batra, Gagangeet Singh Aujla, Graham Morgan, Neeraj Kumar, Albert Y. Zomaya, and Rajiv Ranjan, "En-ABC: An ensemble artificial bee colony-based anomaly detection scheme for cloud environment," *Journal of Parallel and Distributed Computing* 135, 2020, pp. 219-233.

[3] Islam, Mohammad S., William Pourmajidi, Lei Zhang, John Steinbacher, Tony Erwin, and Andriy Miranskyy, "Anomaly detection in a large-scale cloud platform," *In 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pp. 150-159, IEEE, 2021.

[4] Xin, Ruyue, Hongyun Liu, Peng Chen, and Zhiming Zhao, "Robust and accurate performance anomaly detection and prediction for cloud applications: a novel ensemble learning-based framework," *Journal of Cloud Computing* 12, no. 1 ,2023, pp. 1-16.

[5] Sherubha, P., S. P. Sasirekha, A. Dinesh Kumar Anguraj, J. Vakula Rani, Raju Anitha, S. Phani Praveen, and R. Hariharan Krishnan, "An efficient unsupervised learning approach for detecting anomaly in cloud," *Computer Systems Science and Engineering* 45, no. 1 ,2023, pp. 149-166.

[6] He, Zecheng, Guangyuan Hu, and Ruby B. Lee, "CloudShield: Real-time Anomaly Detection in the Cloud," *In Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy*, pp. 91-102, 2023.

[7] Shahzad, Faisal, Abdul Mannan, Abdul Rehman Javed, Ahmad S. Almadhor, Thar Baker, and Dhiya Al-Jumeily OBE, "Cloud-based multiclass anomaly detection and categorization using ensemble learning," *Journal of Cloud Computing* 11, no. 1, 2022, pp. 1-12.

[8] Garg, Sahil, Kuljeet Kaur, Neeraj Kumar, Georges Kaddoum, Albert Y. Zomaya, and Rajiv Ranjan, "A hybrid deep learning-based model for anomaly detection in cloud datacenter networks," *IEEE Transactions on Network and Service Management* 16, no. 3, 2019, pp. 924-935.

[9] He, Zilong, Pengfei Chen, Xiaoyun Li, Yongfeng Wang, Guangba Yu, Cailin Chen, Xinrui Li, and Zibin Zheng, "A spatiotemporal deep learning approach for unsupervised anomaly detection in cloud systems," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[10] Garg, Sahil, Kuljeet Kaur, Neeraj Kumar, Georges Kaddoum, Albert Y. Zomaya, and Rajiv Ranjan, "A hybrid deep learning-based model for anomaly detection in cloud datacenter networks," *IEEE Transactions on Network and Service Management* 16, no. 3, 2019, pp. 924-935.

[11] He, Zilong, Pengfei Chen, Xiaoyun Li, Yongfeng Wang, Guangba Yu, Cailin Chen, Xinrui Li, and Zibin Zheng, "A spatiotemporal deep learning approach for unsupervised anomaly detection in cloud systems," *IEEE Transactions on Neural Networks and Learning Systems* 2020.

[12] Thakkar, Nidhi, Miren Karamta, Seema Joshi, and M. B. Potdar, "Anomaly detection and categorization in cloud environment using deep learning Techniques," *Int. J. Comput. Sci. Eng. (IJCSE)* 7, no. 5, 2019 pp. 211-214.

[13] Hagemann, Tanja, and Katerina Katsarou, "A systematic review on anomaly detection for cloud computing environments," *In Proceedings of the 2020 3rd Artificial Intelligence and Cloud Computing Conference*, pp. 83-96, 2020.

[14] Chkirbene, Zina, Aiman Erbad, Ridha Hamila, Ala Gouissem, Amr Mohamed, and Mounir Hamdi, "Machine learning based cloud computing

anomalies detection," *IEEE Network* 34, no. 6 2020, pp. 178-183.

[15] NG, Bhuvaneswari Amma, and S. Selvakumar, "Anomaly detection framework for Internet of things traffic using vector convolutional deep learning approach in fog environment," *Future Generation Computer Systems* 113, 2020, pp. 255-265.

[16] Gulenko, Anton, Marcel Wallschläger, Florian Schmidt, Odej Kao, and Feng Liu, "Evaluating machine learning algorithms for anomaly detection in clouds," *In 2016 IEEE International Conference on Big Data (Big Data)*, pp. 2716-2721. IEEE, 2016.

[17] Shakya, Dr Subarna, and Dr S. Smys, "Anomalies detection in fog computing architectures using deep learning," *Journal of Trends in Computer Science and Smart Technology* 2, no. 1, 2020, pp. 46-55.

[18] Meng, Qiuhan, and Songye Zhu, "Anomaly detection for construction vibration signals using unsupervised deep learning and cloud computing," *Advanced Engineering Informatics* 55, 2023, pp. 101907.

[19] Fuat, T. Ü. R. K, "Analysis of Intrusion Detection Systems in UNSW-NB15 and NSL-KDD Datasets with Machine Learning Algorithms," *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi* 12, no. 2 2023, pp. 465-477.

[20] Walling, Supongmen, and Sibesh Lodh, "Performance Evaluation of Supervised Machine Learning Based Intrusion Detection with Univariate Feature Selection on NSL KDD Dataset," 2023.

[21] Venkatesan, Srinath, "Design an intrusion detection system based on feature selection using ML algorithms," *Mathematical Statistician and Engineering Applications* 72, no. 1, 2023, pp. 702-710.

[22] Mondal, Birupaxha, Fahim Faisal, Zeba Tusnia Towshi, Md Fahad Monir, and Tarem Ahmed, "A gradient boosted ml approach to feature selection for wireless intrusion detection," *In 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, pp. 1-5. IEEE, 2023.

[23] Al Lail, Mustafa, Alejandro Garcia, and Saul Olivo, "Machine learning for network intrusion detection A comparative study," Future Internet 15, no. 7, 2023, pp. 243.