International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group| Volume 2, Issue 2, 43-54, 2021
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.IJERET-V2I2P107

Original Article

Enhancing Cloud Infrastructure Security Through AI-Powered Big Data Anomaly Detection

Sunil Jacob Enokkaren[1], Varun Bitkuri[2], Raghuvaran Kendyala[3], Jagan Kurma[4], Jaya Vardhani Mamidala[5], Avinash Attipalli[6]
[1]ADP, Solution Architect, USA.
[2]Stratford University, Software Engineer, USA.
[3]University of Illinois at Springfield, Department of Computer Science, USA.
[4]Christian Brothers University, Computer Information Systems, USA.
[5]University of Central Missouri, Department of Computer Science, USA.
[6]University of Bridgeport, Department of Computer Science, USA.

Abstract *- Numerous resources and computer capabilities are made available over the Internet via cloud computing. Because of its appealing characteristics, cloud systems draw a lot of users. Cloud systems may still have serious security problems despite this. Accordingly, it's crucial to develop a system capable of detecting abnormalities in cloud environments, allowing for the high detection rate of both insider and outsider assaults. The suggested approach makes use of cutting-edge ML models. XGBoost and Multi-Layer Perceptron (MLP) combined with the necessary preprocessing techniques, i.e., feature selection and SMOTE-based class balancing, are accurate and resilient to identify anomalies in the context of a complex cloud environment. The XGBoost model performed better than other classifiers with 97.5 percent accuracy and 1.00 ROC-AUC. The Multi-Load Pump model also showed excellent results with 96.20 percent accuracy and 0.99 ROC-AUC. The superiority of the suggested models in comparison with conventional methods such as Naive Bayes (NB) and Random Forest (RF) is proved with the assistance of comparative analysis. In general, AI and big data analytics have transformed into a scalable, dependable, and proactive cloud automation framework to secure cloud environments against even advanced cyber threats.*

Keywords *- Cloud Security, Intrusion Detection System (IDS), Machine Learning, UNSW-NB15 Dataset, Cyber Threats, Anomaly Detection, Feature Selection, Network Security, Artificial Intelligence (AI), Intrusion Detection Systems (IDS).*

1. Introduction

Cloud computing has revolutionized current information technology by enabling worldwide clients to obtain scalable, on-demand services via a pay-as-you-go model [1]. It addresses the three primary service models: software as a service (SaaS), infrastructure as a service (IaaS), and platform as a service (PaaS) [2][3], and is elastic, multi-tenanted, ubiquitous and cost-effective. Nevertheless, with the growing magnitude and intricacy of cloud environments, achieving security and resilience becomes more and more difficult. The shift from traditional localized infrastructure to cloud-based solutions introduces several ethical and security concerns. In conventional infrastructures, organizations maintained direct control over their data and systems, enabling tighter security measures. In contrast, cloud environments rely on shared infrastructure, wherein control and responsibility are distributed from the customer all the way to the cloud provider [4][5][6]. This shared model increases the risk of unauthorized access to sensitive information, especially when hardware is shared across multiple tenants.

Cloud systems are getting more complex, which demands high levels of automation in failure management. Autonomous computing is required because it can perform analysis in real time and predict anomalies based on stream processing of system events [7][8]. The first objective is to attempt to anticipate system failures and security breaches before they occur by detecting them early and remediating them in time. Security-based anomalies Security is a major concern in cloud infrastructure, and unusual or malicious activities that do not fall within the scope of normal activities are a major cause of concern as well [9][10]. Such anomalies can be either when communicating between users or when transmitting data, and this is usually due to cyber-attacks or system intrusion. IDS are usually used to identify such activities [11]. IDS may be further classified as either host-based (HIDS) or network-based (NIDS), and each is designed to monitor individual hosts or whole networks, respectively.

Although traditional IDS mechanisms are necessary, they tend to lack scalability and flexibility when it comes to massive and dynamic cloud environment [12][13]. To overcome these shortcomings, the application of AI and ML to clouds has been proposed to an increasing degree with the goal of anomaly detection. Network security and fraud detection rely heavily on anomaly detection, which is the process of finding observations that act significantly differently from what is anticipated, medical diagnostics, and cyber-physical systems. These methods may be trained in supervised, unsupervised, or semi-supervised regime, based on access to

labeled data. The addition of big data analytics, as well as AI-driven models, however, has meant that cloud systems are now able such that abnormalities may be more quickly and effectively identified in real-time processing of massive quantities of diverse data [14][15][16]. Such combination of AI and big data offers a powerful means of strengthening the security of cloud infrastructure, allowing to proactively recognize threats and reduce the chances of critical system errors.

1.1. Motivation and Contribution of Paper

The growing simplicity and size of the cloud infrastructure has brought forth considerable issues in ensuring effective cybersecurity. The rule-based IDS have a tendency of failing to scale to the changing attack patterns and big data surroundings. The proposed study overcome these limitations by considering AI-driven solutions in combination with big data analytics to support real-time and reliable anomaly detection. Preprocessing in addition to using powerful models such as XGBoost and MLP. Reduces time to respond to threats but also provides scalability and flexibility in dynamic cloud environments that help to create a more resilient and protective cloud environment. The main contributions of the work as separate points:

- Modelled current network traffic and cyber threats using a realistic and large-scale benchmark dataset, UNSW-NB15.
- Performs effective data preprocessing comprising of data filtering, feature selection and class balancing through SMOTE to strengthen the applicability of the model.
- Uses and analyses powerful ML models (XGBoost and MLP) to find intrusions in a way that is both accurate and scalable.
- The study utilized many measures, such as F1-score, recall, accuracy, and precision, to evaluate the model's efficacy and make sure it could identify intrusions well.

1.2. Novelty & Justification of the Study

The novelty of the focus of this research is on the deliberate implementation of state-of-the-art ML models, XGBoost and MLP, to effectively detect anomalies within complex and large-scale cloud infrastructure environments. Unlike conventional IDSs that often miss the mark in adapting to evolving threat patterns, the proposed approach leverages the predictive power and scalability of using deep learning and ensemble methods to improve detection precision. The justification for this research stems from the growing reliance on cloud services across critical sectors and the corresponding rise in sophisticated the, ever-changing nature of contemporary cloud ecosystems, underscoring the critical need for smart, real-time security solutions to combat cyber-attacks.

1.3. Structure of paper

The paper's framework consists of many key sections. **Section II** offers a thorough analysis of the current research for enhancing security within cloud infrastructure. **Section III** outlines the planned approach. The results of the experiments are reported in Section IV. The article is wrapped up in Section V, which addresses its shortcomings and suggests possible directions for further study.

2. Literature Review

A thorough examination of current developments with the goal of bolstering cloud infrastructure security is presented in this literature study. Table I summarizes the reviewed studies, detailing the methodologies applied, performance outcomes, key findings, identified limitations, and proposed future research directions. Saad et al. (2019) work towards the goal of making cloud networks more secure and making them more resistant to harmful assaults. For incident detection across a cloud-based unified threat management (UTM) platform, this article employs bidirectional LSTM. The results are evaluated in comparison to the baseline approach, K-nearest neighbor. In order to train and test, time series input samples are captured across the UTM platform. By comparing the two methods, they find that BLSTM achieves a higher accuracy score of 98.6% with a lower loss of 0.002 than KNN does with a MSE of 0.0186 [17]. Lin, Ye and Xu (2019) To keep the network secure, they developed and deployed a DL-based dynamic anomaly detection system. They construct a deep neural network model using LSTM and then improve its performance by adding an Attention Mechanism (AM).

An enhanced loss function and the SMOTE method are that were used to resolve the problem of class imbalance in the CSE-CIC-IDS2018 dataset. Their model outperforms competing ML methods in terms of classification accuracy, according to the experimental findings, which reach 96.2% [18]. Wani et al. (2019) Operation was executed inside the secure private cloud that employs Tor Hammer as a weapon, a new dataset was built utilizing intrusion detection systems. Several ML algorithms are used in this project: For classification, SVM achieved an accuracy of 99.7 percent; for Random Forest, it was 97.5 percent; and for Naive Bayes, it was 98.0 percent [19].Aljamal et al. (2019) present two new proposals that seek to provide a hybrid detection mechanism that may identify unknown assaults using anomalies in addition to highlighting the advantages of detection systems that rely on signatures. At the Cloud Hypervisor level, they provide the use of a combination of the SVM classification algorithm and the K-means clustering technique for network-based anomaly detection. The anomaly detection system's accuracy is improved by this approach.

To test the efficacy of the suggested method, they compare their findings to those of prior research using data from the UNSW-NB15 study. Their suggested K-means clustering approach outperforms the competition by a little margin. Nevertheless, when it comes to supervised methods, the SVM model's accuracy remains poor [20]. Zaman and Lung (2018) ML classification algorithms are the foundation of a current trend in anomaly identification. Applying information entropy computation to the Kyoto 2006+ dataset, they evaluate seven different ML algorithms. Their findings show that for this dataset in particular, most ML algorithms achieve accuracy, recall, and precision levels over 90%. Nevertheless, when evaluating the seven approaches discussed in this study, the Radial Basis Function (RBF) is determined to be the superior option when the area under the Receiver Operating Characteristic (ROC) measure is used [21].

Nezarat and Shams (2017) network of mobile agents detects malicious activities in the cloud. Initiating a Playing a game that does not include cooperation with the individual they believe is threatening them enables them to determine the utility and Nash equilibrium value, which helps them distinguish between attacks and valid requests, as well as assess the intensity and origin of attacks. Results from the simulations demonstrate that this strategy has an 86% success rate in detecting the assaults. System overhead has been decreased and the detection process has been expedited thanks to the usage of mobile agents and their trainability feature [22]

Table 1: Summary of literature Overview and Review on enhancing Cloud Infrastructure Security

Study	Methodology	Dataset	Key Findings	Challenges	Future Strategy
Saad et al., (2019)	BLSTM and KNN applied to UTM data for incident detection in cloud networks	Time-series data from UTM platform	BLSTM outperforms KNN with 98.6% accuracy and 0.002 loss	Limited comparison with only one baseline (KNN)	Integrate more diverse models and test on larger, real-time datasets
Lin, Ye, and Xu (2019)	LSTM with Attention Mechanism; SMOTE for class imbalance; Improved loss function	CSE-CIC-IDS2018	Achieved 96.2% accuracy, better than other ML models	Handling class imbalance and dynamic network behaviors	Explore real-time deployment and lightweight models
Wani et al., (2019)	SVM, NB, RF tested on custom dataset generated with Tor Hammer	Custom dataset with IDS using Tor Hammer	Accuracy: SVM (99.7%), RF (97.6%), NB (98.0%)	Specific to generated environment, generalization to real-world data may be limited	Test across varied attack scenarios and real cloud environments
Aljamal et al., (2019)	Hybrid model combining K-means clustering with SVM	UNSW-NB15	K-means clustering shows higher accuracy; SVM underperforms	SVMs' lower performance in supervised detection	Enhance hybrid models using advanced clustering + deep learning
Zaman and Lung (2018)	7 ML techniques evaluated with entropy-based analysis	Kyoto 2006+	Most models >90% accuracy; RBF performs best in AUC	Dataset-specific tuning required	Apply ensemble and deep models on diverse datasets
Nezarat and Shams (2017)	Mobile agents in a game-theoretic approach (non-cooperative game, Nash equilibrium)	Simulated cloud environment	86% accuracy; reduced overhead; faster detection	Moderate detection accuracy compared to ML-based models	Enhance agent learning, integrate with AI/ML for hybrid detection

3. Methodology

The methodology adopted for enhancing cloud infrastructure security through big data anomaly detection is illustrated in Figure 1. Initially, the UNSW-NB15 dataset is collected from Kaggle and subjected to data preprocessing to eliminate inconsistencies and redundant entries. Following this, irrelevant or low-impact features are removed through feature selection techniques, and to eliminate, A technique called SMOTE is used when there is a dataset with an imbalance of classes. The training and testing sets of the processed dataset are divided 80:20. The training data is used for XGBoost and MLP, two ML models, while the testing data is utilized for other models. Common measures for evaluating model performance include F1-score, recall, accuracy, and precision to ensure reliable anomaly detection in cloud environments. This comprehensive pipeline facilitates robust and scalable intrusion detection, ultimately contributing to improved cloud infrastructure security.

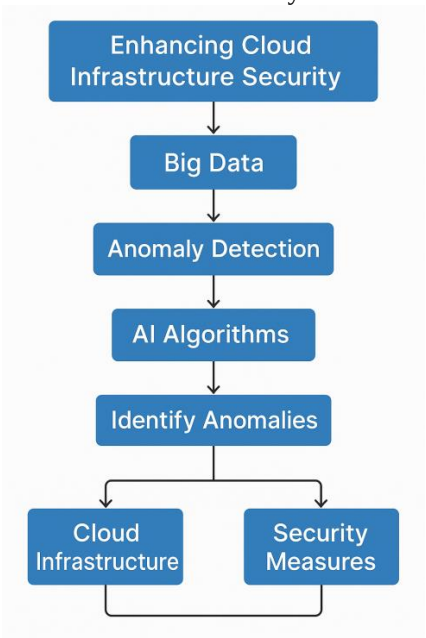

Fig 1: Flowchart diagram for Enhancing Cloud Infrastructure Security

The following provides a comprehensive, step-by-step explanation of the processes depicted in the flowchart:

3.1. Data Collection

The study utilized data set obtained from Kaggle by UNSW-NB15. In the open-source UNSW-NB15 dataset, which is further broken into 10 groups, there are 42 unlabelled features. The following groups are included: Everyday, Analysis, Fuzzers, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms are varieties of malware. It is trained using a dataset consisting of 175,341 records, and tested using a dataset consisting of 82,332 records. Several classes' worth of data from the testing and training sets of the UNSW-NB15 are drastically different from one another.

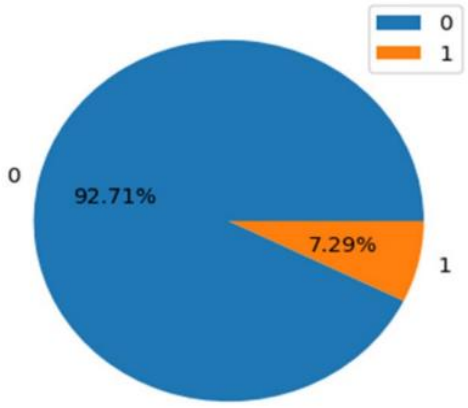

Fig 2: The EDA on the UNSW-NB15 dataset

Figure 2, the EDA on the UNSW-NB15 dataset shows how the original UNSW-NB15 dataset is distributed over two classes, '0' and '1'. The majority class, '0', constitutes 92.71% of the dataset, while the minority class, '1', represents 7.29%. This significant class imbalance is a critical characteristic of the UNSW-NB15 dataset, highlighting the need for appropriate handling techniques in ML tasks to prevent biased model performance towards the majority class.

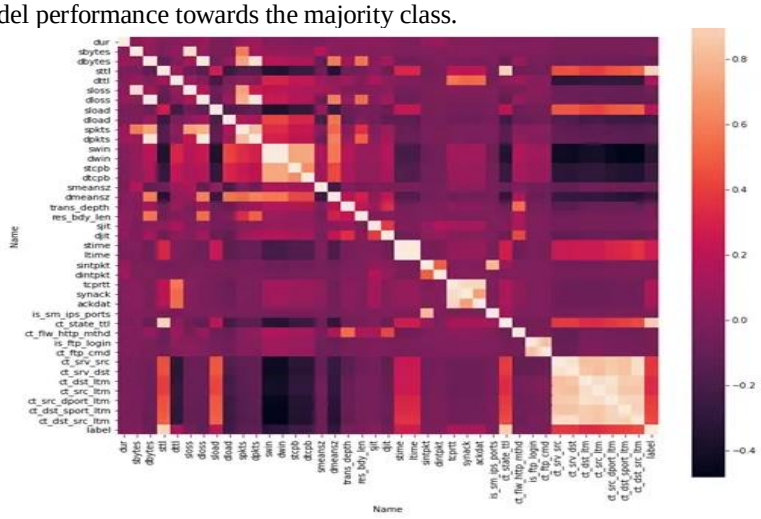

Fig 3: Correlation heatmap of all features

Figure 3 illustrates the correlation heatmap representing the pairwise relationships among all extracted features within database known as UNSW-NB15. In the heatmap, a continuous colour gradient is used to display the linear relationships. Darker colours (values near to +1) indicate strong positive correlations, whereas lighter colours (numbers farther from +1) indicate weak or negative correlations or opposite colours (values close to 0 or -1, respectively). The graphical representation is an essential feature to detect multicollinearity between features that can negatively affect some ML algorithms.

3.2. Data Preparation

To get the best possible outcomes in terms of cloud infrastructure security, the datasets used should be properly pre-processed to remove inconsistencies and any irrelevant information that might lower the accuracy of the models. Such preprocessing step usually incorporates a number of key operations, in an effort to boost the models' performance and computational efficiency, using processes like data filtering and feature selection.

- **Data Filtering:** A database query and the application of filtering rules are necessary for filtering [23]. A public, decentralized, and real-time computer system throws away data that isn't relevant or doesn't fit in. So, it lowers the cost of storage and makes sure that data is sent quickly.
- **Feature Selection:** Simplifying and improving the model's performance are both made possible via feature selection, and cut down on the time it takes to train it[24]. Picking out the most important characteristics is what feature selection (sometimes called variable selection) is all about.
- **Data Balancing:** To circumvent the issue of overfitting, the SMOTE was used. Synthetic samples for minority groups may be generated using SMOTE, an over-sampling approach. The algorithmic principle behind SMOTE is to take a dataset sample and think about its nearest neighbours in the feature space. For data points (x1, x2) that belong to a minority class and whose closest neighbor is also (x1, x2), the synthesis is done using Equation (1).

$$(X_1, X_2 = \left(x_{1,} x_2\right) + random(0,1) \times \Delta \qquad (1)$$

Where $\Delta = \{(x_{1,} x_2), (X_1, X_2)\}$ Δ in addition to the random (0, 1) value, which may take on any value between zero and one. Figure 4 displays the distribution of the dataset after SMOTE.

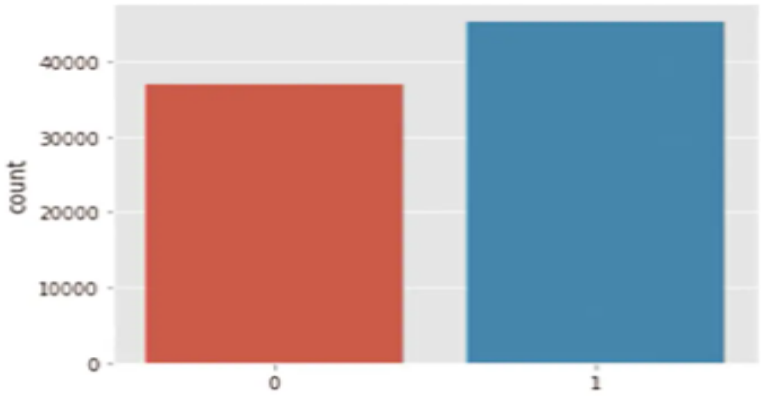

Fig 4: Data distribution after SMOTE

3.3. Data Splitting

A common ML approach is data splitting, in order to assess how well a model performs on new, unlabelled data, the dataset must be divided into training and testing sets. The models were trained using 80% of the dataset in this study, with the remaining 20% kept aside for testing.

3.4. Classification Model with XGBoost and MLP

In this section, the XGB and MLP models are discussed:

3.4.1. XGBoost

The primary goal in developing XGBoost was to maximize performance and speed by using gradient-boosted decision trees. "Machine boosting" refers to a specific approach of applying boosting to automated systems. The acronym for "extreme Gradient Boosting" is "XGBoost." a mechanism for optimizing the use of memory and hardware resources via tree boosting methods. You may use it to make models better, enhance algorithms, and implement it in your computer systems. In the context of regularized boosting, stochastic boosting, and gradient boosting, XGBoost can handle it all. It differs from other libraries since it lets you add and tweak regularization settings. When it comes to optimizing memory use and decreasing computation time, the method is second to none.

It may manage missing values, perform optimization using additional data already included in the trained model, and enable parallel structure while building trees, among its distinctive features. It is also sparse aware. XGBoost additionally uses decision-tree algorithms to classify data based on what's already known about the dataset [25]. A key component of XGBoost is supervised learning, which is based on gradient-boosted trees. XGBoost additionally uses decision-tree algorithms to classify data based on what's already known about the dataset. With supervised learning at its core, the XGBoost idea is based on gradient-boosted trees. In supervised learning, the input datatypically training datais carefully monitored and managed by a human.p_i using a variety of characteristics to foretell desired outcomes s_i.

The program uses mathematics to provide forecastss_i that is, using p_i. as trained data. To create a forecast, for example, a linear model$\widehat{s_i} = \sum_j \theta_j p_{ij}$. a number of input characteristics are combined and assigned weights. Parameter learning from data is essential. Typically, θ is used for parameter representation, and the dataset determines how many parameters are used. Whether the problem at hand is a regression, classification, ranking, or some other kind, the predict_s_if helps its categorization. Finding the optimal parameters from the training dataset is the main objective. An objective function first characterizes the model's performance. One thing to keep in mind is that different parameters might cause models to behave differently. To illustrate, let's pretend that "length" and "height" are characteristics of a dataset. Hence, many models may be constructed on the same dataset by adjusting the parameters.

There are two components to the objective function: the training loss and the regularization in Equation (2).

$$obj(\theta) = TL(\theta) + R(\theta) \qquad (2)$$

Both R and TL stand for the regularization term and training loss, respectively. The TL is nothing more than a gauge for the model's predictive power. By preventing issues like over-stacking and overfitting, regularization ensures that the model's complexity

stays within acceptable boundaries, resulting in a more accurate model. To optimize the outcome, XGBoost simply adds the predictions of all the trees in the dataset.

3.4.2. Multi-layer perceptron (MLP)

A feed forward artificial neural network (ANN) is built upon the MLP. More specifically, MLPs are a sort of feedforward ANN that is sometimes called "deep neural networks" (with threshold activation). Supervised and unsupervised learning, generation and classification algorithms, and others are typical forms of ML. A perceptron is a basic linear classifier that is both simple and effective. The perceptron model is error-driven and may adjust its parameters in response to errors [26]. In the same way that humans use precedent to guide their decision-making, supervised ML systems like DT do the same. As a method for predictive modelling, decision tree learning is used in data science, statistical analysis, and AI.

Using decision trees. An MLP requires three distinct kinds of node need the three levels to work: input, hidden, and output. The input nodes are the only exception; every other node in the network is a neuron, which has activation functions that are not linear. While training, MLP makes use of backpropagation, a supervised learning technique. Different from a linear perceptron, an MLP has a multi-layer architecture and uses an activation that is not linear. It may be used to distinguish between data sets that is neatly categorizable and data that is not. They used a classification variant MLP with the following settings because there is a discrete label in their experiment: activation relu and random state 42.

3.4.3. Evaluation Parameters

They used the F1-score, together with accuracy, precision, and recall, to measure how well the XGB-based cloud infrastructure security model performed. The capacity of the model to identify and categorize security threats in cloud systems is highly dependent on these characteristics. The following metrics were used: F-measure, Precision, Accuracy, and ROC curve, six performance measures used to assess the suggested model. The following is a definition of these assessment metrics:

- **True Positives (TP):** A **true positive** occurs when the security model correctly detects a malicious activity or cyber threat in the cloud infrastructure.
- **True Negatives (TN):** A real negative occurs when the model correctly detects normal or benign Behavior as non-malicious.
- **False Positives (FP):** The term "false positive" describes the situation in which the model flags normal activity as a security threat.
- **False Negatives (FN):** In the event that the model is unable to identify an actual cyber threat, classifying it as benign.

The evaluation metrics for each class were calculated using well-established formulas typically employed in classification tasks:

3.4.4. Accuracy

It reflects how often the model correctly identifies both cyber threats and legitimate operations in the cloud infrastructure. It is more formally defined as in Equation (3):

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \qquad (3)$$

3.4.5. Precision

A high precision indicates the model generates few false alarms, making it reliable for triggering security responses without overwhelming administrators. Precision is determined using the following formula (4):

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \qquad (4)$$

3.4.6. Recall

It reflects how effectively the model identifies real threats in the cloud infrastructure without missing them. Mathematically, It is define it as in Equation (5):

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \qquad (5)$$

3.4.7. F -measure

It is particularly important in cloud security where both detecting actual attacks (recall) and minimizing false alerts (precision) are crucial, and it is calculated as demonstrated below in Equation (6):

$$\text{F} - \text{measure} = 2 \times \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \qquad (6)$$

3.4.8. ROC-AUC

The receiver operating characteristic (ROC) curve is a useful tool for evaluating the performance of a classification task. There is no separation between the True Positive Rate (TPR) and the False Positive Rate (FPR). Analysis of the receiver operating characteristic (ROC) or area under the curve (AUC) may be used to evaluate a model's ability to separate classes. Accumulating more training data improves the model's class prediction accuracy. (As seen in Equations 7 and 8).

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \quad (7)$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP+TN}} \quad (8)$$

Our dataset has been collected from two large enterprise systems, named ESX-1 and ESX-2. The security raw events were collected over 5 months for ESX-1, over 30 days for ESX-2, respectively, in which the detecting threat information was separately recorded by the SOC security analysts whenever a network intrusion occurs. The list of threat detection information contains threat occurrence time, related attacks, category of attack, respond contents, attack IP address, and victim network information. In our datasets, we investigated 798 detecting cyber threats in ESX-1, which are dispersed across the entire collection period. Looking at the type of occurred attacks in recorded cyber threats, there are 240 scanning, 547 system hacking, and 11 worm attacks. Similarly, in ESX-2 there are 941 scannings, 3,077 system hacking, and 51 worm attacks. This categorising of attack type was manually performed by SOC analysts. By category, the system hacking attack includes a cross site script, DDoS, brute force attack, and injection attack.

A trojan and backdoor attack belongs to scanning attack. Overall the number of attacks were found 4,079 cyber-threats Our dataset has been collected from two large enterprise systems, named ESX-1 and ESX-2. The security raw events were collected over 5 months for ESX-1, over 30 days for ESX-2, respectively, in which the detecting threat information was separately recorded by the SOC security analysts whenever a network intrusion occurred. The list of threat detection information contains threat occurrence time, related attacks, category of attack, respond contents, attack IPaddress, and victim network information. In our datasets, we investigated 798 detecting cyber threats in ESX-1, which are dispersed across the entire collection period. Looking at the type of occurred attacks in recorded cyber threats, there are 240 scanning, 547 system hacking, and 11 worm attacks. Similarly, in ESX-2 there are 941 scanning, 3,077 system hacking and 51 worm attacks. This categorising of attack type was manually performed by SOC analysts. By category, the system hacking attack includes a cross site script, DDoS, brute force attack, and injection attack.

A trojan and backdoor attack belongs to scanning attack. Overall the number of attacks were found 4,079 cyber-threats Our dataset has been collected from two large enterprise systems, named ESX-1 and ESX-2. The security raw events were collected over 5 months for ESX-1, over 30 days for ESX-2, respectively, in which the detecting threat information was separately recorded by the SOC security analysts whenever a network intrusion occurred. The list of threat detection information contains threat occurrence time, related attacks, category of attack, respond contents, attack IPaddress, and victim network information. In our datasets, we investigated 798 detecting cyber threats in ESX-1, which are dispersed across the entire collection period. Looking at the type of occurred attacks in recorded cyber threats, there are 240 scanning, 547 system hacking, and 11 worm attacks. Similarly, in ESX-2 there are 941 scanning, 3,077 system hacking, and 51 worm attacks.

This categorising of attack type was manually performed by SOC analysts. By category, the system hacking attack includes a cross site script, DDoS, brute force attack, and injection attack. A trojan and backdoor attack belongs to scanning attack. Overall the number of attacks were found 4,079 cyber-threats Our dataset has been collected from two large enterprise systems, named ESX-1 and ESX-2. The security raw events were collected over 5 months for ESX-1, over 30 days for ESX-2, respectively, in which the detecting threat information was separately recorded by the SOC security analysts whenever a network intrusion occurred. The list of threat detection information contains threat occurrence time, related attacks, category of attack, respond contents, attack IPaddress, and victim network information. In our datasets, we investigated 798 detecting cyber threats in ESX-1, which are dispersed across the entire collection period.

Looking at the type of occurred attacks in recorded cyber threats, there are 240 scanning, 547 system hacking, and 11 worm attacks. Similarly, in ESX-2 there are 941 scanning, 3,077 system hacking, and 51 worm attacks. This categorising of attack type was manually performed by SOC analysts. By category, the system hacking attack includes a cross site script, DoS, brute force attack, and injection attack. A trojan and backdoor attack belongs to scanning attack. Overall the number of attacks were found 4,079 cyber-threats.

4. Results Analysis and Discussions

In this endeavour, a cloud-based instance of the Google Collab platform was utilised. The experiments were carried out on Windows 10 and a 2.30 GHz Intel Xeon CPU. Google worked in collaboration with NVIDIA to provide GPUs/CPUs that accelerated the calculations. All the experiments were carried out on Python version 3.8, and Google Collab provides 16 GB of RAM. Table II shows the performance measures of XGBoost (XGB) and MLP models in anomaly detection. The XGB model had the best outcome, with an impressive accuracy rate of 97.5%, precision and recall of 97.6%, and an F1-score of 97.5%.

This model excels at properly identifying anomalous events while maintaining a fair distribution of accurate and incorrect results. Additionally, the XGB model exhibited outstanding discriminative abilities with an ROC-AUC score of 1.00, which is considered flawless. With a recall of 96.20%, an F1-score of 96.20%, accuracy of 96.20%, and precision of 96.21%, the MLP model was also competitive. , which indicates a steady and trusted performance in terms of anomaly detection. It also has a ROC-AUC value of 99, which indicates its good classification ability, which shows its effectiveness in separating normal and abnormal patterns.

Table 2: Results of XGB Model for Anomaly Detection

Metrics	XGB	MLP
Accuracy	97.5	96.20
Precision	97.6	96.21
Recall	97.6	96.20
F1-Score	97.5	96.20
ROC-AUC	1.00	0.99

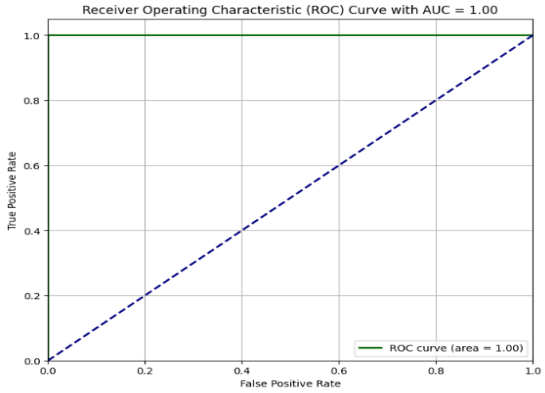

Fig 5: ROC Curve Visualization of XGB Model

The ROC curve, Figure 5, that was presented in the XGBoost (XGB) classification model is indicative of an outstanding model performance because the curve follows the upper-left boundary starting at (0,0) and passing through (0,1) and then (1,1). The Area Under the Curve (AUC) for this ideal curve is 1.00, indicating that it has superb discriminating power between the positive and negative categories.

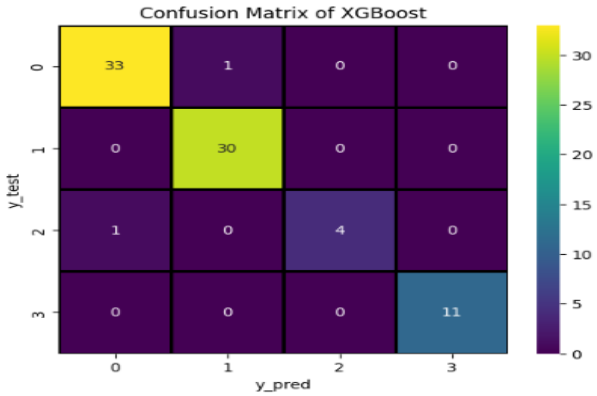

Fig 6: Confusion Matrix of XGB Model

Figure 6 contains a confusion matrix that displays the XGBoost model's classification results across four classes (0–3), with the actual labels (y_test) shown on the vertical axis and the predicted labels (x predict) on the horizontal axis (y pred). The matrix highlights strong diagonal dominance with correct classifications: 33, 30, 4, and 11 for classes 0, 1, 2, and 3, respectively. Minimal misclassifications occurred. One instance of class 2 was mistakenly classified as class 0, and one class 0 instance was incorrectly labelled as class 1. The model exhibits high accuracy and minimal class confusion, particularly for classes 0, 1, and 3, and with visual colour coding of lighter colours representing high counts and darker colours representing low ones.

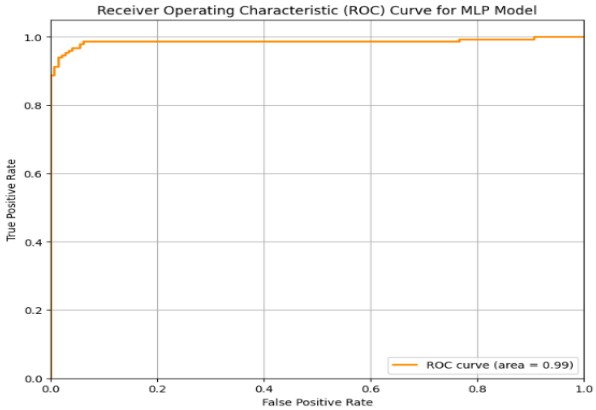

Fig 7: Receiver Operating Characteristic (ROC) graph of the MLP model

Figure 7 shows the ROC curve of the True Positive Rate (Sensitivity) vs the False Positive Rate (1 - Specificity) for different threshold settings. This curve analyzes the performance of the MLP model. An orange curve, called the ROC curve (area = 0.99), indicates a high rate of true positives with few false positives; it closely follows the top-left border of the graph. The model fits the data well with an AUC of 0.99, exceptional discriminative capability, confirming the MLP as a highly accurate and reliable classifier.

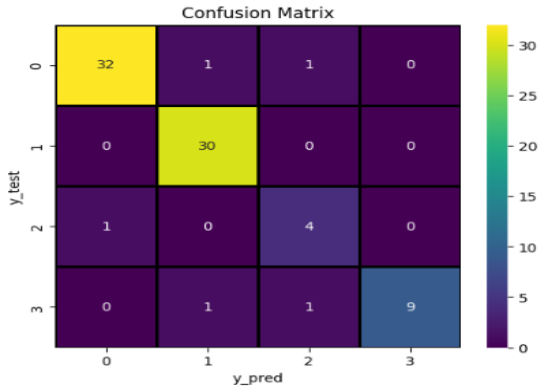

Fig 8: Confusion matrix of the MLP model

Figure 8 displays the matrix of confusion, which is used for evaluating the effectiveness of the categorization. The MLP model across four classes (0–3), with real labels shown vertically and forecasted labels displayed horizontally. The model achieved correct predictions for 32 instances of class 0, 30 of class 1, 4 of class 2, and 9 of class 3, as indicated by the diagonal elements. Misclassifications include one instance each of class 0 as class 1 and class 2, one of class 2 as class 0, and one of class 3 as class 1. Despite these few off-diagonal errors, the MLP model exhibits high classification accuracy, as reflected by the bright diagonal cells in the matrix.

4.1. Comparative Analysis

A variety of ML models that have been used for anomaly detection have been compared in Table III to enhance cloud security. Among the evaluated models, the XGBoost classifier has a remarkable 97.5% accuracy rate, showing that it is quite good at detecting intricate patterns in the data. The MLP model follows closely with an accuracy of 96.20%, demonstrating the efficacy of designs based on deep learning in security-related domains. Naive Bayes and Random Forest, two popular traditional ML models, achieve 86% and 93.6% accuracy, respectively, highlighting their limitations in handling high-dimensional and imbalanced datasets commonly found in cloud environments. These results affirm the advantage of advanced ensemble and neural network-based models for robust and precise anomaly detection in cloud infrastructure.

Table 3: Existing Models comparison Performance for Enhancing Cloud Security in anomaly detection

Metrics	Accuracy
XGBoost	97.5
MLP	96.20
Naive Bayes[27]	86
Random Forest[28]	93.6

The proposed anomaly detection framework, which integrates XGBoost and MLP models, achieves a high accuracy of 97.5% and 96.20%, respectively. This result validates the efficacy of merging DL and ensemble learning methods for discovering cloud infrastructure abnormalities in environments. The integration of SMOTE for class balancing and rigorous model performance and generalization is further improved by feature selection. The suggested solution's key strength is in its ability to efficiently analyze high-dimensional large data and provide solid, scalable, and precise results on intrusion detection and help build a safer and more sustainable cloud environment.

5. Conclusion and Future Work

The security of cloud infrastructure has risen to be a crucial issue, as it is the foundation of current digital services. This is because of the dynamic and distributed environment of a cloud, which presents complicated vulnerabilities and attack surfaces. Conventional security controls are not always effective at detecting advanced threats promptly. One of the proactive methods to detect anomalies and mitigate threats is to use AI and big data analytics. In order to enhance the safety of cloud infrastructure using the UNSW-NB15 dataset, this study suggests an anomaly detection system that is based on AI. The proposed assessment of the XGBoost and MLP-based methods is, advanced ML model. As experimental outcomes show, the XGBoost model is superior to other models, as it shows an accuracy of 97.5%, and MLP also performs remarkably well with an accuracy of 96.20%.

This is a highly resilient and elastic framework that would go a long way in proactive threat detection and superior security of cloud-based applications. To make this research even more solid, one can suggest future efforts that will focus on the inclusion of XAI methods to enhance model interpretability and build trust among cloud security analysts. Moreover, it is possible to apply streaming data frameworks to realize real-time anomaly detection capabilities. Exploring transformer-based architectures or hybrid models with AM may also enhance detection accuracy in evolving cloud environments. Lastly, applying transfer learning to adapt the trained models to other cyber threat datasets and domains can improve model generalizability and widen the applicability of the proposed framework.

References

[1] Song Fu, “Performance Metric Selection for Autonomic Anomaly Detection on Cloud Computing Systems,” in *2011 IEEE Global Telecommunications Conference - Globecom 2011*, IEEE, Dec. 2011, pp. 1–5. doi: 10.1109/GLOCOM.2011.6134532.

[2] H. R. Faragardi, A. Rajabi, T. Nolte, and A. H. Heidarizadeh, “A Profit-aware Allocation of High Performance Computing Applications on Distributed Cloud Data Centers with Environmental Considerations A Profit-aware Allocation of High Performance Computing Applications on Distributed Cloud Data Centers with Environmen,” *CSI J. Comput. Sci. Eng.*, pp. 1–11, 2014.

[3] S. Garg, “Predictive Analytics and Auto Remediation using Artificial Inteligence and Machine learning in Cloud Computing Operations,” *Int. J. Innov. Res. Eng. Multidiscip. Phys. Sci.*, vol. 7, no. 2, 2019, doi: 10.5281/zenodo.15362327.

[4] B. de Bruin and L. Floridi, “The Ethics of Cloud Computing,” *Sci. Eng. Ethics*, vol. 23, no. 1, pp. 21–39, Feb. 2017, doi: 10.1007/s11948-016-9759-0.

[5] H. R. Faragardi, “Ethical Considerations in Cloud Computing Systems,” in *Proceedings of the IS4SI 2017 Summit Digitalisation For A Sustainable Society, Gothenburg, Sweden*, Basel Switzerland: MDPI, Jun. 2017, p. 166. doi: 10.3390/IS4SI-2017-04016.

[6] S. Garg, “AI/ML Driven Proactive Performance Monitoring, Resource Allocation And Effective Cost Management In Saas Operations,” *Int. J. Core Eng. Manag.*, vol. 6, no. 06, pp. 263–273, 2019.

[7] J. Park and J. Park, “Blockchain Security in Cloud Computing: Use Cases, Challenges, and Solutions,” *Symmetry (Basel).*, vol. 9, no. 8, p. 164, Aug. 2017, doi: 10.3390/sym9080164.

[8] K. Hashizume, D. G. Rosado, E. Fernández-Medina, and E. B. Fernandez, “An analysis of security issues for cloud computing,” *J. Internet Serv. Appl.*, vol. 4, no. 1, p. 5, 2013, doi: 10.1186/1869-0238-4-5.

[9] S. Singh, Y.-S. Jeong, and J. H. Park, “A survey on cloud computing security: Issues, threats, and solutions,” *J. Netw. Comput. Appl.*, vol. 75, pp. 200–222, Nov. 2016, doi: 10.1016/j.jnca.2016.09.002.

[10] V. Kolluri, “A Pioneering Approach To Forensic Insights: Utilization of AI for Cybersecurity Incident Investigations,” *Int. J. Res. Anal. Rev.*, no. August 2016, 2016.

[11] M. Nawir, A. Amir, O. B. Lynn, N. Yaakob, and R. B. Ahmad, "Performances of Machine Learning Algorithms for Binary Classification of Network Anomaly Detection System," *J. Phys. Conf. Ser.*, pp. 1–9, May 2018, doi: 10.1088/1742-6596/1018/1/012015.

[12] A. Mikail and B. Pranggono, "Securing Infrastructure-as-a-Service Public Clouds Using Security Onion," *Appl. Syst. Innov.*, vol. 2, no. 1, p. 6, Jan. 2019, doi: 10.3390/asi2010006.

[13] S. K. Yoo and B. Y. Kim, "A Decision-Making Model for Adopting a Cloud Computing System," *Sustainability*, vol. 10, no. 8, p. 2952, Aug. 2018, doi: 10.3390/su10082952.

[14] R. C. Aygun and A. G. Yavuz, "Network Anomaly Detection with Stochastically Improved Autoencoder Based Models," in *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, IEEE, Jun. 2017, pp. 193–198. doi: 10.1109/CSCloud.2017.39.

[15] V. Kolluri, "Cutting-Edge Insights into Unmasking Malware: AI-Powered Analysis and Detection Techniques," *JETIR*, vol. 4, no. 2, 2017.

[16] S. Singamsetty, "Fuzzy-Optimized Lightweight Cyber-Attack Detection for Secure Edge-Based IoT," *J. Crit. Rev.*, vol. 6, no. 07, pp. 1028–1033, 2019, doi: 10.53555/jcr.v6:i7.13156.

[17] M. M. Saad, T. Iqbal, H. Ali, M. F. Bulbul, S. Khan, and C. Tanougast, "Incident Detection over Unified Threat Management Platform on a Cloud Network," in *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, IEEE, Sep. 2019, pp. 592–596. doi: 10.1109/IDAACS.2019.8924299.

[18] P. Lin, K. Ye, and C. Z. Xu, "Dynamic Network Anomaly Detection System by Using Deep Learning Techniques," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, pp. 161–176. doi: 10.1007/978-3-030-23502-4_12.

[19] A. R. Wani, Q. P. Rana, U. Saxena, and N. Pandey, "Analysis and Detection of DDoS Attacks on Cloud Computing Environment using Machine Learning Techniques," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, IEEE, Feb. 2019, pp. 870–875. doi: 10.1109/AICAI.2019.8701238.

[20] I. Aljamal, A. Tekeoglu, K. Bekiroglu, and S. Sengupta, "Hybrid Intrusion Detection System Using Machine Learning Techniques in Cloud Computing Environments," in *2019 IEEE 17th International Conference on Software Engineering Research, Management and Applications (SERA)*, IEEE, May 2019, pp. 84–89. doi: 10.1109/SERA.2019.8886794.

[21] M. Zaman and C. H. Lung, "Evaluation of machine learning techniques for network intrusion detection," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, IEEE, Apr. 2018, pp. 1–5. doi: 10.1109/NOMS.2018.8406212.

[22] A. Nezarat and Y. Shams, "A game theoretic-based distributed detection method for VM-to-hypervisor attacks in cloud environment," *J. Supercomput.*, vol. 73, no. 10, pp. 4407–4427, Oct. 2017, doi: 10.1007/s11227-017-2025-7.

[23] M. Saqlain, M. Piao, Y. Shim, and J. Y. Lee, "Framework of an IoT-based Industrial Data Management for Smart Manufacturing," *J. Sens. Actuator Networks*, vol. 8, no. 2, p. 25, Apr. 2019, doi: 10.3390/jsan8020025.

[24] B. S. Khater, A. A. B. W. Wahab, M. Y. I. Bin Idris, M. A. Hussain, and A. A. Ibrahim, "A Lightweight Perceptron-Based Intrusion Detection System for Fog Computing," *Appl. Sci.*, vol. 9, no. 1, p. 178, Jan. 2019, doi: 10.3390/app9010178.

[25] S. S. Dhaliwal, A.-A. Nahid, and R. Abbas, "Effective Intrusion Detection System Using XGBoost," *Information*, vol. 9, no. 7, p. 149, Jun. 2018, doi: 10.3390/info9070149.

[26] T. T. Teoh, G. Chiew, E. J. Franco, P. C. Ng, M. . Benjamin, and Y. J. Goh, "Anomaly detection in cyber security attacks on networks using MLP deep learning," in *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, IEEE, Jul. 2018, pp. 1–5. doi: 10.1109/ICSCEE.2018.8538395.

[27] K. Kostas, "Anomaly Detection in Networks Using Machine Learning," 2018.

[28] T. Salman, D. Bhamare, A. Erbad, R. Jain, and M. Samaka, "Machine Learning for Anomaly Detection and Categorization in Multi-Cloud Environments," in *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, IEEE, Jun. 2017, pp. 97–103. doi: 10.1109/CSCloud.2017.15.

[29] Kalla, D., & Samiuddin, V. (2020). Chatbot for medical treatment using NLTK Lib. *IOSR J. Comput. Eng*, *22*, 12.

[30] Kuraku, S., & Kalla, D. (2020). Emotet malware a banking credentials stealer. *Iosr J. Comput. Eng*, *22*, 31-41.

[31] Sreejith Sreekandan Nair, Govindarajan Lakshmikanthan (2020). Beyond VPNs: Advanced Security Strategies for the Remote Work Revolution. International Journal of Multidisciplinary Research in Science, Engineering and Technology 3 (5):1283-1294.

[32] Masud, M. M., Moniruzzaman, M., Rahman, M. M., & Noor, S. (2009). Effect of poultry manure in combination with chemical fertilizers on the yield and nutrient uptake by chilli in the hilly region. *J. Soil Nat*, *3*(2), 24-27.