iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** X **Month of publication:** October 2025

DOI: https://doi.org/10.22214/ijraset.2025.74139

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue X Oct 2025- Available at www.ijraset.com

Detecting Network Traffic Anomalies with Machine Learning: A Comprehensive Approach

N. Vinisha[1], Sri Gopala Krishna[2]
[1]*M. Tech (Data Science), Post Graduate Student, Department of IT, GRIET, INDIA*
[2]*Ph.D (Computer Science), Professor, Head of the Department of IT, GRIET, INDIA*

Abstract: ***Continued growth in network traffic as well as increased complexity in network architecture make anomaly detection critical to maintaining the security and reliability of networks. This abstract provides a detailed review of utilizing machine learning techniques in anomaly detection for network traffic. Machine learning methodologies have thus far proven promising approaches in identifying anomalies of network traffic since they can detect pattern-specific characteristics in large datasets. Different approaches to machine learning; that this abstract discusses, both supervised and unsupervised learning techniques deployed in anomaly detection. The supervised learning methods are basically those in which classifiers are trained on labeled data to classify data into normal versus anomalies in terms of traffic patterns. Instead, this technique employs unsupervised learning that finds the presence of anomalies without pre-classified data. A copy of the similar instances would be made and outliers flagged out. Another significant area that this abstract talks about is the difficulties that arise in anomaly detection in network traffic. These include the imbalanced nature of network data, evolving attack strategies, and the requirement for real-time detection. It aims to develop robust models that can identify different types of anomalies which may arise with cyberattacks like DDoS, port scanning, or even simple data exfiltration. Using LOF, SVM, KNN.***
Keywords: Cyberattacks, Isolation Forest, LOF, SVM, KNN.

INTRODUCTION

Millions of people and hundreds of thousands of institutions exchange messages with each other through the Internet every day. Though the number of users of the Internet has grown very rapidly in the last two decades, today this figure has reached over 4 billion and the increment is quickly accelerating. Along with these trends, attacks made on the Internet are growing every day. For these attacks, two basic methods have been found to detect them so that information safety can be assured. These two methods are identification based on signature and detection based on anomaly.

Signature-based methods make use of a database for identification of the attacks. This method is very effective, but the databases must be constantly updated and new attack information processed. Besides, even with an up-to-date database, it is still not immune to the zero-day (previously unseen) attacks. Since these attacks are not in the database, it cannot prevent these attacks. The anomaly-based approach aims to detect unusual network behaviors by looking at the network flow. This approach had passed all tests so far in the detection of attacks it had never seen hence effective against zero-day attacks.

More than half of today's use of the internet is encrypted by SSL / TLS (Secure Sockets Layer / Transport Layer Security) protocols and this rate is going up day by day. This feature makes such a method based on signature ineffective because, for instance, one cannot view the contents of the encrypted internet stream. Whereby, for anomaly-based approaches, the analysis is taken based on general properties of the data concerning size, duration of connection, and the number of packets. Hence, it does not need to see the message content and can also do the analysis of encrypted protocols. Owing to all these merits, the anomaly-based detection method is highly in use to detect the network attacks and prevent them. This study aims to contribute to the literature by developing a system that will quickly and effectively detect network anomalies through the use of machine learning methods.

A. Goals

Goals are the set of achievements that are to be attained at the end of this study, which include:

1) Examination of algorithms using machine learning applicable to network anomaly detection.
2) Study the network anomaly to detect network attacks quickly and efficiently using approaches from machine learning.
3) Measure the level of success of the study by comparing the results with previous studies carried out in this domain.
4) Supporting literature for close outcome gathering from the prior studies as in the case of network anomaly detection.

B. Objectives

The objectives that are intended to be achieved at the end of this study are as below:

1) To consult the previous work done in the field by carrying out wide field research
2) Selecting the right dataset after carrying out wide research on options of dataset.
3) Choosing right algorithms after doing wide research on machine learning algorithms
4) Choosing the right algorithms after conducting good research on various machine learning techniques
5) Choosing a suitable software platform.
6) Choosing the right hardware/Equipment platform.
7) Choosing suitable selection criteria.
8) Choosing Benchmarking parameters to be measured and compared during the Evaluation stage.

I. INTRODUCTION TO ANOMALY DETECTION

A. Anomaly detection in network traffics through the use of machine learning

The process of determining unusual patterns that deviate from the expected behavior. It becomes critical in ascertaining cyber threats, intrusion, and other forms of security incidents. Below is a guide on how to put machine learning into work to detect anomaly in network traffics:

B. Definition

Anomaly Detection of Network Traffic is the process through which abnormal patterns, which may lead to malicious activity such as unauthorized access, malware, and exfiltration of data, are identified.

Importance: Anomalies in network traffic lead to preventing future security breaches, decreases damage, and protects sensitive information.

C. Types of Anomalies

A sample is said to be an anomaly when it lacks well-defined properties of a normal sample. The anomaly can only be understood if there exist specific and valid rules forming the normal concept. Anomaly is analyzed under 3 headings:

1) Point anomaly: In this anomaly, the samples of data depict other characteristics than the overall data set where they have been comprised. In example, it is a kind of point anomaly that a man who has been making very low shopping every day by using a credit card makes on a random day a very high amount of shopping.
2) Contextual anomaly: An activity of a set of data out of pattern which depends upon the specific conditions or happens under specific conditions. For example, it is a contextual anomaly for the individual who daily makes very low amounts of shopping through credit cards making a very high amount of shopping on the feast day.
3) Collective anomaly: It is called collective anomaly when it is abnormal in properties for a data heap of similar data as compared to normal data. In other words, the increase in credit cards spending on Valentine's Day is a collective anomaly.

D. Types of Network Attacks.

1) Confidentiality: Only the legitimate users should have access to the information and unauthorized access should be avoided.
2) Integrity: Legitimate user alone can addition, modification and deletion of information. Unauthorized persons should not be able to modify the information.
3) Accessibility: The system should always be accessible to the legitimate user.
4) Denial of Service (DoS): Here, the attacker uses the system resources in order to prevent the legitimate user from availing the service. Indeed, perhaps the most primitive example would be when it is sent a large number of requests to drop out of service a web server. The Dos attacks can be categorized into two subparts: bandwidth depletion and resource depletion. Whereas bandwidth depletion tries to exhaust a victim's bandwidth by providing a very high data flow, resource depletion attacks generally try to consume a victim's memory and processor resources with abundant packets.
5) Probe (Information Gathering): This category of attacks is information gathering-oriented, that is, intended to gather information about the target. In this form of attack, the intruder can capture as much useful information he or she can gain, such as the network structure, the type of operating system in use, kinds and properties of the networked devices. Though this attack does not have an immediate impact on the system, it is very significant because it is preparing the ground for so many attacks that would cause harm to the system.

6) U2R - User to Root Attack This attack refers to the attacker's effort to achieve the administrative control of an account, whereby he or she would be able to access and obtain resources. Acquisition of the Administrator account is achieved through either exploitation of the system's vulnerability or brute-force attacks.
7) R2U / R2L (Remote to User / Remote to Local): An attacker obtains privilege of sending packets from a victim computer by breaking into the victim's network. Privilege can be achieved through system vulnerability or brute-force attacks. Cyber security attacks target persons, companies and even government organizations. Let us address some of the most common hacker's book attack methods which include DDoS attacks, port scanning, and data exfiltration.
8) DDoS Attacks (Distributed Denial of Service): A DDoS attack sends an overwhelming amount of traffic to a target system, usually a website or network, from a multitude of sources to exhaust all of the target's resources - bandwidth and processing power, thereby reducing it to uselessness.
9) Port Scanning: Port scanning is a form of reconnaissance attack in which attackers want to identify open ports in the system networked. Ports represent communication endpoints that have the ability for two systems to talk with one another. Through determining which ports are open, and which services are operating on those open ports, attackers discover various points of vulnerabilities from which to take advantage.
10) Data Exfiltration: Data exfiltration is the unauthorized transfer of sensitive data out of a network. That's usually what attackers are looking to do because they can monetize stolen information, including personal data, intellectual property, and financial records.

II. EXISTING SYSTEM

Anomaly detection of network traffic using machine learning is one of the new important branches of modern cybersecurity systems. The existing systems concentrate on patterns of network data that are not in normal behavior and can be of intrusion, malware, or any other unauthenticated access. These systems apply various techniques of machine learning that can be either applied online or offline. Here is a general description of the existing solutions, widely known machine learning technologies, datasets, and system architectures applied for anomaly detection in network traffic.

Despite tremendous progress in technology and data science, several gaps are yet to be bridged in Anomaly detection on network traffic prediction.

As the former model comprises of the general ML models in which the datasets are been trained and tested and algorithms used where, Naive Bayes 86%, QDA 86%, Random Forest 94%, ID3 95%, AdaBoost 94%, MLP 83%

Existing system also comprises of traditional methods for anomaly detection that includes Rule-Based Systems, Statistical Methods, Signature-Based Detection.

III. LITERATURE REVIEW

Chandola, Banerjee, and Kumar, [1] An extensive survey on machine learning-based approaches toward network traffic anomaly detection is presented in this paper. Supervised, semi-supervised, and unsupervised learning methodologies are presented and discussed. It also emphasizes the problem of high-dimensional network traffic data processing and the need for feature engineering.

A. Lakhina, M. Crovella, and C. Diot, [2] In this paper, the authors introduce a framework for an unsupervised learning approach in anomaly detection for large-scale network traffic. The authors suggest one identifying anomalies with low-dimensional flows of network data. The authors focus on dimensionality reduction methods like PCA and clustering.

J. Kim, T. Shon, [3] This paper deals with deep learning methods for network anomaly detection. It includes different architectures, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Auto-encoders, which are examined in the context of applying to various anomaly detection problems in network traffic.

Tavallaee, Ebrahim, Lu, Wei, et al. [4] The authors describe an unsupervised learning technique that will need to be adopted for anomaly detection in network traffic and primarily rely on clustering techniques and outlier detection methods. The article is focused on comparing several variants of clustering techniques like K-means and DBSCAN for analysis purposes in network flows.

Dong Wei, Xiao Wen, et al, [5] A review of deep learning-based models for anomaly detection in network traffic, discussing how these models are capable of learning abstract representations from raw traffic data. The paper compares deep learning models with classical machine learning models and emphasizes strength differences particularly in feature extraction and representation learning.

M. Ahmed, A. N. Mahmood, and J. Hu, [6] This paper proposes a new framework for anomaly detection that utilizes unsupervised clustering algorithms. Many different clustering techniques are tested and a hybrid method that combines several clustering algorithms are used in order to improve performance in anomaly detection.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue X Oct 2025- Available at www.ijraset.com

G. Sommer and K. Paxson, [7]This paper discusses supervised machine learning models for intrusion detection in network systems by studying the performance of classes of classifiers that include Random Forests, SVMs, and neural networks, with regard to identifying malicious behavior in network traffic.

IV. PROPOSED METHODLOGY

Suggested system has the implementation on anomaly detection in network traffic using ML models where the datasets are been trained and tested and algorithms used are:

1) KMEANS - 99.61%
2) ISOLATION FOREST-99%
3) ONE-CLASS SVM-94%
4) LOCAL OUTLIERS FACTOR -94%
5) #Using Scikit-learn's Nearest Neighbors -94%
6) PyOD KNN Anomaly Detection Accuracy: 95%

The proposed anomaly detection system in network traffic is the implementation of various machine learning models where datasets have been trained and tested to recognize anomalous behavior in network traffic. The aim of this paper is to compare different algorithms for anomaly detection based on how they performed over the accuracies achieved on the datasets. Here's an overview of the system along with the accuracies attached to the algorithms used.

A. Overview of the System Suggested

The system here detects anomalies by training numerous models based on network traffic datasets on what normal behavior looks like, flagging any deviation from the said pattern as a potential anomaly example such as unauthorized access, malware attacks, or DDoS. The system mainly employs unsupervised and semi-supervised machine learning models because labeling network data may become intricate and time-consuming.

1) K-Means Clustering

Description: K-Means is a type of unsupervised clustering algorithm that can be applied to network data. For this purpose, K-means can be applied by clustering network data points similar to each other. In general, K-means-based anomaly detection involves partitioning the network traffic into normal or anomalous clusters based on the features used for the similarity metric of the application. The outliers here refer to anomalies.

How It Works: The algorithm determines K centroids and maps every data point to the closest centroid, minimizing within-cluster variance. Many anomalies are detected as points far from any centroid or small isolated clusters.

2) Isolation Forest

Description: Isolation Forest is a tree-based algorithm which was developed specifically with anomaly detection in mind. The isolation forests method of isolation detects anomalies by randomly choosing features and splitting data. Anomalies are isolated easily because less splitting is needed.

How It Works: The algorithm builds random trees, and subsequently computes the "path length" to separate individual points. Points with a smaller path length are anomalies. The smaller the path length, the more anomalous is that data point.

3) One-Class SVM (Support Vector Machine)

Description: One-Class SVM is a semi-supervised technique, which learns a boundary of the decision around the normal data points. Points are anomalous when they fall outside the boundary of this decision boundary.

How It Works: It embeds a hyperplane around the normal data points in feature space. Those points that are far-off from this hyperplane are rated as anomalies. It works great if the normal data follows some known distribution, but it does not work well with high-dimensional data

4) Local Outlier Factor (LOF)

Description: LOF is an unsupervised anomaly detection technique, in which data points with a lower density than their neighbors have been classified. It calculates the local density deviation of all data points and flags those with significantly lower densities as anomalies.

How It Works: LOF it compares the density of a point with the density of its neighbors. If a point is significantly less dense than its neighbors it gets marked as an outlier. It's pretty good at detecting outliers for regions where the densities are different.

5) Scikit-learn's Nearest Neighbors

Description: Nearest Neighbors is another method for detecting outliers. It actually just depends on the choice of K-nearest neighbours to be made for each point in the data and then classify points with fewer or further-away neighbors as anomalies.

How It Works: For every single data point, this algorithm chooses its nearest neighbours. The point will be marked as a novelty if those neighboring points were either far away or fewer in number than that expected.

6) PyOD KNN Anomaly Detection

Description: The PyOD anomaly detection algorithm based on KNN uses the KNN algorithm for anomaly detection. This algorithm forms part of the highly inclusive library called PyOD.

How It Works: Anomaly is detected with the help of the KNN algorithm, and to determine this, calculate the distance to the K-nearest neighbours. If the point is further away from its nearest neighbors than most other points are, then it's declared anomalous

V. DATASET DESCRIPTION

The data set used here is assumed to be one of network traffic with 42 columns, and many of these are traditionally used for anomaly detection and intrusion detection tasks. The key columns described briefly, along with their meanings, are given below:

1) tduration : Connection length.
2) tprotocol_type: Kind of protocol (TCP, UDP etc.).
3) tservice: the network service on the destination (HTTP, FTP, etc.).
4) flag: Connection status. Could be SF for successful connection.
5) src_bytes: Bytes from source to destination.
6) dst_bytes: Bytes from source to destination.
7) land: Connection is either from the same or to the same host.
8) wrong_fragment, urgent, hot, etc.: Features that feature the characteristics of the connection.
9) dst_host_srv_count, dst_host_same_srv_rate, and many more: Features defining how the destination host behaves to connections.
10) \tclass: The target column, with a label of either 'normal' or 'anomaly'.

The dataset is likely to be qualified for network anomaly detection using machine learning methods. In it, a variety of features describe the behavior of network traffic and could thus form the basis of classifiers aiming to identify anomalous behavior.

VI. STAGES IN THE PROJECT

A. Data Collection

Data collection In anomaly detection system implementation, appropriate data must be collected from network traffic. Network data are always being generated, and it could be gathered from different devices such as routers, firewalls, or an IDS. The most common types of data are ;

- Packet data: raw information of the network packet, such as packet size, protocol type, source/destination IP.
- Flow data: summarized communication between pairs of hosts, like NetFlow or IPFIX.
- Log data: Data recorded by devices and applications, such as web server logs and syslogs.
- Time series data: Network activity measured at different points in time, such as volume of traffic and bandwidth usage
- Few of the most commonly used network traffic datasets for research are:
- UNSW-NB15: Relatively new dataset, designed for benchmarking network intrusion detection systems.
- KDD Cup 1999: An older, much-used dataset related to network anomaly detection.
- CICIDS2017: This contains good as well as malicious traffic, and is a first choice for evaluating intrusion detection models.

B. Data Preprocessing

Raw data for network traffic is usually noisy and very complex. It therefore has to undergo preprocessing effectively with the aid of a machine learning model. The preprocessing involves several stages:

1) Data Cleaning: The process of filling in missing data: There may be missing entries within a network log or traffic data. This should be addressed either by filling in reasonable values into it, or dropping of the incomplete rows.
2) Removing duplicates: Duplicate records will lead to bias in the performance of the model.
3) Filter out noise data: Filter out not intrusive or irrelevant traffics and focus on useful patterns.
4) Normalization and Scaling: Normalizing/Scaling Features of network traffics often do not have the same range. Packet size, time intervals in one traffics may be different from another. In feature normalization, for example, with MinMaxScaler or Z-score normalization, all the features are normalized onto the same range. Scaling: Scaling enables algorithms such as k-NN and SVM to perform better by not letting any one feature dominate due to the larger scale.
5) Feature Selection/Extraction: Feature selection: Not all the features contribute equally in the anomaly detection, whereas techniques such as Chi- square, Recursive Feature Elimination (RFE) can be used to determine which feature is the most important and reduce dimensions for better performance.
6) Dimensionality Reduction: PCA or autoencoders reduce the feature set size while keeping the important information. This makes the model more efficient
7) Encoding Categorical Features: Network data usually consists of categorical features like protocol type: TCP or UDP. Such features need to be encoded as numbers for machine learning models. Techniques that are often used are One-Hot Encoding or Label Encoding.
8) Handling Imbalanced Data: Network anomalies are relatively scarce compared to normal traffic, giving rise to unbalanced datasets. Oversampling (for example, SMOTE) or under sampling can be used to balance the dataset.
9) Cost-sensitive learning: In some cases, one can tweak the algorithm with the incorporation of providing more weights to the minority class (anomalies) in order to deal with imbalance.
10) Time-Series Processing: Since network traffic data typically has a time component (i.e., timestamps) attached, preprocessing at times can transform raw data into some form of time-series representation. Some techniques include: Aggregation over windows in time such as seconds or minutes.
11) Feature building using time (e.g., spiky traffic, session length).
12) Noise Reduction: Noisy data or features that are ignorable may be filtered through techniques like smoothing to focus on meaningful patterns. Summary:
13) Data collection: Packet, flow and log data collection from various network sources
14) Data preprocessing: Data cleaning Normalization Feature Selection Handling imbalanced data Encoding categorical features Preparation of time series data.

C. Exploratory Data Analysis (EDA)

Plotting the data: Plot plots and graphs to identify potential patterns of traffic, correlations, and outliers in the dataset.
Statistical Analysis: analyze distribution properties of attributes, time-series trends, and correlation among features.
Anomaly finding: Identify preliminary anomaly patterns by using clustering algorithms or statistical methods for better insight into the data structure.

D. Model Selection

Choose anomaly detection method:
Supervised Learning: Based on labeled data, using Random Forest, SVM, or Decision Trees.
Unsupervised Learning: detecting anomalies when no labeled data exists; examples include Isolation Forest, DBSCAN, k-means
Semi-supervised Learning: little labelled data used along with unsupervised technique; one example is One-Class SVM

E. Model Training

Split data: Training, validation, and test sets.
Hyper parameter tuning: Method to look for optimal values of model parameters through either a grid search or random search.
Cross-validation: The method checks the models whether they generalize well on unseen data.

Model Evaluation: It uses measures like accuracy, precision, recall, F1 score, and ROC-AUC, to measure performance. Anomalies detection scenarios are of great importance towards precision (avoid false positives) and recall.

F. Model Testing and Validation

Test the model: Test dataset is used for validating the performance of the model.
Overfitting Detection: Overfitting check to see if the model overfits by looking in training and test performance.
Validation on realworld data: Let the model see real data in simulated or real world network environment.
Iterate: Refine the model by going back to feature engineering or taking another algorithm

G. Model Monitoring and Upkeep

Monitor performance: Monitor model performance over time on real-time data. Track false positives and false negatives.
Re-train the model: At periodic intervals re-train the model with new data in case of changes in network patterns or emerging threats.
Feedback loops for network administrators to improve detection overtime

VII. RESULTS AND CONCLUSION

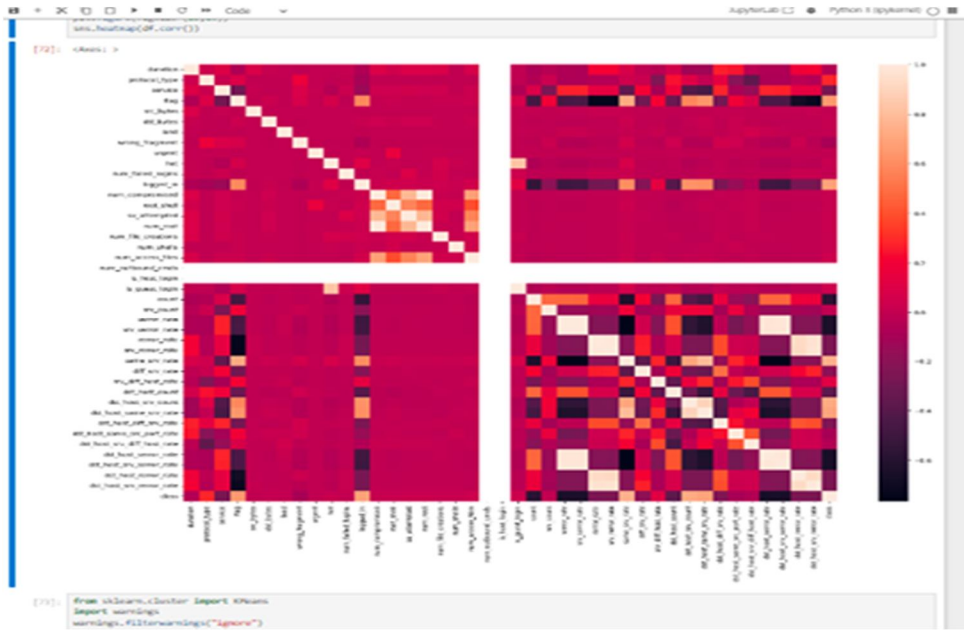


Figure 1: correlation matrix formed by the give data

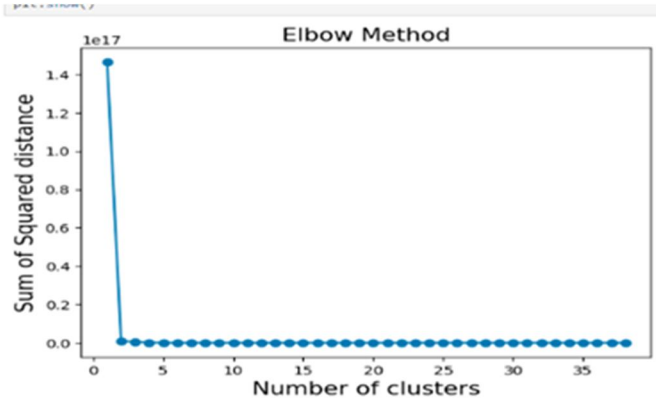


Figure 2: K-Means Elbow Method

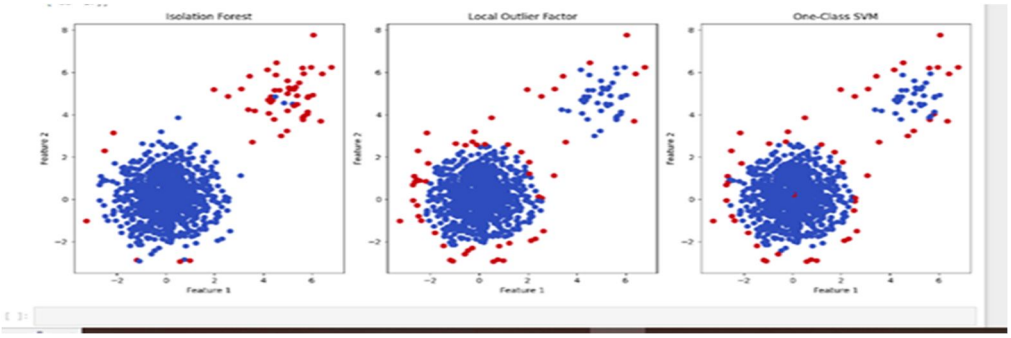


Figure 3: Visualize the results of Isolation forest, Local Outlier Factor, One-Class SVM

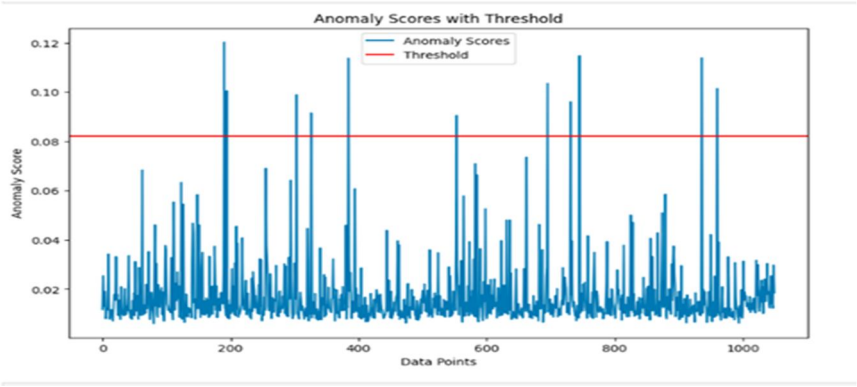


Figure 4: Plotting the anomaly scores and the threshold

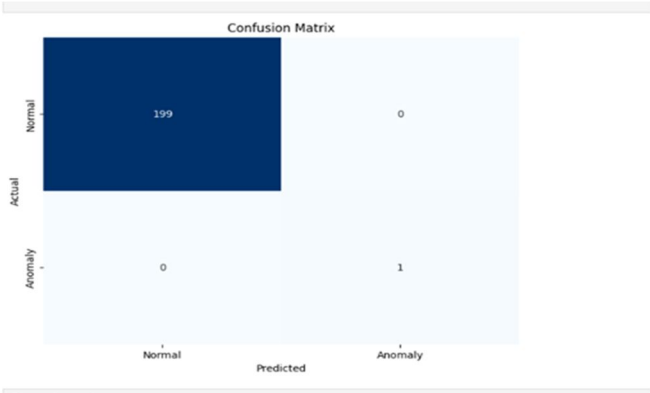


Figure 5: Confusion Matrix

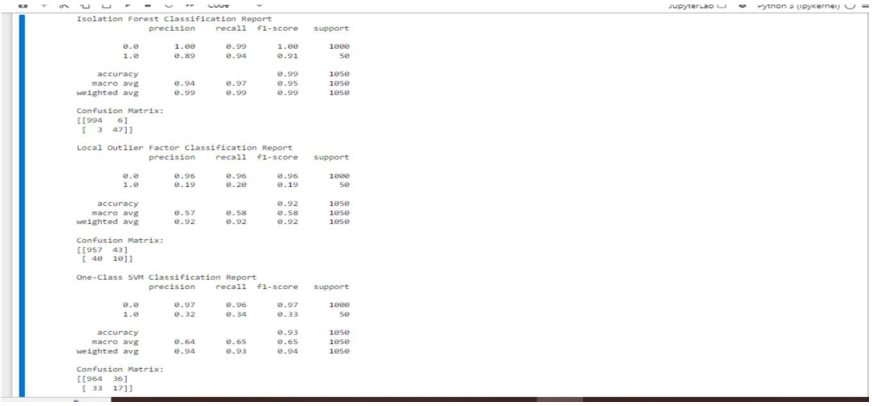


Figure 6: classification report for Isolation forest, Local Outlier Factor, One-Class SVM

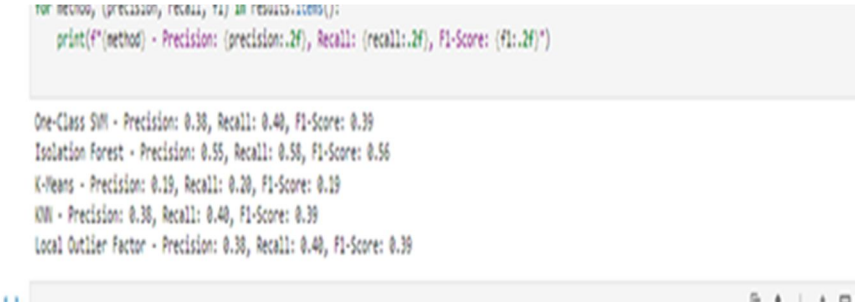


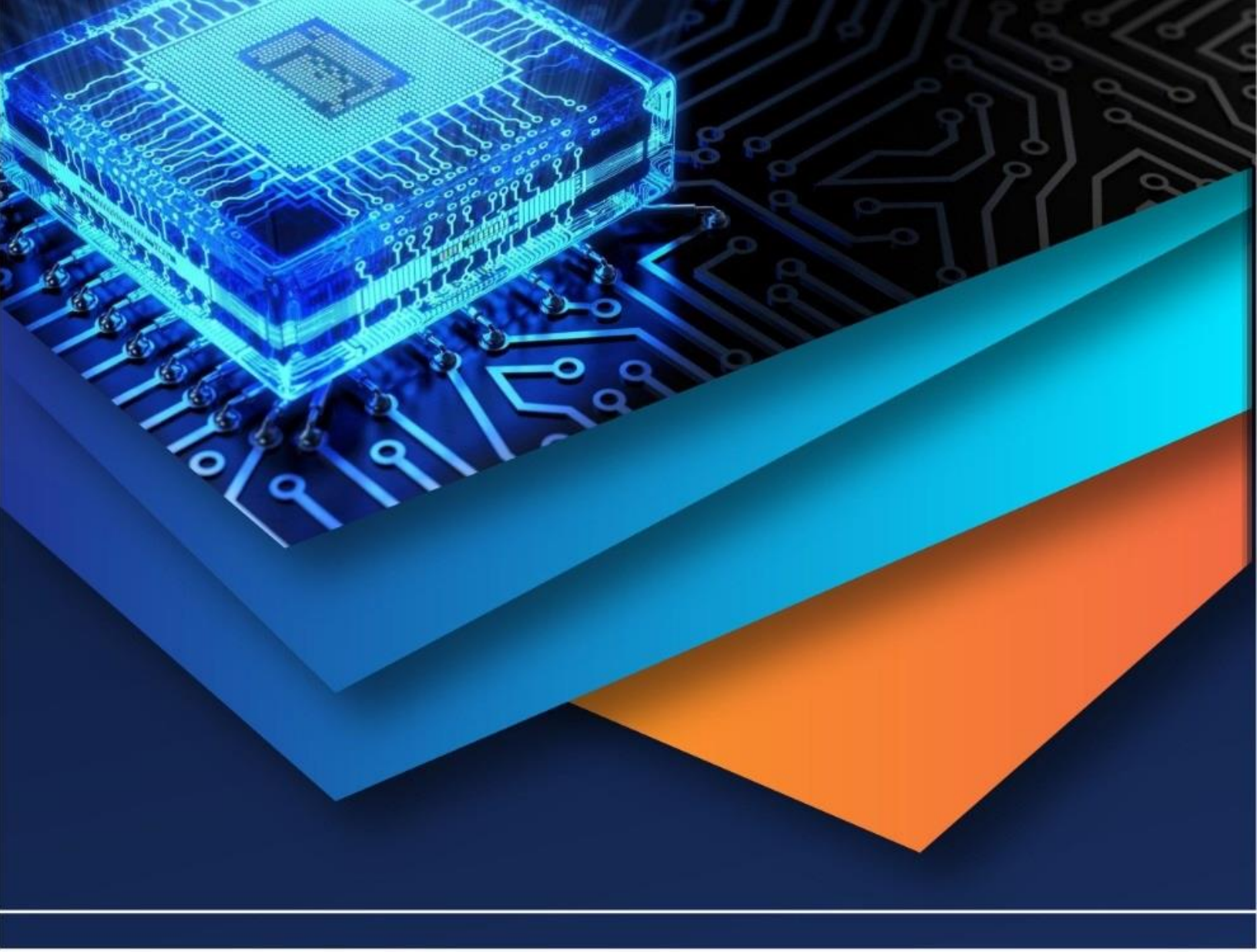
Figure 7: results of evaluation metric precision, recall and F-1 score

VIII. CONCLUSION

Machine learning-based anomaly detection in network traffic has been of great use in identifying possible security threats as well as performance issues and network failures. By using supervised, unsupervised, or semi-supervised learning, machine learning algorithms can thus identify anomalies in usual network traffic for possible malicious activities like a Distributed Denial of Service attack, intrusion attempts, or unusual user behavior. The execution of these models can monitor in real-time with increased precision and is capable of processing large-scale network data efficiently. As more advanced algorithms and deep learning techniques have been released into the market, it is expected that the ability to detect known and unknown, or zero-day threats, becomes more robust. But there are challenges in the interpretability of the models, handling evolving streams of data, and high false-positive rates that have to be removed

REFERENCES

[1] Chandola, V., Banerjee, A., & Kumar, V. (2009). "Anomaly detection: A survey." ACM Computing Surveys (CSUR), 41(3), 1-58.
[2] Sommer, P., & Paxson, V. "Outside the Closed World: On Using Machine Learning For Network Intrusion Detection." 2010 IEEE European Symposium on Security and Privacy.
[3] Ahamed, M., Mahmood, A. N., & Hu, J. (2016). "A survey of network anomaly detection techniques." Journal of Network and Computer Applications, 60, 19-31.
[4] Tavallaee, M., Bagheri, A., Sharif, B., & Hamdi, M. (2009). "A detailed analysis of the KDD CUP 99 data set." 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications.
[5] Yin, C., Zhang, Z., & Yang, Y. (2018). "Deep learning for network anomaly detection: A review." IEEE Access, 6, 65625-65638.
[6] Liu, Y., Wu, J., & Liu, W. (2018). "A novel hybrid method for network anomaly detection using convolutional neural networks." IEEE Transactions on Neural Networks and Learning Systems.
[7] Khan, M. A., & Alzubaidi, J. (2021). "Machine Learning Approaches for Network Intrusion Detection: A Survey." IEEE Access, 9, 151572-151596.
[8] Hodge, V. J., & Austin, J. (2004). "A survey of outlier detection methodologies." Artificial Intelligence Review, 22(2), 85-126.
[9] Buczak, A. L., & Guven, E. (2016). "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection." IEEE Communications Surveys & Tutorials, 18(2), 1138-1159.
[10] Tavallaee, M., et al. "Toward the Evaluation of Intrusion Detection Systems." 2009 1st International Conference on Computer and Electrical Engineering.
[11] Zhou, W., et al. "Anomaly Detection in Network Traffic Based on Convolutional Neural Network." IEEE Access, 8, 175246-175259.
[12] Makhdoom, I., et al. 2020. "Network Anomaly Detection Using Deep Learning: A Review." IEEE Access 8, pp. 148223-148238.
[13] Rashid, M. T., et al. 2021. "An Enhanced Machine Learning Approach for Network Intrusion Detection." International Journal of Computer Applications, 975, 8887.
[14] Sahu, S. K., et al. "A Review of Machine Learning Approaches for Network Traffic Analysis and Anomaly Detection." Journal of King Saud University - Computer and Information Sciences.
[15] Gonzalez, S., et al. "Exploring Deep Learning Techniques for Anomaly Detection in Network Traffic." Journal of Network and Computer Applications, 217, 103331.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY