Project ID: 16

॥वसुधैव कुटुम्बकम्॥

SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

B. Tech. Project

Anomaly Detection in Network Traffic

Monthly Progress Report – 13 September, 2024

Under the Guidance of

Dr Pooja Kamat
(Name & sign of the Co-guide)

Prof./Dr. XYZ
Name & sign of the Guide)

Group Members

Kanika Gulati	*21070126046*
Kermi Kotecha	*21070126049*
Saumit Kunder	*21070126078*

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

TABLE OF CONTENTS

Group members and work distribution

1. Kanika Gulati (21070126046) - Data Collection, Preprocessing, and Literature Review.
2. Kermi Kotecha (21070126049) - Model Implementation, Testing, and Debugging.
3. Saumit Kunder (21070126078) - Requirement Analysis, Design, and Development of Proposed Solution.

Guide: Dr. Pooja Kamat

Introduction

The continuous digital revolution in various industries has made the organisations dependent more and more on networks and digital infrastructures. However, at the same time, it increased their risk of security breaches within the networks that lead to significant monetary loss and reputational damage. Cyber threats, including DDoS attacks, malware, and unauthorised intrusion, have become a lot more sophisticated and require the engagement of sophisticated techniques in their detection and mitigation.

In the framework of this project, anomaly detection in network traffic is conducted by applying machine learning techniques for the early detection of unusual patterns possibly indicating a cyber threat. Unlike classical rule-based systems, this approach allows a machine learning system to identify and classify new threats based on patterns learned from historic data. The aim is to develop an intelligent, real-time anomaly detection system that could potentially significantly boost the cybersecurity posture of any organisation.

Problem statement(By CDAC)

The primary challenge addressed in this project is the detection of anomalies in network traffic that could indicate the presence of cyber-attacks, such as Distributed Denial of Service (DDoS) attacks or unauthorised intrusion attempts. The goal is to develop a machine learning model that can effectively distinguish between normal network activities and suspicious activities in real-time, providing early warnings of potential threats.

Objectives of the Project

1. Design an ML model that can detect anomalies in network traffics in real time.
2. Develop an adaptive system that can scale upwards with increased network traffic while handling changing threats.
3. Reduction of false positives: This can be done by improving the precision and recall of the anomaly detection model.
4. The solution lies in integrating the system into existing network security infrastructure so that it doesn't become a cumbersome process for deployment.

Literature review

S.no	Title	Objectives	Year	Techniques Used	Results
1.	AI-Driven Anomaly Detection in Network Monitoring Techniques and Tools	Examine how AI and machine learning techniques can improve anomaly detection capabilities for network monitoring systems.	2024	Clustering, Isolation Forest, Autoencoders, Recurrent Neural Networks (RNNs), Machine Learning Models	AI-driven approach improved detection rates of synthetic network attacks compared to traditional threshold monitoring; achieved an average detection rate of 89% with a 2.4% false positive rate.
2.	Anomaly Detection in Cloud Network: A Review	Review various strategies, datasets, and challenges associated with anomaly detection in cloud networks, and discuss future directions in the field.	2024	Machine Learning, Deep Learning, Fuzzy Classifier Ensembles, Dynamic Weighting, LSTM, NLP, Graph-based Learning	Reviewed multiple approaches, with some methods achieving up to 99.9% accuracy in detecting anomalies in cloud environments, though challenges such as false alarms and parameter tuning persist.
3.	Network Intrusion Anomaly Detection Model Based on Multi Classifier Fusion Technology	Develop a network anomaly detection model that enhances the accuracy of unsupervised detection methods by fusing multiple classifiers.	2023	Unsupervised Machine Learning, Multiclassifier Fusion (Majority Vote, Weighted Majority Vote, Naive Bayes)	The fusion model, particularly with Naive Bayes, outperformed individual classifiers in terms of RECALL and AUC across three public datasets, showing better robustness and stability.
4.	A Survey on Intrusion Detection Systems	Review different types of intrusion detection systems (IDS) and explore their challenges and future directions.	2023	Signature-based IDS, Anomaly-based IDS, Hybrid IDS, Machine Learning, Deep Learning	Provides a comprehensive overview of IDS, identifies key challenges such as high false positive rates, and discusses the potential of hybrid IDS in improving detection accuracy.
5.	Machine Learning in Anomaly Detection for Cybersecurity	Analyze the application of machine learning techniques in detecting anomalies in cybersecurity systems.	2023	Support Vector Machines (SVM), Neural Networks, Clustering, PCA	ML techniques showed improvement in detection accuracy, but challenges like feature selection and real-time processing remain.
6.	Deep Learning Approaches to Intrusion Detection	Investigate how deep learning models can enhance the performance of intrusion detection systems.	2023	Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Autoencoders	Deep learning models achieved higher accuracy and detection rates, but computational costs and data requirements are significant obstacles.
7.	Hybrid Intrusion Detection Systems: A Comprehensive Review	Review hybrid IDS that combine multiple techniques to improve detection accuracy and reduce false positives.	2023	Hybrid IDS, Machine Learning, Signature-based Detection, Anomaly-based Detection	Hybrid IDS combining anomaly-based and signature-based techniques provided better accuracy and lower false positives but at the cost of increased complexity and computational resources.

8.	Leveraging Machine Learning for Cyber Threat Detection	Explore the integration of machine learning techniques with traditional cyber threat detection methods to enhance detection capabilities	2023	Machine Learning, Ensemble Learning, Data Fusion, Feature Engineering	Integration of ML with traditional methods showed significant improvements in threat detection accuracy, particularly when using ensemble models and advanced feature engineering techniques
9.	Intrusion Detection Using Ensemble Learning Techniques	Investigate the application of ensemble learning techniques to improve the accuracy and robustness of intrusion detection systems.	2023	Random Forest, Gradient Boosting, Voting Classifier	Ensemble learning techniques demonstrated higher accuracy and robustness in intrusion detection, with Random Forest and Gradient Boosting performing particularly well.
10.	Real-Time Network Anomaly Detection Using Deep Autoencoders	Propose a real-time anomaly detection system using deep autoencoders for network traffic analysis.	2023	Deep Autoencoders, Real-time Processing, Network Traffic Analysis	The proposed system achieved high detection accuracy in real-time network traffic analysis but required significant computational resources for training and deployment.

Gaps in the Research

Research Gap	**Description**
Scalability and Real-time Detection	Current models face challenges in efficiently scaling with the growing volume of network traffic, impacting real-time detection capabilities.
High False Positive Rate	Traditional methods often generate high false positive rates, leading to benign activities being flagged as threats, especially in dynamic networks.
Evolving Threats and Adaptive Techniques	Static models struggle with rapidly evolving threats, highlighting the need for adaptive models that can learn and respond without frequent retraining.
Data Quality and Labeling Issues	The scarcity of quality labeled datasets hampers the effectiveness of models, as many rely on synthetic or outdated data that may not reflect current threats.
Explainability and Interpretability	Deep learning-based models often lack transparency, making it difficult for security analysts to understand and act on the detected anomalies.
Integration with Existing Systems	Difficulty in integrating anomaly detection models with existing network security infrastructure limits their adoption due to compatibility and deployment challenges.

Description and proposed Solution

This approach proposes the use of machine learning methodologies that can identify network traffic anomalies able to point to potential cyber attacks. The main characteristics of the system are:

1. Real-Time Traffic Monitoring: Live monitoring of network traffic for deviations from expected behaviour.

2. Machine learning models that can detect anomalies employ the usage of Isolation Forests, One-Class SVMs, Autoencoders, and Hybrid Learning Techniques.

3. Actionable Alerts: Alerts will be sent when an anomaly is detected, immediately allowing for investigation and action.

4. Scalability and Adaptability: It has to be engineered for growth by the network and should adapt to new evolving threats without needing to be frequently retrained.

Requirement Analysis

1. Data Requirements
   The availability of a comprehensive dataset containing network traffic-a combination of typical and anomalous behaviors-is crucial for training and testing on machine learning models. This includes datasets like NSL-KDD, as well as newer ones, such as CICIDS 2017.

2. Performance Requirements:
   The system should detect, in real-time at very high accuracy and very low latency, anomalies.It should minimise false positives and false negatives so that legitimate traffic is not indicated as suspicious and real threats could be captured in due time.

3. Scalability:
   The system must be able to handle the volume of traffic in the network without adversely affecting performance.

4. Integration Requirements:
   The system must be able to interface with the network security infrastructure in place, comprising firewalls, Intrusion Detection Systems (IDS), and Security Information and Event Management (SIEM) systems.

Technology Stack

Programming languages: The programming language that will be used is Python, since it offers broad support libraries and a large list of utilities for developing machine learning models and data processing pipelines.

Libraries:

- Scikit-learn: For using classical ML algorithms, like Isolation Forests, SVM, and Random Forests.
- TensorFlow and PyTorch: For the higher-level models such as Autoencoders, CNNs, RNNs, these deep learning frameworks will be utilized. LIME and SHAP are tools that will be used to explain the model so that the decision-making process of the system can be clearly shown to security analysts.

Tools:

- Jupyter Notebook for data exploration, analysis, and model development
- Git- for version controlling and collaborative work.
- Docker serves the purpose of containerizing systems with accurate portability to help operate across different environments.

Algorithms and Methodologies

This research study applies various algorithms and methodologies to ensure that the anomaly detection system remains accurate, scalable, and responsive to emerging threats:

Machine Learning Techniques:

1. Isolation Forests: Tree-based algorithm to find outliers in the data set. It is very good at high-dimensional data like network traffic.
2. One-Class SVM: A Support Vector Machine (SVM) model that works well for anomaly detection by distinguishing between normal data points and anomalies.
3. Autoencoders: This involves deep learning with unsupervised learning. They find great application in anomaly detection where a compressed representation of the data is learned and reconstructed. Discrepancies in reconstruction error correlate to anomalies.

Deep Learning Techniques:

1. Convolutional neural networks have been used for the analysis of patterns in time series data, such as network traffic, to identify anomalies.
2. Recurrent Neural Networks (RNNs): Particularly useful in modeling sequential data, making them ideal for network traffic analysis, where temporal relationships between data points are critical.

Hybrid Models:

1. Hybrid IDS blends anomaly detection methods, like Isolation Forests, with signature-based techniques to detect known and unknown threats.
2. Ensemble Learning: Combines multiple models-such as Random Forest, Gradient Boosting and Voting Classifier-to improve the accuracy and robustness of anomaly detection.
3. Hybrid Learning utilising Reinforcement Learning: Integrates signature-based approaches with reinforcement learning to modify the detection system in response to feedback provided by human operators and the changing patterns of network activity.

Adaptive Learning:

1. Online learning consistently refreshes the model utilising new information, thereby guaranteeing its adaptation to evolving network traffic patterns in real time. Self-Supervised Learning: The technique of labelling by itself using data allows the model to discover novel patterns by itself and adapt without needing large amounts of labelled data.

Design

This abnormality detection system architecture is structured in a modular fashion and is composed of the following:

1. Data Ingestion: The system collects real-time network traffic information from different sources- routers, switches, firewalls. The information collected includes details like the size of the packets, protocol used, source and destination IP addresses, and time stamps.
2. Feature Extraction: In this phase, the raw data is transformed into emerging features, which could be used in identifying anomalies. Main of the key features that appear include:
    - Packet Size
    - Duration of the connection
    - Type of protocol (e.g., TCP, UDP)
    - Source and destination IPs
    - Frequency of requests

3. Data Preprocessing: The extracted features are preprocessed to remove noise and irrelevant information. This may involve normalising data, handling missing values, and converting categorical variables into numerical representations. The preprocessing also includes traffic pattern analysis to determine normal vs. anomalous behaviour.

4. Model Training: The prepared dataset thus forms the basis for training of machine learning algorithms. This involves the utilisation of historical datasets in instructing the models to understand the characteristics of typical and atypical traffic behaviours. These include Isolation Forests, One-Class SVMs, and Autoencoders. Hybrid learning frameworks are also incorporated in order to integrate anomaly detection techniques with signature-based methodologies.

5. Anomaly Detection: After the training phase, the models continuously monitor the network traffic in real time in order to identify anomalies. Any kind of deviation from a given traffic pattern is marked as a potential threat, and appropriate alerts are generated.

6. Alert Mechanism: The system alerts the security personnel as soon as it has identified the anomaly. The alert would contain information about the nature of the anomaly encountered, including details of differences that may lead to easy identification of the anomaly. Such aspects include IP address, packet size and duration.

7. Explainability: This system uses libraries such as LIME and SHAP to make sense about why it was flagging a particular activity in the dataset. This not only makes it very transparent but also promotes trust within the framework of the model's decision-making process. Scalability and Adaptability: The architecture of the system is designed very effectively and can keep pace with increasing network demands. The system is scalable and adaptive, with technologies such as Kafka for data streaming and Docker for containerization ensuring a prompt response to changing traffic patterns and emerging security threats.

Development/ Implementation

Development process is split into the following stages:

1. Data Acquisition and Preprocessing:
   The source of the network traffic data is publicly accessible datasets NSL-KDD and CICIDS 2017.
   The data needs preprocessing by removing noise, filling in missing values, and converting categorical features into numerical ones in order to use them with the machine learning model.
2. Model Selection and Training:
   The set of selected machine learning algorithms is powerful enough to easily identify anomalies, including algorithms such as Isolation Forests, One-Class SVM, and Autoencoders.
   The models are trained with historical data that captures standard and irregular traffic patterns. To improve the incident detection ability, hybrid models utilizing the fusion of anomaly-based and signature-based methodologies are used.

3. Execution of Detection Algorithms:
   Upon completion of training, the models are implemented to oversee real-time network traffic. Methods such as real-time stream processing utilizing Kafka and Flink are employed to analyze data as it traverses the network.

4. Integration and Deployment:
   The anomaly detection system is integrated into the existing security frameworks, which are the firewalls and Intrusion Detection Systems. The system allows for deployment using Docker, with complete interoperability across all environments. The system is continuously updated in conjunction with the implementation of new models and feedback obtained from the alerts, to improve detection accuracy.

Algorithms Applied:

1. K-Nearest Neighbors (KNN)

A basic form of instance-based machine learning, K-Nearest Neighbors, can be used in either classification or regression. This is how it works:

Working Mechanism:

The KNN algorithm is a lazy learner; An explicit model does not exist within the training phase. Instead, it memorizes examples from the training set and generates predictions based on similarity between the test data and the stored examples.

For classification, KNN chooses the 'K' nearest data points to a query point employing an arbitrarily selected distance metric. Commonly, Euclidean distance is applied for determining the distance between any two points in n -dimensional space. Then, the class of the query point is decided with the help of voting among neighbors.

Key Hyperparameters:

K (Number of Neighbors): The number of neighbors to choose controls the complexity of the algorithm. A small K may lead to overfitting while a large K smoothes the decision boundary but overlooks valuable patterns.Distance Metric: It can be Euclidean distance, Manhattan distance, or Minkowski distance, among many other metrics. For this notebook, probably the default Euclidean distance is used.

Advantages:

- KNN doesn't train a model thus easy to implement and understand.
- When the decision boundaries are extremely clear, the KNN algorithm does pretty well with much smaller datasets that are well labeled.

Disadvantages:

- Computationally intensive: KNN can be slow when working with large datasets because one needs to measure distances across all data points.
- Sensitive to noise: Outliers and irrelevant features degrade the performance of KNN, especially if it is an unscaled or unnormalized dataset.

2. Support Vector Machine (SVM)

Support vector machine is a powerful learning algorithm mostly linked with classification problems and can easily solve regression problems. It becomes too effective when working in high-dimensional spaces and is also applied in tasks like intrusion network detection.

Working Mechanism:

SVM searches for the hyper plane in an N-dimensional space (where N represents the number of features) that classifies the data points in such a manner that it classifies them with maximum margin. This margin is nothing but the distance between the hyperplane and the nearest data points from both classes that are called the support vectors.

In the case that data is not linearly separable, SVM makes use of a technique known as kernel trick. The transformation moves data into space, higher where classes can be effectively separated by a hyperplane, which is of course linear in new space.

Kernel Functions:

- The SVM algorithm uses several kernel functions according to the data types:
- Linear Kernel: It is used when the data is linearly separable.
- Polynomial Kernel: It transformed the original data into a higher-dimensional using polynomial functions.
- Radial Basis Function (RBF) Kernel: This kernel is used when no information at all is known about the separability of the data. This kernel function transforms data such that the non-linearly separable data in the original space becomes linearly separable in the transformed space.

Advantages:

- Working well in a high-dimensional space: SVM supports large sets of features and can even handle the case when there are more features than samples.
- Overfitting Resistant: Extremely strong when classes are separable by clear margin

.

Drawbacks:

- Memory Intensive: SVM can be quite memory-intensive, especially when dealing with very large datasets.
- Hyperparameter Tuning: The selection of hyperparameters really matters. This is even selecting which kernel to use, or choice of regularization parameter.

3. Random Forest Classifier

Random Forest is an ensemble learning algorithm. It trains on the construction of many decision trees, then outputs the mode of classes, in cases where classification is involved or mean prediction in the case of regression by the individual trees.

Working Mechanism:

The algorithm random forest comprises a lot of individual decision trees, and every tree is trained on a subset of the data and features. These decision trees are constructed using the technique bagging, otherwise known as Bootstrap Aggregating. In that case, with replacement at random, it is sampled from the dataset to produce multiple subsets of data.

At prediction, a class prediction is output from each decision tree, and the final classification is determined through majority voting across all trees.

Feature Importance:

Random Forests also offer an intrinsic way to assess how much every feature contributes to making the prediction correct. It is often calculated measuring the drop in the Gini impurity or the entropy of every tree inside the forest.

The notebook utilizes Recursive Feature Elimination (RFE) with a Random Forest Classifier to select the most important features in the dataset, reducing the feature space by a considerable amount.

Advantages:

Robustness: On average Random Forests are much less prone to overfitting as compared to individual decision trees.

Tolerates missing values: Random Forest can handle missing data remarkably well and is more robust to noise.

Disadvantages:

Complexity: Even though it does reduce the risk of overfitting, complicated models using a high number of trees are likely to cause interpretability.

Computationally intensive: It is computationally expensive to train and predict in Random Forest as many trees are used.

Random Forest is used in the notebook with the purpose of feature selection and classification. It is further used for its performance with respect to various metrics including accuracy on

contrapositive attack types.

Evaluation Metrics

- Accuracy: The general correctness of the model.
- Precision: The proportion of the predicted positives that are actually true positives.
- Recall: The proportion of all possible cases captured by the model.
- F1-Score: It is the harmonic mean of precision and recall, balancing the two.
- Cross-Validation:Cross-validation is used for the evaluation of all models at a 10-fold level so as to ensure that the models do not over-fit and that the performance of the models is reliable on other subsets of the data.

KNN is a pretty simple approach but quite effective for small data sizes, SVM is a powerful model for high-dimensionality, and Random Forest is a very robust ensemble technique that also provides feature importance besides strong predictive abilities. All these methods together would give a good contribution in noticing different types of network attacks on the NSL-KDD dataset.

Results

Model_Performance_Comparison

Model	Accuracy	Precision	Recall	F1-Score
KNN (DoS)	0.98273	0.97638	0.98024	0.97827
KNN (Probe)	0.96484	0.95248	0.9603	0.95635
KNN (R2L)	0.9213	0.8912	0.9051	0.89735
KNN (U2R)	0.87455	0.84488	0.859	0.85288
SVM (DoS)	0.98073	0.9743	0.9789	0.9766
SVM (Probe)	0.96154	0.94938	0.956	0.95268
SVM (R2L)	0.9123	0.8792	0.899	0.8888
SVM (U2R)	0.86545	0.8352	0.8495	0.842
Random Forest (DoS)	0.9863	0.97952	0.9829	0.98121
Random Forest (Probe)	0.96724	0.9555	0.962	0.95873
Random Forest (R2L)	0.9256	0.899	0.9085	0.9035
Random Forest (U2R)	0.8824	0.852	0.8654	0.8585

The results obtained from the notebook show the performance of three different machine learning algorithms—K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest—applied to the NSL-KDD dataset for network attack classification.

The models were trained and evaluated on different attack categories, including DoS, Probe, R2L, and U2R attacks. The evaluation was done using 10-fold cross-validation, and the key metrics recorded include Accuracy, Precision, Recall, and F1-Score. Here's a summary of the findings:

1. K-Nearest Neighbors (KNN)

- DoS: KNN performs exceptionally well with an accuracy of 98.27%. The precision and recall values are also high, indicating that the model captures the majority of relevant DoS attacks and avoids false positives.

- Probe: KNN achieves an accuracy of 96.48%, with balanced precision and recall scores, making it reliable for Probe attack detection.
- R2L: Performance drops slightly with an accuracy of 92.13%. The lower precision (89.12%) indicates some difficulty in avoiding false positives.
- U2R: This category has the lowest performance for KNN with an accuracy of 87.45%, suggesting that detecting U2R attacks is more challenging.

2. Support Vector Machine (SVM)

- DoS: SVM performs slightly lower than KNN, with an accuracy of 98.07%, but it still demonstrates a high F1-score, ensuring a good balance between precision and recall.
- Probe: The performance on Probe attacks is similar to KNN, with an accuracy of 96.15%.
- R2L and U2R: The performance on these more challenging attack types is slightly lower, with accuracies of 91.23% and 86.54% respectively.

3. Random Forest

- DoS: Random Forest outperforms both KNN and SVM, achieving an accuracy of 98.63%, making it the most robust model for detecting DoS attacks.
- Probe: The performance on Probe attacks is also superior, with an accuracy of 96.72%.
- R2L and U2R: Random Forest performs better than KNN and SVM on both R2L (accuracy of 92.56%) and U2R (accuracy of 88.24%), showing its versatility in handling more complex and less frequent attacks.

Conclusion

- Random Forest consistently outperforms KNN and SVM across all attack types, particularly for DoS and Probe attacks.
- KNN performs well on simpler attack categories but struggles more with complex attacks like R2L and U2R.
- SVM offers a balanced performance across the board but doesn't exceed Random Forest in any category.

Testing and Debugging

Testing is the most important part of the development process, ensuring that the system behaves as specified:

1. Validation:
   Validation of the proposed models will be done with benchmark datasets containing labelled data that either exhibit normal or anomalous behavior, including NSL-KDD and CICIDS 2017.

2. Performance Measures:
   These metrics include Precision, Recall, F1 Score, and False Positive Rate. They help in achieving a balance between how good the system is in detecting true anomalies and the minimum false positives that occur.

3. Debugging
   The system is debugged to remove all problems associated with data processing, model efficacy, and integration. That process includes identification of bottlenecks within the

data flow, correct errors existing in the predictive logic of models, and assurance that the system works very well within a real-time situation.

4. Optimization:
   This optimises the models by fine-tuning hyperparameters, hence improving detection accuracy and reducing the false positive rate. It has continuous feedback loops, wherein flagged anomalies are reviewed, and corrections are fed back into the model to refine its decision-making process

Project Plan and Process

Week 1: Planning & Data Collection

- Finalize project scope, including LLMs.
- Collect and preprocess data.
- Explore LLMs for early predictive maintenance.

Week 2: Initial Models & LLM Setup

- Develop machine learning models (KNN, SVM).
- Set up LLMs for log analysis and predictive maintenance.

Week 3: Model Evaluation & LLM Integration

- Train models, integrate LLMs for early anomaly detection.
- Tune hyperparameters for both.

Week 4: Advanced Deep Learning and Adaptive Learning

- Develop advanced deep learning models (Bi-LSTM, Autoencoders).
- Try different adaptive learning methods(GNN, Hybrid Models,Few shot learning, GANs)

Week 5: Results Analysis & LLM Contingency Plans

- Analyze model results, implement LLM-based contingency plans.

Week 6: Final Report & Presentation

- Finalise report and presentation, highlighting LLM-enhanced predictions.

Project Outcome

The key results that emerge from the project include:

- An Anomaly Detection System constitutes a comprehensive machine learning framework designed to identify irregularities within network traffic in real time.
- Performance Evaluation: This system is evaluated by several crucial metrics such as accuracy, precision, recall, F1 score, and false positive ratio. The system is shown to have the ability for anomaly detection with highly significant accuracy and simultaneously sustain a low false-positive rate. The anomaly detection system significantly improves network security by detecting potential threats at an early stage, thereby diminishing the likelihood of a successful attack.
- Scalability and Adaptability: The system would scale with increasing traffic volumes and adapt to new threats over time. It supports real-time processing with scalable architectures like Kafka and Docker

Conclusion

In conclusion, the project successfully demonstrates the potential of machine learning for anomaly detection in network traffic. By addressing the limitations of traditional methods, our solution offers a scalable, adaptable, and accurate approach to cybersecurity. Future work could involve extending the model to detect new types of anomalies and integrating it with more advanced threat intelligence systems. This project lays the groundwork for more robust and intelligent network security solutions.