National College of Ireland

A Comparative Analysis of Hybrid Machine Learning Techniques for Network Intrusion Detection in Cloud Environments.

MSc Research Project
MSc Cloud Computing

Prajesh Nikhil Rajendrasubbu
Student ID: 22233784

School of Computing
National College of Ireland

Supervisor: Aqeel Kazmi

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	PRAJESH NIKHIL RAJENDRASUBBU
Student ID:	X22233784
Programme:	MSCCLOUD **Year:** 2023-2024
Module:	MSCCLOUD Research Project
Supervisor:	Aqeel Kazmi
Submission Due Date:	12/08/2024
Project Title:	A Comparative Analysis of Hybrid Machine Learning Techniques for Network Intrusion Detection in Cloud Environments.
Word Count:	8700 **Page Count:** 22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: PRAJESH NIKHIL RAJENDRASUBBU

Date: 12/08/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Comparative Analysis of Hybrid Machine Learning Techniques for Network Intrusion Detection in Cloud Environments.

Prajesh Nikhil Rajendrasubbu
22233784

Abstract

Network Intrusion Detection Systems (NIDS) within cloud environments by using and comparing machine learning algorithms. Due to the increasing number of cyber threats, there is a need to safeguard critical data in cloud environments. The existing methods such as NIDS, in particular traditional signature-based NIDS, are becoming less and less effective against advanced threats like polymorphic malware and zero-day attacks. Previous work raised attention to the problems of these conventional strategies and suggested replacements such as advanced machine learning methods and anomaly-based detection. By developing a hybrid NIDS model that combines Random Forest (RF) classifiers and Long Short-Term Memory (LSTM) networks, this study fills in these gaps. this study explores the diverse machine learning technologies using deep learning to determine an efficient and highly accurate model to detect intruders in cloud environments. The goal of this research is to develop a strong and effective network intrusion detection system (NIDS) that can navigate the intricacies of cloud environments and offer superior defence against cyber threats by utilizing the extensive UNSW-NB15 dataset. The main goal is to develop an advanced NIDS model capable of navigating cloud complexities and give a valuable enhanced model to stakeholders to help defend against cyber threats by achieving high accuracy using hybrid models and with comparison with the other Machine learning model which can lead to the development of more robust NIDS in cloud environments.

1 Introduction

Cloud Computing has a distributed software architecture the primary aim of cloud computing is to provide worldwide access to the internet for the users. In recent years Cloud computing market has seen huge growth and many new users are adopting this technology. These users access data from different parts of the globe by using the data centres. However, there is an increase in the number of users using cloud technologies so the threat to these data also increases constantly Cloud security is one of the most important aspects of cloud technologies, and this research aimed to overcome the security challenges and to make sure there is a secure connection connecting the data centres. Threats for cloud computing services arise from various sources as the intruders continuously work to find the weak spot in the network to make use of private data. To overcome this threat an effective intruder detection system is needed. This cloud evolution has also brought forth enormous challenges in the security realm, thereby warranting solutions that are as advanced to secure the data against any malevolent attacks. From this point of view, NIDS becomes the main pillar against

threats in cloud environments. Signature-based detection methods within many traditional NIDSs are decreasingly effective in contrast to modern, sophisticated cyber threats, like DoS attacks and APTs. These methods face challenges in detecting novel attacks, such as zero-day exploits and polymorphic malware, underscoring the need for more adaptive and scalable solutions.

The first step to overcome the threats is to study the existing NIDS models and how they attack the cloud environments, cyber threats like DoS and advanced persistent threats (APTs) have become more advanced there is a need for advanced NIDS system and there is a growing demand for security in cloud. The research project drives through the most effective and suitable way to approach these attacks for to enhance the NIDS capabilities in cloud environments. How the Machine learning algorithms can be used to improve security by the usage of machine learning techniques with deep learning and anomaly detection systems the aim is to study the most effective approach for NIDS in cloud environments.

Cloud Services providers and their users face many challenges in maintaining the security of the data when a huge crucial data is been stored or transmitted in the cloud it is very difficult to maintain security and there is a need to monitor and analyse the network traffic data which is tedious and challenging this research address machine learning algorithms to NIDS, the usage of Deep learning models can be used to understand the complex models to provide more security to the environment. This research will explore machine learning approaches like Random Forests, Long Short-Term Memory (LSTM), and hybrid algorithms the research aims at high accuracy and efficiency in deleting the intrusions within cloud networking. Other research has been done in coming up with different approaches to improving NIDS effectiveness. Aldallal et al. (2021) and Azam et al. (2021) also recognized the shortcomings of traditional signature-based systems and suggested that the threat detection through learning normal network behaviour may be better achieved by anomaly-based methods. The approaches still have, however, to reach the needed level of accuracy to avoid reduction in false positives.

Models proposed will be trained and validated using the popular UNSW-NB15 dataset, which is by definition thorough in representing data on network traffic. This dataset overcomes limits found in older datasets, namely KDD99 and NSL-KDD, and hence it offers a more realistic ground for the evaluation of NIDS performance. This study focuses on very high accuracy and efficiency in intrusion detection, thereby making more secure cloud infrastructures. Finally, this research has been built upon the work of past scholars and addresses gaps in the existing literature by designing an advanced hybrid NIDS model. In this model, the use of the LSTM and RF techniques with integration improves detection capabilities in the cloud security field.

In the coming section of the research will see more about the data, data preprocessing, feature extraction, training, and Machine learning modelling with a deep dive into the methodology and implementation of the proposed NIDS. A detailed analysis of the results will be presented to achieve improvements in the hybrid model approaches. By creating a

sophisticated hybrid NIDS model, this study fills in the gaps in the body of literature, building on the fundamental work of earlier researchers. Enhancing detection capabilities is the goal of integrating LSTM and RF techniques, which makes a significant contribution to the field of cloud security. The data set will be explained, and procedures will be discussed providing the foundation of the experiments in the next sections.

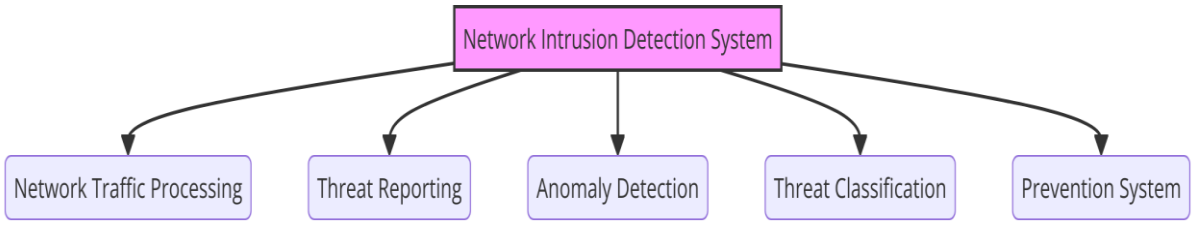

Figure 1: ***Network Intrusion detection system***

1.1 Research Questions

Threat models are subject to changes every year and can they become distinct as years go on, an algorithm is required too efficiently identify the attacks, as established models have the potential to overcome traditional methods. there are some security problems that are not addressed by the Commercial cloud providers. **How do advanced machine learning models enhance Intrusion detection for security problems that are not been addressed by commercial cloud providers, and to what extent can they improve the detection and performance in cloud computing environments?** We will use a comprehensive dataset to compare several state-of-the-art machine learning algorithms. Preprocessing the data, training several models, and assessing each model's performance using metrics like accuracy, precision, recall, and F1-score are all part of this process. Finding the most effective method for enhancing network intrusion detection in cloud computing settings is the aim.

1.2 Research Objective

- To develop an advanced machine learning model to evaluate the accuracy and effectiveness by using a contemporary dataset in identifying cyber-attacks against traditional detections to benchmark the new techniques.

- To Demonstrate the capability of the trained machine learning model to predict normal activities find the attacks and distinguish them from existing methods and provide a user interface for user to predict normal or attack with classes in real time.

2 Related Work

The literature review section carefully examines the works and studies on Network Intrusion Detection Systems (NIDS) in cloud computing. Sequentially compares various

research papers and works problems they have faced, delineating the objective and their contributions. Following a logical adaptation from the previous works in this field how it is related to the current investigation. Initially, the reviews are made around traditional signature-based NIDS methodologies. these systems are dependent on predefined patterns and these patterns are known by the attacks which make them easy and susceptible to zero-day exploits and polymorphic malware, work by Aldallal.A. et al., (2021) addresses the limitations of the signature-based detection within cloud environments, they have given insights for more adaptive and scalable solutions similar to this work Azam, I. et al., (2021) propose an anomaly-based approach, expert at detection at normal network behavior, which can recover from the novel threats. So, there is a need to find and explore an alternative method to make the system more efficient. Alzahrani, A.O et al., (2021) talks about the effectiveness of advanced machine learning techniques on the available data set to detect and classify the different types of cyber-attacks. Afterward, research work done by Kilincer, I et al., (2021) showcases the efficiency of network intrusions with good accuracy. The focus shifted to machine learning techniques to defend against the attacks of NIDS enhancements. Which includes convolutional neural networks (CNNs) and recurrent neural networks (RNNs) that can capture intriguing patterns and identify malware. Similarly, Parampottupadam.S. et al., (2018) provide new learning methods, to enhance robustness and capabilities. By comparing these methods, it is clearly understood that accuracy depends on the number of variables, network traffic, and the computational ability that are available. In this paper authors tried to develop a cloud-based prototype to investigate the capabilities of machine learning models to detect network intrusions in real life they conducted tests to compare the performance of these models with other commonly used traditional machine learning methods such as random forests by using NSL-KDD datasets.

2.1 Traditional Signature-Based NIDS Approaches.

Traditional signature-based NIDS approaches have been the Key element of network security, even though they are the bedrock of network security they face challenges when they get deployed in cloud-based environments. Alzahrani, A. O.et al., (2021). They exhibit the potency of this approach using classical and advanced machine learning techniques, on an existing dataset. Ulti-class 5 classification task is conducted by detecting attack and cataloging the type of attack (DDoS, PROBE, R2L, and U2R), they have achieved an accuracy of 95.95%. Aldallal.A. et al., (2021) clarify the challenges they faced by explaining the constraints of traditional signature-based detection in cloud environments. With threats like zero-day exploits and polymorphic malware, the durable nature of the signature-based approaches became susceptible to these threats. Javadpour. Meng and Kwok (2014) work address about the enhancing the performance of the Signature based NIDS by using an adaptive framework like the techniques like packet exclusive signature matching and filtration. This method helps in reducing the effects of false alarming and improv more the detecting rates, but they don't address the main fundamental issue of the usual signature-based systems they are being more proacting which reduce the effect of the evolving dynamic threats. A. et al., (2017) this paper suggests an anomaly-based approach that varies from the

usual pattern this method operates by detecting, therefore, it can detect novel and previously unseen threats and discuss the various machine learning methods for intrusion detection which include neural networks, fuzzy logic, and Support vendor machines, and the limitations of traditional signature-based approaches. It can struggle to identify new attacks. From this section, it is understood that there are chances to enhance the NIDS within the cloud environment.

2.2 Machine Learning Techniques for NIDS Enhancement

In the last few years, Machine learning techniques have been one of the best ways to replace the traditional signature-based NIDS approaches, offering ease of adapt and performance. Saranya. et al., (2020) highlighted the capability of convolutional neural networks (CNNs) to identify network intrusions with extraordinary accuracy. CNNs can capture and identify intriguing patterns in traffic networks. Parampottupadam.S. et al., (2018) promote collaborative learning techniques that combine several methods to improve the durability of the system. With the continuous emergence of new cyber threats, there is always a need for more effective learning models to predict and protect against many unseen attacks. This paper mentions that there is a need for further research to explore the capabilities of ML for real-time detection of intrusion. By studying both approaches this section tells the importance of selecting the most suitable based on the complex characteristics of the network traffic and computational resources available and it emphasizes the potential of machine learning for NIDS effectiveness within cloud computing environments.

LSTM and RF combinations helps to improve the strength of the models which results in more efficient system with more accuracy when compared to the isolation models in previous studies the hybrid models outperform the usual traditional single models this method of using dual model capability can fix the errors of the usual IDS which relay on the static rules that can be overcome by using this for these kind of evolving threats (Nan et al., 2023; Wei & Huang, 2020) the usage of hybrid approaches is more better in capturing the intricacies patterns of the network traffic data with more accuracy.

2.3 Challenges and Opportunities in Cloud Computing Security

There are several challenges and opportunities for network security professionals in the field of cloud computing as the threats in the cloud environment evolve every day. Ahmad, Z. et al., (2021) address the importance of real-time detection and response mechanisms in cyber-attacks. It can identify and stabilize DDoS attacks and APTs attacks which is more essential and a systematic construction of a to date dataset is enough to know about all the types of zero-day attacks. Another complexity is ML models which demand a significant amount of time and computer power resources. An efficient feature selection algorithm is required for fast processing researchers are exploring various optimization algorithms for feature selection there is always a scope for progress and development. Azam. Z. et al., (2021) suggest using big data analytics to analyze huge quantities of network traffic data to proactively identify new threats. By understanding both of their works they mention how to proceed with cloud

security by using real-time detection there is a need for new cutting-edge technologies and methods to stabilize the cyber threats in cloud environments.

2.4 Research Niche

In this research niche section, it is identified in this study lies at the intersection of network intrusion detection systems (NIDS) and cloud security. By analyzing all the insights from the paper this section highlights the gaps and ways to improve the efficiency 6 of the system and to identify more threats efficiently to enhance the effectiveness in cloud computing environments. While using the signature-based detection methods they are not good in addressing zero-day exploits and polymorphic malware, but anomaly-based approaches proved the detection of novel attacks from normal network behavior other methodologies possess limitations and need for alternative strategies.

The advent of machine learning techniques provides promising results in NIDS within cloud environments. By utilizing. Convolutional neural networks (CNNs) deep learning methods provide good accuracy in network intrusions. Similarly, new leading methods of combining multiple classifiers provide more robustness. However, selecting the most suitable algorithms and understanding network traffic and the available computational resources is the challenge.

There are always opportunities and challenges for network security professionals to subtly the threats of DDoS and APTS attacks in real-time detection and response mechanisms. By understanding these gaps this research sets a platform to enhance the NIDS effectiveness in cloud environments. With an understanding of the previous research work looking to approach and address this challenge and aim to provide a meaningful contribution to the field of cloud security thereby attempting to increase the efficiency of intrusion detection by developing novel methodologies to bridge the gap in NIDS.

3 Research Methodology

Cloud Computing has changed the entire world in how the data is handled like storing, processing, and managing with more flexibility and scalability. But with this change, there are also new challenges in security that require more advanced solutions to save the data from malicious attacks. NIDS is a solution to overcome this, this system can be designed and used to monitor the traffic and to defend from the threats.

An IDS system can be used to enhance the detection of attacks it has more capabilities to defend as it can handle more volumes of data in data centres with more network traffic also this approach can be more advanced by using machine learning algorithms that can adapt to the new attacks in bigger environments.

This research focuses on the NIDS to avoid cyber threats, especially in cloud environments there is a need to develop more robust and need for more latest security challenges for the

evolving security needs in the current environment this method studies the systematic review of already existing literature the study of the work of Alouffi et al. (2021) which tells and highlight about the current techniques used to avoid threats in this research is to fill the gaps in the current work to enhance the NIDS capabilities. The IDS system which is developed can detect cyber-attacks like DoS attacks, Generics, and Fizzers this can be used in Realtime to create a trigger to alert the administrator of the network or the data centre so that they can block the IP addresses and make the necessary update in their firewall to further prevent the attack the Framework of the IDS can make the critical components in the system to work seamlessly without any trouble the data is collected from the network to predict the patterns and attacks to the analysis they are used in various algorithms to find out the best and accurate models to predict whether the network is traffic is normal or any danger is there. This can be used in a cloud computing environment so that it is scalable in real-time monitoring by using machine learning techniques.

4 Design Specification

The Design specification for the NIDS is designed in such a way that it has a robust system for detecting and avoiding any network intrusions, clints and users need to obtain this feature to improve their networks as of now the traditional detection system suffers a lot with more false positive rate and they can identify the new threats and to even struggle for a large volume of data this design gives an outline of a hybrid approach that combines Long Short-Term Memory (LSTM) and Random Forest for feature extraction and classifications will answer the challenges faced so far. The combination of LSTM and RF classifiers in a single hybrid model gives more advantages in accuracy and efficiency. LSTM have recurrent neural network (RNN) features that can be used for sequential data because they can learn long-term dependencies this is well suited for network traffic data analysis because they can understand long-term dependencies which is used to identify the intrusions by using this in a hybrid model can increase the effect of capturing intruders and detection abilities. Random forest works by creating multiple decision trees during training for the predictions this improves the robustness of the model with accuracy which is a common issue in using other single models they are experts in capturing the important features in the data so when it comes to network data traffic this can greatly improve the model's performances. Where previous studies have shown that hybrid models have single-algorithm approaches, especially in cybersecurity (Nan et al., 2023; Wei & Huang, 2020) new research shows that using the LSTM and RF compared to other existing models can have higher accuracy.

FIG.2 Stages of the Methodology (Flow diagram Own Illustration)

The block diagram above illustrates the main stages of methodology which has been employed in this research each stage represents the important steps that are required to develop a robust Network Intrusion Detection System (NIDS) in the cloud network environments. System Architecture is made up of multiple interrelated parts, each of which is

essential to overall operation of the system. The IDS architecture can operate well in conventional settings. The first is data acquisition which involves network traffic data that contains real-world scenarios. The data set is chosen which has diverse features and is relevant to cloud network environments. the data set is retrieved from a proper and reliable source and the dataset is licensed and then only further proceeds for processing for the next stage which is the data preprocessing stage where the received dataset is cleaned and made suitable for the training missing data are addressed to maintain integrity, data is categorized here where the numerical is converted label encoding which makes it suitable for the algorithms, standardized to avoid biasing because some of the features can dominate the overall training model and affect the accuracy here so that they are brought into a common scale and the data split did here to make sure the to facilitate the model for training and evaluation. to improve the model performance feature engineering is used which plays a vital role in the model performance in this stage the key features of the model are extracted correlation analysis techniques are used to reduce dimensionality, enhancing model efficiency. Models are trained and evaluated here various models with both the traditional and hybrid are trained here are selected based on their suitability the detecting network intrusions in cloud environments models are trained on the preprocessed dataset which focuses on learning that distinguishes normal traffic from potential intrusions. The trained model is now evaluated and compared with the other models here to find out their effectiveness they are evaluated based on the accuracy, precision, recall, F1-score, and ROC analysis conducted to find out the best model here the highest models are selected, and hybrid models are made to find out the best metrics for deployment. The hybrid model is now deployed in a simple user-friendly interface which allows to find the attack or normal traffic when the data set is passed here the user-friendly interface is developed for the clients to analyze and create an alert to find the IP and the type of attack. Now in the upcoming section let’s see in detail the process is done.

5.1 Data Collection:

The data gathering layer gathers network traffic data by retrieving it from a Google Colab Drive. Because outdated datasets often pose issues for older systems, making it more difficult for them to adapt to new threats, this ensures that the training and assessment dataset is current and relevant, including many qualities required for detecting malicious behavior. The UNSW-NB15 Software-Defined Networking Network Traffic Dataset is pivotal for the training and evaluation of the IDS in detecting malicious activities within network traffic.

This dataset has a variety of features and key elements to predict different patterns of cyber-attacks. The key aspects of the dataset include various elements like source and destination IP addresses, protocol types, packet counts, byte counts, and flow durations these are used in classification tasks.

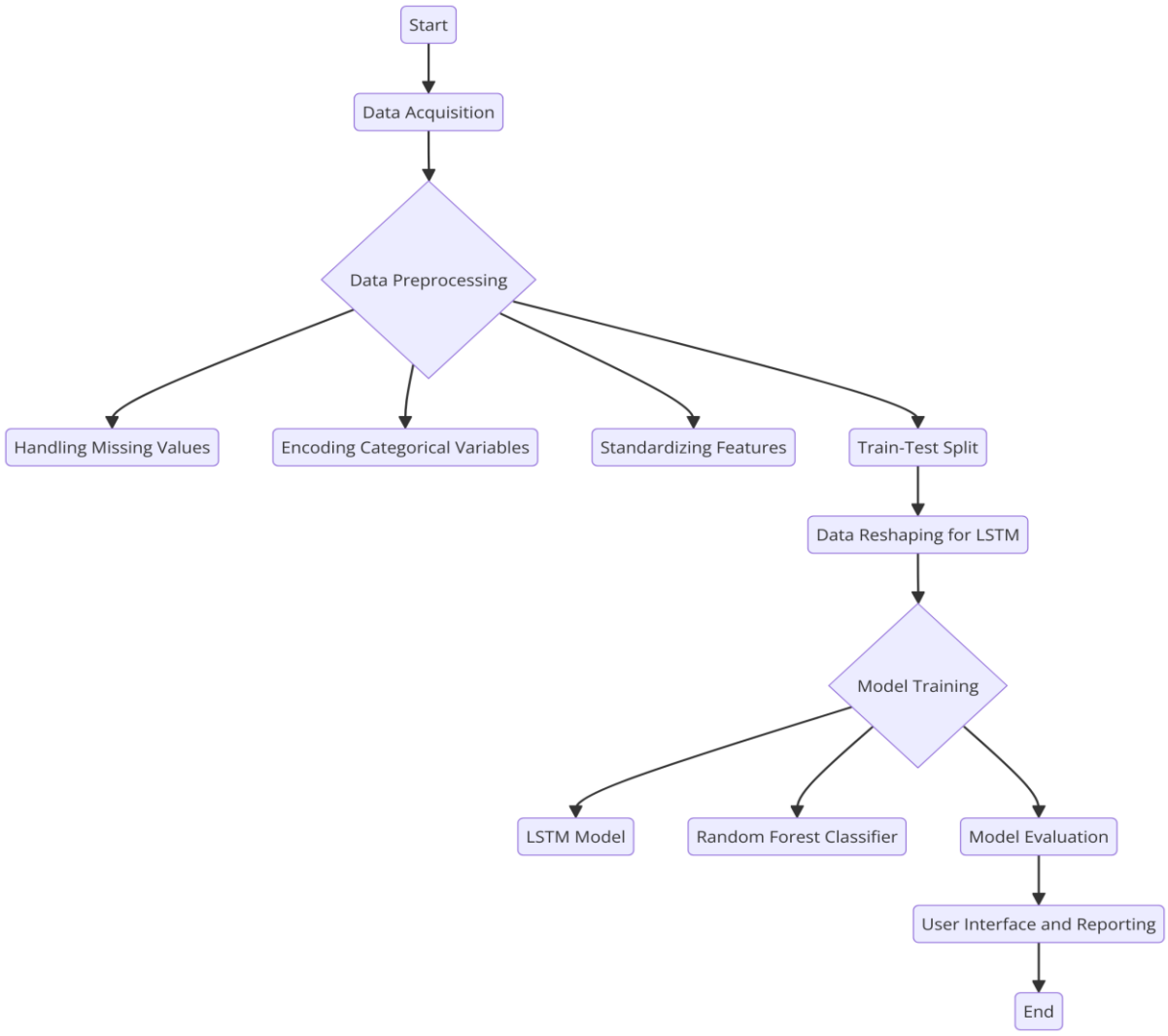

FIG.3 Methodology Flow Design Specification (Own Illustration)

5.2 Data Pre-Processing:

Performing the crucial functions of cleaning and transforming data, the Data Preprocessing Layer fixes missing values in numerical features using the mean, a reliable technique that keeps the data distribution from being distorted. Label Encoding fixes the problems faced by traditional models that are unable to handle by encoding categorical features into numerical representations. Data cleaning and pre-processing are needed to have no errors in the model that can affect the results. The data set is refined to avoid any mistakes in the steps that can cause high bias and low variance in the results. Here numerous NumPy and pandas functions were applied to get the basic data for processing the categorical variables such as proto, service, state, and attack cat, which are encoded by label-encoding. Ensuring dataset was made correct for algorithms to perform machine learning, this helps to overcome problems of previous systems which had trouble with data quality. The Data preprocessing steps are now the missing values are first eliminated by filling the values using mean they are mainly used because it is less sensitive to outliers to handle the missing values this ensures that the missing values are completed these are crucial steps to address the machine learning training

then the categorical variables such as src, dst, and protocol are encoded using lableencoder first they are converted into the numerical format and then trained the other elements in the preprocess which are packets counts, byte counts, flow durations are critical insights are provided these elements are standardized these features using StandardScaler to ensure all treated equally in the learning process. These features allow the model to understand the network activity. This process helps the model to maintain high accuracy and reduce the false positives with a more reliable detection system.

5.3 Feature Engineering:

All the features in the data must be ensured that they are on a similar scale and the data has to be standardized in the feature engineering layer. The machine learning model performance depends upon this since the earlier models frequently produced biased results by not making sure that they were in the same scale to take the feature scale into account. A significant improvement in the performance is noted when the algorithms are sensitive to feature scales, for the models such as LSTM by following this standardization for the dataset.

The two machine learning layers contain the two layers they are modeling Layer are the Random Forest classifier and the LSTM model. Long-term pattern recognition is difficult sometimes in the older models, The LSTM model addresses the issue by finding temporal relationships in network traffic data. The model is trained using the features collected to classify network data as either it is normal traffic or any kind of cyber threat where the classification accuracy is increased, and overfitting is made robust when compared to the previous models.

Precision, recall, precision, and F1-score and various criteria are used to evaluate the different layers that improve the effective layers of the NIDS. When compared to the previous models which often lacked in the evaluation metrics it made the efficacy quite a challenge now with the strong evaluation structure that helps to continuously improve the efficacy of the NIDS.

5.4 Data Processing Workflow:

To use and make sure the unprocessed data is used and converted into useful insights that can be used, which is a key component of the NIDS architecture, the data processing pipeline loads network traffic data into pandas' data frame after downloading. This approach replaces earlier ones that relied on human data collection which was time-consuming and prone to errors.

The dataset is loaded in paraquat because they are more efficient in columnar storage with faster loading and smaller file size when compared to the traditional CSV files. cleaning of data by removing unwanted data in the data like label and axis columns because they have the capability to bias the model. Furthermore, median imputation—which is less susceptible to outliers than mean imputation—is used to manage missing values during the Data Cleaning and Transformation process. Converting categorical variables into numerical representations,

this approach maintains the overall distribution of the data and is consistent with machine learning techniques. The limitations of earlier systems that had trouble with missing or improperly prepared data are resolved at this stage.

By ensuring that numerical features are standardized to maintain scale uniformity, the feature standardization step improves model performance and resolves problems with previous systems that neglected to consider feature scaling. Here, in the Train-Test Split stage, the dataset is split using an 80-20 split into training and testing sets. This indicates that the model is trained on 80% of the data, with the remaining 20% set aside for performance testing. With this, one can ensure that the model can effectively generalize to new data and avoid overfitting, which was a major issue with previous models trained and validated on the same dataset.

To meet the input main requirements of the LSTM model, both the training and testing datasets are transformed into the Reshaping Data for LSTM stage. Some of the problems that were faced previously are now fixed. The model now could learn from sequential data with good effectiveness.

Model training entails using the training dataset to train the LSTM model to identify temporal trends in the data. The Random Forest classifier offers a strong classification mechanism that overcomes the drawbacks of single-model techniques. It is trained using the characteristics recovered from the LSTM after training. Implementation of these models, using scikit-learn for Random Forest and TensorFlow/Keras for LSTM.

The Random Forest classifier's performance is now checked and evaluated using the test dataset in the Model Evaluation stage. Now the key metrics are showcased here. The drawbacks of earlier methods that used a single metric for assessment are fixed by following this evaluation framework.

5.5 Model Training & Evaluation: Performance Analysis:

To evaluate the effectiveness of the NIDS, various performance metrics are employed. Training the ML model is essential in this project when it comes to detecting attacks and generating alerts. the dataset is pre-processed as the steps mentioned in the above section to extract the exact value and relevant features needed for the project the dataset is divided into test and training with an 80:20 ratio respectively this ensures that the model has enough sufficient data to learn and to validate in the future the model has Random Forest classifier and LSTM-based deep learning models. RandomForestClassifier for sci-kit-learn is used to train the RF model, which is good in robustness and, they have the capability to handle many values without overfitting and TensorFlow's Keras API is used for building LSTM Model it could capture the data which has temporal dependencies which are used to capture detecting patterns of the threats and attack activities. They are determined by the proportion of correctly detected cases across all instances, accuracy provides a clear assessment of the model's performance. Precision is defined as the ratio of genuine positive predictions to the

total anticipated positives, which helps address the issue of false positives that plagued earlier systems. Recall compares the percentage of true positive predictions to real positives to reveal information about how well the model detects threats. The F1-Score offers a metric to assess the overall performance of the model. It is the harmonic means of precision and recall. These indicators are crucial for comprehending the NIDS's advantages and disadvantages as well. And now for the deployment Streamlit is used here for the user interface the trained model is deployed by creating a simple user interface the code should the code should reflect this through deployment scripts and possibly a web interface framework such as Streamlit in this UI where he client can pass their network dataflow details that can used to predict the network data flow data that can be used to detect the network traffic is normal or of any attack and the model also specifies the class of the attack.

5 Implementation

The proposed model aims to predict normal, or attack detected if attack is detected with its class what type of attack has been passed in the network. The proposed model undergoes various data transformations, model development, and user interface creation.UNSW-NB15 was the dataset used in this process which is mainly designed for the NIDS which is a modern dataset with an immense amount of network traffic data that was generated for network intrusion detection the missing values are handled by mean which is less sensitive to outliers all numerical features are standardized for preventing bias in the dataset that can cause disturbance in the model so that all will in the same scale to improve effectiveness in algorithms. The entire model is trained, and development was developed in the Google Colab, leveraging the cloud-based enjoinment that can offer the powerful computational resources, including GPUs. For efficiently handling the training of the deep learning models this setup is needed and crucial to train LSTM as they need high significant processing power. The dataset UNSW-NB15 is directly loaded into Google drive the code is written to read the data from the Google Drive path into the Google Colab environment this makes a seamless connection to the dataset ensuring the data remains secure and can be easily assessable through the development process. StandardScaler was utilized here from the scikit-learn library applied to all the numerical features for standardization they ensure that all features are maintained on a similar scale which is essential for the effectiveness of the LSTM model.

Multiple libraries and tools were used in the creation and implementation of the NIDS. A hybrid approach was adopted by combining the strengths of Random Forest (RF) and Long Short-Term Memory (LSTM) models. Because of the RF model's resilience and capacity to manage a high number of features without overfitting, it was chosen for preliminary feature extraction and classification tasks. TensorFlow's Keras was used for the LSTM models which is mainly suited for sequential data and network data patterns captured which is important for anomalies and intrusions. Hyperparameter tuning was carried out for Random Forest and LSTM models for the optimal performance the grid search changed key parameters like the number of trees (n_estimators), the maximum depth (max_depth), and the minimum number of samples needed to split a node. These parameters were chosen based on how they impact model complexity and ability for generalization. The Random Forest model was fine-

tuned using grid search to find the best number of trees and maximum depth. Meanwhile, with the LSTM model, a learning rate scheduling approach was used to dynamically modify the learning rate through training. these implementations improved the ability of the system to detect and classify the threats. All developments were mainly used in Python programming language with the use of libraries Pandas and NumPy for the data manipulation Scikit-learn for training the Random Forest model and TensorFlow/Keras for the LSTM model. for the LSTM Model a learning rate schedule was employed they start with set of learning, and they are reduced gradually over the time this helps in fine tuning the model's wights by accelerating convergence and during the epochs to prevent the overshooting of the minima. The trained models were saved and loaded using Joblib, which was also used to preprocess objects like scalers and encoders. The evaluation metrics—accuracy, precision, recall, and F1-score—were select to provide an in-depth evaluation of the model's performance, particularly within the context of intrusion detection, where the class imbalance is common. Precision and recall are important metrics in this scenario. Training of this model was done on Google Colab. The output attack predicted classes can be of DoS, Exploits, Fuzzers, Generic, Reconnaissance or Normal .

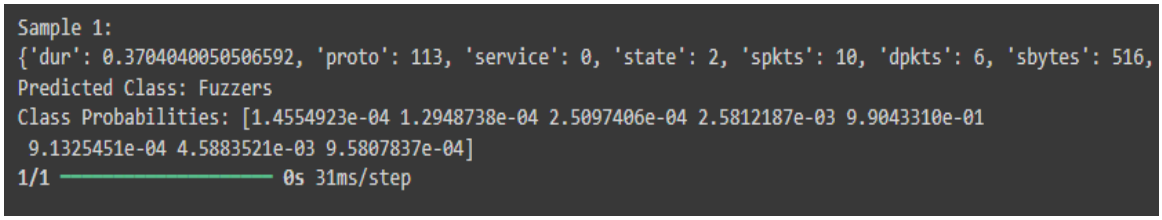

Fig 3. NIDS Predicting Attack with Class 'Fuzzers'

This figure demonstrates the NIDS identifying a 'Fuzzers' class attack in real-time here the model processes the input data given and classifies the network data as intrusion and give the class type immediately to the user in the interface.

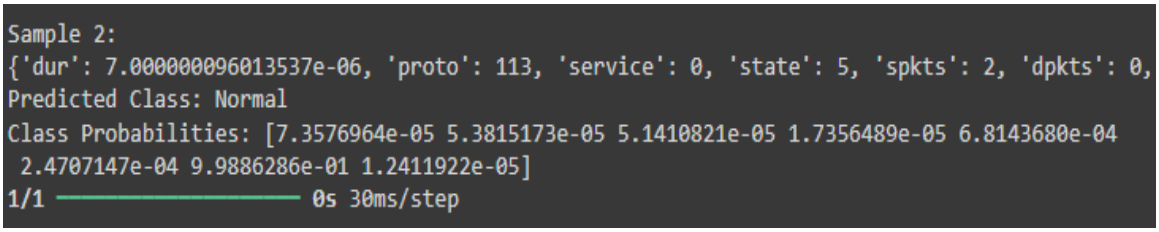

Fig 4. NIDS Predicting Normal Network Traffic

This figure demonstrates the NIDS identifying network traffic as normal. Here the model processes the input data given and classifies the network data as non-malicious and ensures threats alerts are raised only when the genuine threats occurred.

Streamlit was utilized to create an intuitive web interface that lets users pass through network traffic data and acquire predictions in real time. Streamlit is used for deployment which is a Python-based web application framework in the Google environment which is a user-friendly approach to create that is used for real time predictions on the network traffic data and practical for end-users. This interface has featured dropdown menus to enter the values for

categorical and numerical fields that can also be typed making sure non-technical and technical users have ease of use of the UI These elements were successfully integrated to create a reliable and effective Network Intrusion Detection System (NIDS) that can predict network attacks in real-time.

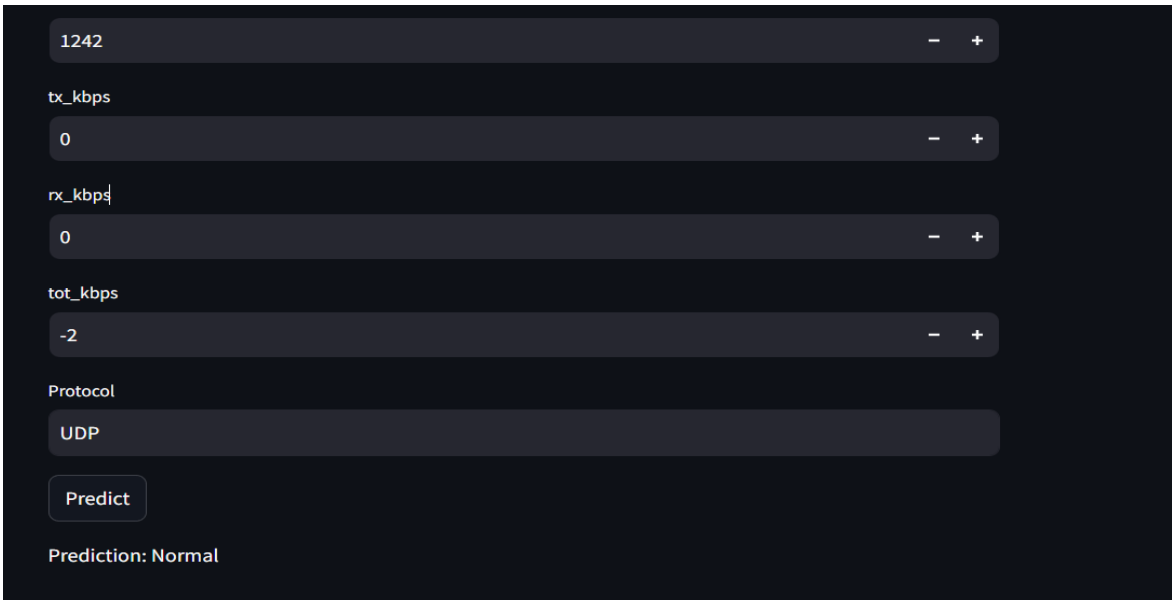

Fig 6. Streamlit UI prediction of Normal traffic

Fig 7. Streamlit UI prediction of Attack with class Exploits.

6 Evaluation

The Main purpose of this section is to an in-depth analysis of the study's primary findings and outcomes in order to support the goals of the research, evaluation entails a thorough analysis of the experimental results, statistical analysis, and visual representation of the data. The purpose of the experiments was to thoroughly evaluate the efficacy and functionality of the

suggested Network Intrusion Detection System (NIDS) the evaluations include accuracy, loss, recall, and precision scores these results were made both with the training and the test data comparative analysis of three distinct models: Random Forest (RF), Long Short-Term Memory (LSTM), and a hybrid model combining both RF and LSTM this hybrid approach is aimed to improve the accuracy of the detection by integrating the features extraction of RF and sequence learning of LSTM. 10 epochs were used to train the LSTM Model and early stopping is used to prevent overfitting with batch size of 32. The model that performs the best overall based on these metrics will be determined to be the best option for our IDS.

Performance Metrics:

In the evaluation, the model effectiveness is tested in several performance metrics which provide detailed insights into the model's capabilities Accuracy measures the overcall correctness of the predictions of the models by calculation of the ratio of correctness of the instances of true positives and negatives into the total numbers of the instances It provides an overall overview of the model's performance but might not accurately represent the model's ability to deal with class imbalances. The ratio of positives which is true to the total number positive predictions is used to calculate the Precision it is more important where the data has more false positives so they can lead to false alarms can lead to security measures which is not required. Recall that the sensitivity of the model is calculated by the ratio of positives which is true to the total number of actual positive instances which enables the model to capture all the positive cases which is important so that no threats are missed to rise to alert the mean of the precision and recall is F1-Score which has a balance of both false positives and false negatives they are used when the dataset is imbalanced it make sure that precision and recall are calculated properly as a critical measure. AUC-ROC metric is used to plot the graph here to show the model's ability which plots the sensitivity against the false positive rate at various thresholds with higher values of this metric it is said to be that the model can perform well in finding the different classes of attacks. In network intrusion data class to maintain the class imbalance precision and recall are the crucial elements that are prioritized false positives can lead to unnecessary alerts and also false negatives can miss actual threats to provide balance in the model performance F1-score and to access the overall performance of the model AUC-ROC was used.

Interpretation of Results:

The results of each experiment are analysed using the metrics to find out the effectiveness models trained. The accuracy score and AUC-ROC curve are evaluated to find the performance of the Random Forest model which gives detailed insights into their classification ability. For the sequential data in identifying the intrusions the LSTM model's precision-recall curve is used it has the capacity to identify it. Then finally the newly trained hybrid model's performance is analysed by cross comparing all the ability and metrics scores which is mainly designed to maintain high accuracy while balancing precision and recall these combinations allow the evaluation of the models and make sure that the models selected

are well performed in all metrics not only accuracy and also can manage the between false positives and false negatives.

6.1 Experiment Random Forest Model:

In this first experiment, the UNSW-NB15 dataset was used for evaluating the performance of the Random Forest (RF) model. The model was evaluated using accuracy, precision, recall, and F1-score as the main metrics. These metrics shed light on how well the model distinguished between legitimate traffic and network intrusions. In the experiment, the dataset was divided into training and testing subsets in an 80:20 ratio, and the effectiveness of the model was assessed using data that had not yet been seen.

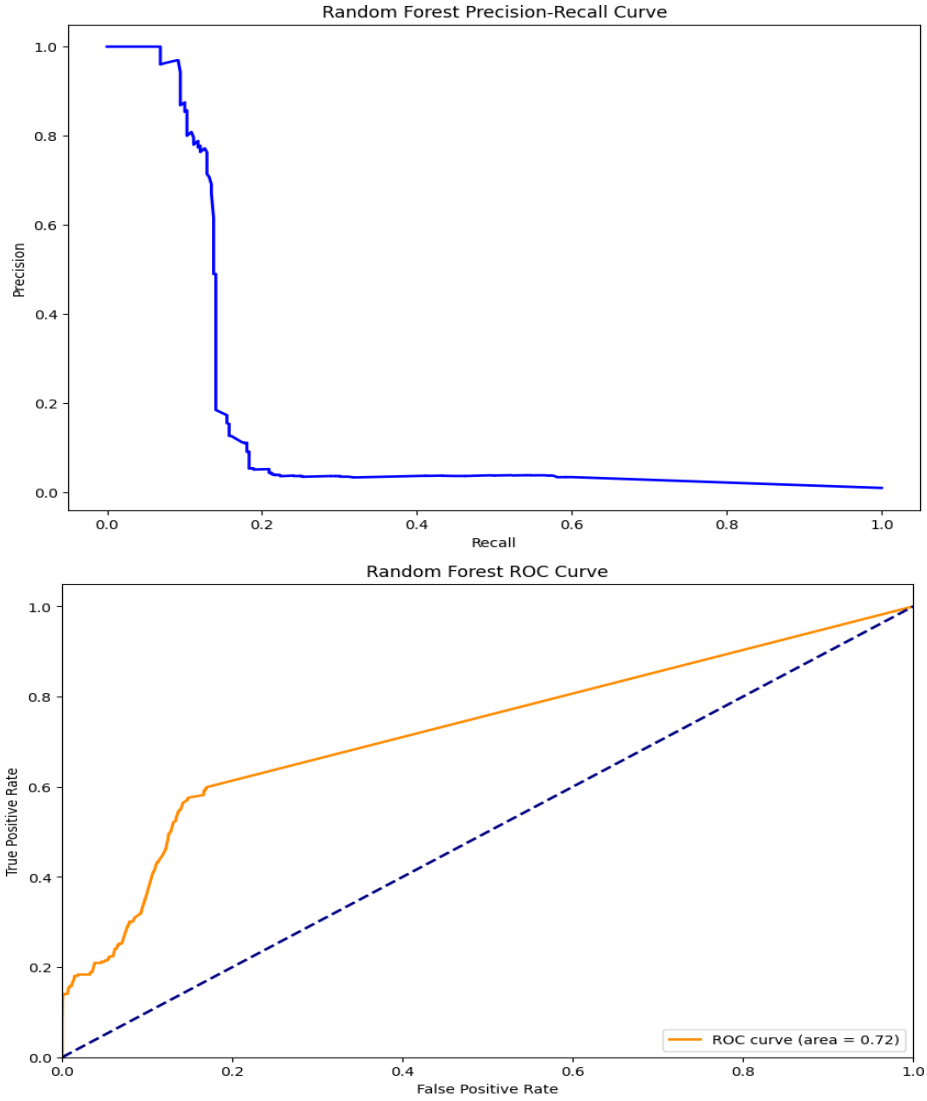

Fig 8. Recall curve and ROC curve of RF Model.

Accuracy: 81.71%, F1 Score: 80.57%, Recall: 81.71% are achieved in the model. the model was trained over 15 epochs, on every epoch the performance of each model will be calculated the results are shown here ROC Curve is used here to display the true positive rate against the false positive rate, providing an understanding of the model's diagnostic ability With an AUC

of 0.72, the Random Forest model's ROC curve shows fair discriminative ability. With a True Positive Rate (TPR) of roughly 0.7 and a False Positive Rate (FPR) of roughly 0.4, the model demonstrates a moderate level of efficacy in detecting true positives. The Random Forest model's Precision-Recall curve reveals that precision sharply declines beyond recall values of about 0.2, suggesting that the model only sustains high precision at low recall levels.

6.2 Experiment LSTM Model:

The UNSW_NB15 dataset was used to train the LSTM model for this evaluation experiment. Scaling the data and encoding categorical features were the preprocessing steps for the dataset. The LSTM model was trained over ten epochs, the loss function is categorical cross-entropy, and the primary metric is accuracy. which also has early stopping validation loss the performance has been evaluated using several metrics. Accuracy: 80.79. %, F1 Score: 77.54%, Recall: 80.79%.

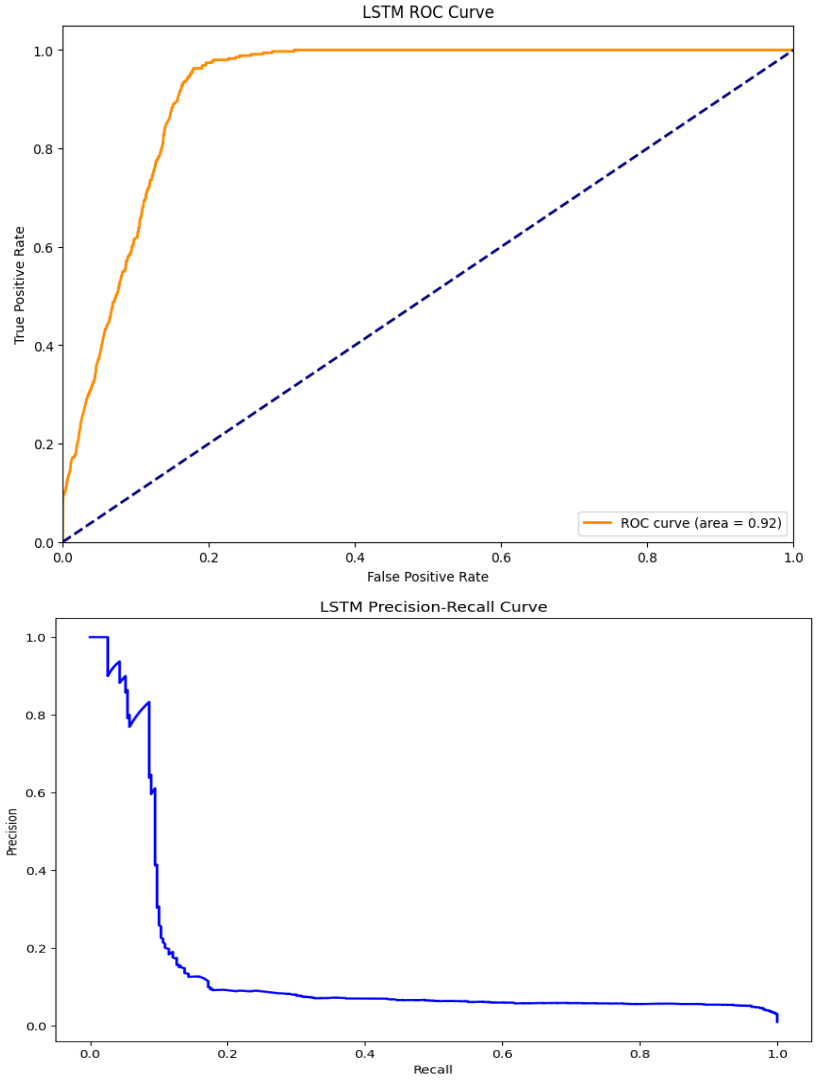

Fig 9. LSTM Model's Recall Curve and ROC Curve.

With an AUC of 0.91, The ROC curve of the LSTM model shows a strong precision capability. The model has the ability to identify with high accuracy positive cases which is

clearly evident, as it achieves a True Positive Rate (TPR) of around 0.95 while keeping a False Positive Rate (FPR) below 0.2.However, by referring the Precision-Recall curve plots a very sharp drop in precision at recall values which is around 0.1, that indicates that the model can maintain high precision only with very low recall levels . This proves that when the model can identify a small ratio of true positives, it has the high confidence in capturing the positive predictions.

6.3 Experiment Hybrid Model:

The performance of a hybrid model that combines the advantages of Random Forest (RF) and Long Short-Term Memory (LSTM) networks was assessed in Experiment 3. The hybrid model makes use of RF's ensemble learning capabilities to produce preliminary predictions that are subsequently improved by an LSTM network. By combining the sequential learning powers of the LSTM with the robustness of the Random Forest, this method tries to find a way to improve the overall efficiency with the prediction accuracy. To perform better the hybrid model tries to take the benefits of both the Random Forest (RF) and Long Short-Term Memory (LSTM) networks by using the main features of the models. Probability predictions are used by the LSTM models that the RF models initially generate which reshapes the data to relate and match to the input features specifications, treating each probability vector as a time step. now the dataset is split for testing and training accordingly. To assess overall performance, the hybrid model's combined predictions are analysed. The hybrid model performs better than the other models, achieving 91.00% accuracy, F1 score, and recall. These metrics show that the hybrid model performs better than the RF and LSTM models. The hybrid model has more ability to detect complex and novel threats.

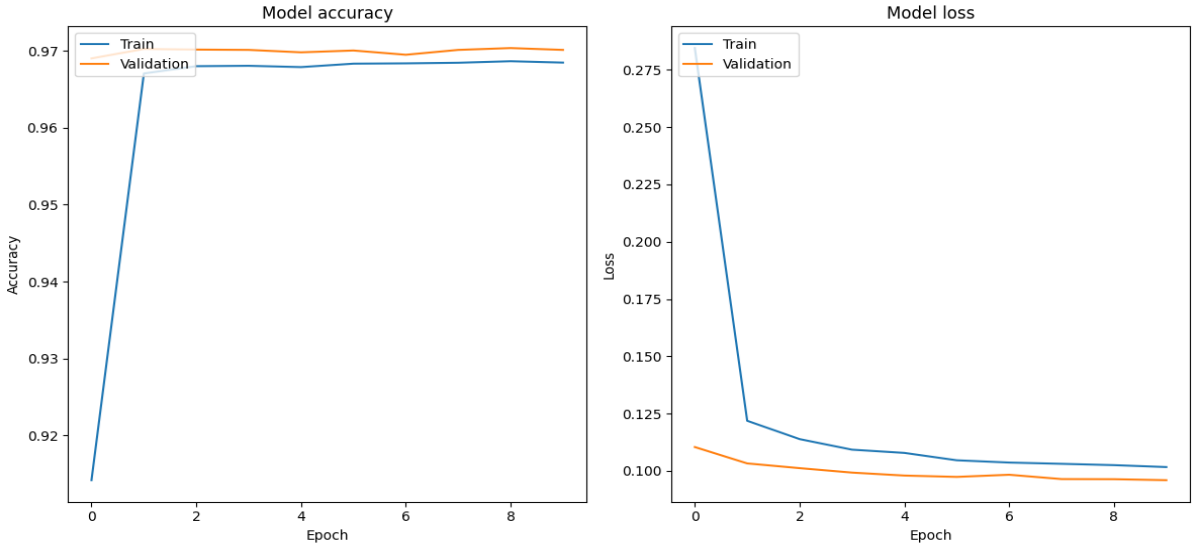

Fig 10. Accuracy and Loss of Hybrid model

The hybrid model performs well right away, as shown in the training graphs. The model's accuracy and loss graphs show that within the first epoch, improvements are quickly made. The training accuracy rises quickly, surpassing 96%, and then stays steady at 97%. Comparably, this pattern is closely followed by the validation accuracy, which stabilizes

slightly below the training accuracy. The model appears to be functioning well on untested data and is not overfitting based on the close alignment of training and validation accuracy.

These findings are further supported by the loss graph. During the first epoch, the training loss shows a sharp decline from above 0.275 to below 0.125, then it steadily declines until stabilizing at 0.075. Similar trends can be seen in the validation loss, which first drops off quickly before stabilizing at 0. 100.The model is learning efficiently and generalizing well, as evidenced by the steady decrease in training and validation loss.

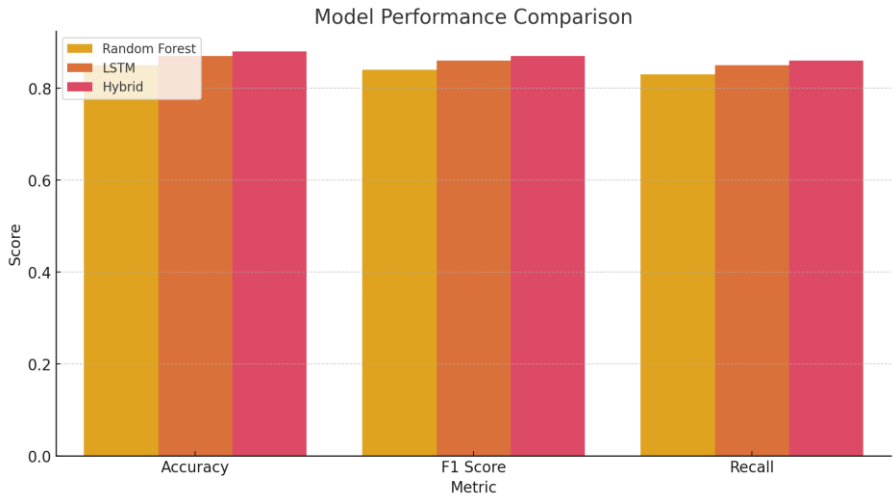

Fig 11. Comparison of Accuracy,F1 Score, Recall of all three models

6.4 Discussion

The aim of this malicious activity is to provide a better security feature in the clouding environment using an advanced NIDS (Network Intrusion Detection System) combined with machine learning algorithms which are RF and LSTM. The combination of RF and LSTM (hybrid model) substantially improves the robustness and accuracy with high performance, at 91.00% detection rate over a variety of attack classes. This hybrid approach, because of the combined strengths in LSTM's temporal pattern recognition (for known attack patterns) and RF's feature extraction capabilities (in addition to concatenated data), allows it to adapt effectively against both kinds of attacks. But the RF model was mostly fair intelligence (AUC 0.72), it had a larger limitation in providing good classification of new kind attacks, The LSTM model had a higher discriminative power (AUC 0.91) but was sensitive to the trade-off between precision and recall, thus requiring fine-tuning in deep learning models. To provide a rigorous evaluation, 20% of the data was split to testing and rest is kept for model training.

The hybrid model outperformed the standalone RF and LSTM models, as shown by its balanced precision-recall diagram and increased accuracy. However if more time was given dynamic threat adaptation could be made which helps to identify new and more unseen attacks.The related works analysis demonstrated that compared to the literature our hybrid

model was very competitive and even exceeded significantly other techniques which generally have difficulties in zero-day exploits and emerging threats detection. Some previous works have shown the limitations of signature-based network intrusion detection systems, and that an ensemble classifier might be better to increase the detection accuracy. Detecting a surge in calls it is built architecture with robust detection methodologies using powerful data preprocessing pipeline, RF feature extraction and LSTM temporal pattern recognition. But, relying on a single dataset such as UNSW-NB15 may limit the generalizability; to overcome this issue future works should consider multiple datasets. The hybrid model which has beaten the traditional methods, and this is very promising to improve NIDS accuracy in cloud environments.

7 Conclusion and Future Work

In this research project, the major advantages of creating a hybrid model by strengths of integrating Random Forest and Long Short-Term Memory for Network Intrusion Detection Systems for the cloud environments the hybrid model has the ability to improve the detection by maintaining the balance of both the accuracy and the managing false positives and negatives are demonstrated. This maintaining of balance is very important in cloud security scenarios because they can create a false detection of threat alarm and missed threats can also affect the system, data, and the reliability of the cloud providers. The crucial key takeaway from this study is discovering the potential of hybrid machine learning to encounter new novel attacks and the evolving challenges security of the cloud infrastructures. By developing the hybrid model which helps to learn and provide a reliable, robust, and feasible solution that can be able to encounter the cloud network traffic which is complex and dynamic. The threats in the cloud network continue to evolve all the time the sophistication of the threats grows continuously this research aids and underscores the need for new adaptive and advanced mechanisms in cloud computing.

The current study has made major progress in enhancing cloud security through a hybrid NIDS. But still there are challenges remain in the real world deployment for adapting new threats and for handling dynamic network environments for new threats online learning techniques could ensure the model evolves with emerging threats with some system scaling features to handle it efficiently for integrating this SHAP and LIME can further enhance its robustness and adaptability and there are several paths that could further enhance these security features by advancing this work immediate ideas that can enhance the project can be done by conducting testing of the hybrid model across multiple datasets to ensure robustness in different cloud environments platforms that could help further to add strength to the model and help to confirm the effectiveness of the model in different settings that too in many diverse network traffic data. Integrating blockchain technologies with Machine learning models is another promising direction to proceed with this research work in the future that can help in developing a decentralized and tamper-proof intrusion detection system because hacking a blockchain network is too tedious and any changes made in the data can be found then and there can be logged there without affecting the entire system this can greatly enhance the NIDS which can give an extra layer to reduce the risk of data tampering. With

this developing an adaptive learning mechanism that updates the NIDS based on current threat intelligence could ensure that the system remains effective against emerging threats. This could include integrating real-time threat feeds and automatically relearning the model to detect new types of intrusions as they appear. By completing this future research, the hybrid NIDS developed in this study can be further developed in the right direction and can be refined and adapted to meet the always-evolving cloud security in the future which can greatly help.

References:

Ahmad, Z., Shahid Khan, A., Shiang, C. and Ahmad, F. (2021) 'Network intrusion detection system: A systematic study of machine learning and deep learning approaches', Transactions on Emerging Telecommunications Technologies, 32. Available at: https://doi.org/10.1002/ett.4150.

Aljamal, I., Tekeoğlu, A., Bekiroglu, K. and Sengupta, S. (2019) '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, pp. 84-89.

Alouffi, B., Hasnain, M., Alharbi, A., Alosaimi, W., Alyami, H. and Ayaz, M. (2021) 'A systematic literature review on cloud computing security: threats and mitigation strategies', IEEE Access, 9, pp. 57792-57807.

Alzahrani, A. O. and Alenazi, M. J. (2021) 'Designing a network intrusion detection system based on machine learning for software-defined networks', Future Internet, 13(5), p. 111.

Aldallal, A. and Alisa, F. (2021) 'Effective Intrusion Detection System to Secure Data in Cloud Using Machine Learning', 13(12), p. 2306. Available at: https://doi.org/10.3390/sym13122306.

Azam, Z., Islam, M. and Huda, M. (2021) 'Comparative Analysis of Intrusion Detection Systems and Machine Learning Based Model Analysis Through Decision Tree', IEEE Access. Available at: https://doi.org/10.1109/ACCESS.2023.3296444.

Balamurugan, V. and Saravanan, R. (2019) 'Enhanced intrusion detection and prevention system on cloud environment using hybrid classification and OTS generation', Cluster Computing, 22(Suppl 6), pp. 13027-13039.

Gharib, M., Abdelbar, A. M. and Hassanien, A. E. (2016) 'An adaptive framework for intelligent intrusion detection systems using genetic algorithms', Journal of Network and Systems Management, 24(3), pp. 463-486. Available at: https://doi.org/10.1007/s10922-016-9374-6.

HSC (n.d.) 'Securing the Internet of Things with Intrusion Detection Systems'. Available at: https://www.hsc.com/resources/blog/securing-the-internet-of-things-with-intrusion-detection-systems/.

Imad, M., Hassan, M., Bangash, S. and Naim, N. (2022) 'A Comparative Analysis of Intrusion Detection in IoT Network Using Machine Learning'. Available at: https://doi.org/10.1007/978-3-031-05752-6_10.

Javadpour, A., Kazemi Abharian, S. and Wang, G. (2017) 'Feature Selection and Intrusion Detection in Cloud Environment Based on Machine Learning Algorithms', in 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), Guangzhou, China, pp. 1417-1421.

Kilincer, I. F., Ertam, F. and Sengur, A. (2021) 'Machine Learning Methods for Cyber Security Intrusion Detection: Datasets and Comparative Study', Computer Networks, 188, p. 107840. Available at: https://doi.org/10.1016/j.comnet.2021.107840.

Meng, W. and Kwok, L.F. (2014) 'Enhancing the Performance of Signature-Based Network Intrusion Detection Systems: An Engineering Approach'. No. 4, pp. 209–222. Available at: http://dx.doi.org/10.1080/1023697X.2014.970750.

Nan, H. and Jilin, U. (2023) 'Apply RF-LSTM to Predicting Future Share Price', SHS Web of Conferences, 170, 02012. Available at: https://doi.org/10.1051/shsconf/202317002012.

Parampottupadam, S. and Moldovann, A.-N. (2018) 'Cloud-based Real-time Network Intrusion Detection Using Deep Learning', in 2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Glasgow, UK, pp. 1-8.

Patel, A., Taghavi, M., Bakhtiyari, K. and Celestino Júnior, J. (2013) 'An intrusion detection and prevention system in cloud computing: A systematic review', Journal of Network and Computer Applications, 36(1), pp. 25-41.

Rajendrasubbu, P.N., 2023. Network Intrusion Detection System for Security Threats in Cloud Computing by Comparative Analysis using Machine Learning Algorithms. National College of Ireland.

Saranya, T., Sridevi, S., Deisy, C. D., Chung, T. D. and Khan, M. A. (2020) 'Performance analysis of machine learning algorithms in intrusion detection system: A review', Procedia Computer Science, 171, pp. 1251-1260. Available at: https://doi.org/10.1016/j.procs.2020.04.133.

Sharma, V., Chen, Y., Park, J. H. and Park, J. H. (2020) 'A Software-Defined Secure Management Framework for IoT Devices', IEEE Access, 8, pp. 11124-11135. Available at: https://doi.org/10.1109/ACCESS.2020.2966988.

Thangaraj, T., Sridevi, S., Chelliah, C. D., Chung, T. and Khan, A. (2020) 'Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review', Procedia Computer Science, 171, pp. 1251-1260. Available at: https://doi.org/10.1016/j.procs.2020.04.133.

Wei, W. and Huang, J. (2020) 'Short-term load forecasting based on LSTM-RF-SVM combined model', Journal of Physics: Conference Series, 1651, 012028. Available at: https://doi.org/10.1088/1742-6596/1651/1/012028.

Yang, Y., Gu, Y. and Yan, Y. (2023) 'Machine Learning-Based Intrusion Detection for Rare-Class Network Attacks', Electronics, 12, p. 3911. Available at: https://doi.org/10.3390/electronics12183911.

GoogleColab: https://colab.research.google.com/drive/1CQbDkiA5ws3R6Jc7JveyrsvUbUS-1Oxr?usp=drive_link & https://colab.research.google.com/drive/1k3J7bP4m1uwnRxy4uCp5ls_72aT5ARt7?usp=drive_link