Deep Learning-Based Intrusion Detection with Resource-Aware Auto-Scaling for Secure Cloud Computing

THESIS

Submitted towards the partial fulfilment of

the requirements of the award of the degree of

Master of Technology
In
COMPUTER SCIENCE AND ENGINEERING

Submitted by

NANDIT BHARDWAJ

2K24/CSE/02

Under the guidance of

Dr. R.K. Yadav

Associate Professor

Department of Computer Science and Engineering

DELHI TECHNOLOGICAL UNIVERSITY

(FORMERLY Delhi College of Engineering)

Bawana Road, New Delhi-110042

May 2026

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, New Delhi-110042

CANDIDATE'S DECLARATION

I, **Nandit Bhardwaj**, Roll No. **2K24/CSE/02**, student of M.Tech (Computer Science and Engineering), hereby declare that the thesis on "**Deep Learning-Based Intrusion Detection with Resource-Aware Auto-Scaling for Secure Cloud Computing**", presented before the Department of Computer Science and Engineering, Delhi Technological University, New Delhi, as partial fulfillment for the award of Master of Technology degree, is the outcome of my own research efforts and that no part of this thesis has been copied without due acknowledgment. This piece of work has not been used either in whole or in part for obtaining any other Degree, Diploma, Associateship, Fellowship, or other similar distinction.

Place: Delhi **Nandit Bhardwaj**

Date: 2K24/CSE/02

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Signature of Supervisor **Signature of External Examiner**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, New Delhi-110042

CERTIFICATE

I hereby declare that thesis titled "**Deep Learning-Based Intrusion Detection with Resource-Aware Auto-Scaling for Secure Cloud Computing**" which is submitted by Nandit Bhardwaj - 2K24/CSE/02, Department of Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfillment of the degree of Master of Technology is a record of the thesis work carried out by the student under my supervision. To the best of my knowledge, this work has not been submitted in any part or fulfillment for any Degree or Diploma to this University or elsewhere.

Place : Delhi

Date :

Dr. R.K. Yadav

SUPERVISOR

(Associate Professor, CSE, DTU)

ACKNOWLEDGEMENT

I am grateful to Dr. R.K. Yadav, Associate Professor, Department of Computer Science and Engineering, and all the faculty members at DTU for their guidance and support throughout the seminar. I would like to extend my heartfelt gratitude to Prof. Anil Singh Parihar, Head of the Department, Computer Science and Engineering, Delhi Technological University, for his continuous guidance, motivation, unwavering encouragement and invaluable insights.

I would also like to thank the university for the laboratories and other resources it made available to me, which enabled me to proceed seamlessly through the process.

I am also grateful to everyone who directly or indirectly helped to the completion of this thesis, and my parents for their constant support, belief in me, and encouragement.

Nandit Bhardwaj
2K24/CSE/02

ABSTRACT

Cloud computing infrastructures have become a part of the modern digital ecosystem that supports, For enterprise usage such as high availability, elasticity, and operational efficiency. With this increasing dependency, cloud environments have simultaneously become major targets for advanced cyber-attacks like Denial of Service (DDoS), brute-force and more. Attacks (attempts) and botnet attacks. Traditional intrusion detection systems (IDS), are usually limited to static signatures or rule-based logic, and are often not able to detect attack patterns were emerging or dealing with the high dimensional nature of real network traffic. Moreover, current security solutions don't typically measure the impact of malicious traffic on cloud resources. CPU utilization, especially in regard to CPU load, is critical for the quality of service. Providing automatic scaling mechanisms.

This work, a unified deep learning based framework is proposed to simultaneously ad-Design cyber threat detection and resource consumption forecasting in cloud systems. Using the CICIDS2017 dataset, Two types of networks were implemented, namely Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) for binary traffic classification. The LSTM model was able to obtain an accuracy of about 97% which is better than the CNN. An important reason why model is found at 96% is that it is capable of modeling the temporal relationships between sequential network flows. We modeled the operational effects of attacks with a Random Forest regression model. has been trained to predict CPU utilization by malicious traffic intensity with R^2 score of 0.90. This high predictive capability clearly shows that there is a relationship between attack patterns. With the development of cloud security analytics, it will be more feasible to integrate security analytics resources, and security analytics will be more widely used in the future.

In combination with predictive resource management. A simulated auto-scaling was then developed that used the predicted CPU utilization. A mechanism to mimic the cloud-platform behaviour when it is subjected to load fluctuations during an attack. Scale out events were started successfully if CPU thresholds were exceeded, this is an illustration of the potential benefit of predictive, security-aware auto-scaling in minimizing performance impact degradation during cyberattacks.

TABLE OF CONTENTS

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ABBREVIATION	FULL FORM
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
IDS	Intrusion Detection System
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
CPU	Central Processing Unit
VM	Virtual Machine
DDoS	Distributed Denial-of-Service
DoS	Denial-of-Service
API	Application Programming Interface
RF	Random Forest
GPU	Graphics Processing Unit
KPI	Key Performance Indicator
MSE	Mean Squared Error
MAE	Mean Absolute Error
R^2	Coefficient of Determination
CSV	Comma Separated Values
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
CICIDS	Canadian Institute for Cybersecurity Intrusion Detection System

Chapter 1

INTRODUCTION

Cloud Computing is a technology which is required in today's digital world. It provides scalable, and cost-effective computing resources for all application across industries including healthcare, finance, e-commerce, and government. All the Organizations are moving with more important workloads to cloud platforms and which resulted in an improvement in the amount of network traffic moving on complex networks. But this growth has also brought new and advanced levels of cyber threats that take advantage of the openness and multi-tenant aspects of cloud infrastructures. Denial-of-Service (DDoS) attacks, intrusions via brute force, botnets, and reconnaissance scanning are all common methods used for malicious activity, with a threat level that affects not only service availability but also overall system performance.

Traditional IDSs are based on signature or on simple rules. While they can detect known attacks and threats, these systems are unable to detect new attack patterns, obfuscated payloads and new strategies from the attackers. They don't have the capability to evaluate the effect of malicious traffic on cloud resources, particularly CPU utilization. When an auto-scaling mechanism attempts to scale up as a result of a rise in workload demand, it can mistakenly consider attack traffic as legitimate usage, which increases resources consumption, reduces performance, or raises operational costs. The restrictions outlined here highlight the importance of more intelligent and adaptive cybersecurity solutions that can function effectively in dynamic cloud environments.

Deep learning has proven to be a promising technique to deal with high dimensional network traffic and detect complex attack behaviors. Feature learning from raw data can be done automatically by models like Convolutional Neural Networks(CNN) and Long Short-Term Memory networks(LSTM), without requiring a lot of feature engineering. The temporal dependency characteristics of attack sequences are very well captured by LSTM networks, and the structural correlations between the different feature dimensions are captured by CNNs. With all these advances, current deep learning-based intrusion detection systems

are only capable of traffic classification, and do not consider the overall effect of attacks on the cloud infrastructure.

It is crucial to be aware of how cyberattacks can affect cloud resource use in order to keep services reliable. When a lot of bad traffic is generated, it can consume the CPU resources, cause multiple scale-out events or interfere with the auto-scaling process meant for legitimate workload growth. If a cloud system has no way to distinguish between a true surge in usage and an attack, then it will be vulnerable to economic exploitation and performance instability. This difference is what makes it necessary to have a combination of intrusion detection and prediction of the usage of resources in order to make more adaptable and cost-efficient cloud systems.

This research provides an integrated architecture for the deep learning attack detection, CPU utilization prediction and auto-scaling simulation. There are four steps in the Architecture. A preprocessing module is used to select the features from the CICIDS2017 dataset, normalize values and balance the dataset. An intrusion detection module is based on LSTM and CNN models for traffic classification as benign or malicious. The detected attack patterns are used for the prediction of the CPU utilization using the Random Forest regression model. Finally, an autoscaling simulator simulates how the system would scale when facing different attack loads, based on the threshold-based rules. These elements, when combined, create to a single framework to examine the security and performance aspects of cyber threats in cloud environments.

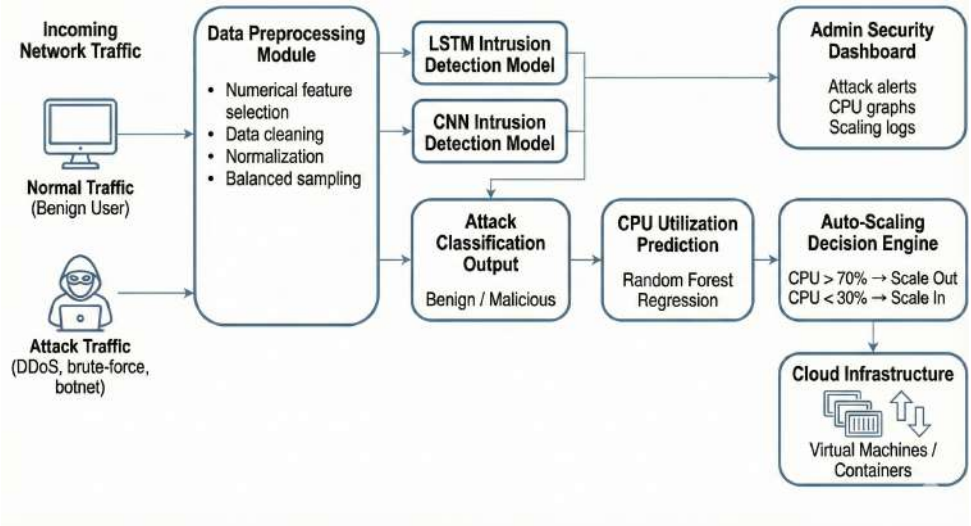

Figure 1.1: Proposed Cloud Security and Auto-Scaling Architecture.

The aim of this research is to show the capability that deep learning models can add to intrusion detection and offer insight regarding the impact of attacks on resources. The system correlates the intensity of attacks to the CPU load trends to give predictive insight which can be used in making better autoscaling decisions. This comprehensive strategy helps reduce the need for over-provisioning resources, increases security resilience on attack cycles and facilitates the development of intelligent cloud systems that can evolve to evolving security demands.

The value of this work stems from the fact that the concept of cybersecurity and resource management has been combined into one approach. The proposed framework does not have a distinction between detection and performance analysis, but combines them as a single analytical pipeline. Lightweight, easy to understand and easily replicable for academic research and for use in real-world cloud orchestration systems, such as (AWS) Auto Scaling Groups, (Azure) VM Scale Sets, or (Kubernetes) Horizontal Pod Autoscalers. This study provides a basis for developing more adaptive, security-aware cloud architectures by linking the two important concepts of detection accuracy and resource behavior prediction.

The concept of the framework is good, but the study is only conducted on offline data and CPU simulation. It doesn't support live data ingestion, multi-class attack classification, or connection with live autoscaling engines. These barriers are challenged and offer avenues for further work, however. The proposed approach could be extended to promising directions such as real-world deployment, streaming IDS pipelines, reinforcement learning-based scaling policies and multi-cloud evaluation.

The growing use of cloud services worldwide makes real time detection, prediction and removal of cyber threats of critical importance to the reliability and safety of cloud-hosted services. This paper aims to make a contribution to building cloud infrastructures that are resilient to evolving cyber threats and perform optimally and cost effectively by evaluating deep learning models for intrusion detection and by incorporating the deep learning models with CPU prediction and autoscaling simulation. The results revealed here will serve as the basis for the design of intelligent and autonomous mechanisms for defense in cloud environments, which can adapt proactively to hostile environments.

1.1 Challenges in Securing Dynamic Cloud Environments

Cloud infrastructures are extremely dynamic and distributed. VMs, containers, serverless functions, microservices are constantly communicating among data centers located in different regions.VMs, containers, serverless functions, microservices are constantly communicating across geographically distributed data centers. This flexibility helps to scale more easier and utilize resources more effectively, but also gives rise with serious security concerns. The number of devices, APIs and services communicating over the network increases attack surface.

The biggest issue in the cloud is the proliferation of advanced cyberattacks, including Distributed Denial of Service (DDoS), brute-force attacks, infiltration attacks, botnets and data exfiltration attacks. These attacks are hard to detect with traditional rule-based IDS systems since they heavily depend on predefined signatures and static patterns. Static systems fail to generalize when they encounter new attacks because the attack is dynamic.

Important problems in cloud computing is the instability of resources in the face of cyber attacks. Malicious traffic uses a lot of CPU, memory, bandwidth and storage space. This abnormal behavior affects application performance and could even result in service downtime. Current cloud auto-scaling technologies address demand and traffic loads of users but rarely take into account impact of cyberattacks on resource consumption.

Considering that intelligent intrusion detection combined with predictive resource management becomes an absolute necessity for the next generation of cloud systems, it is most essential to include them.

Chapter 2

RELATED WORK

Security is an important domain to study in the context of cloud infrastructures, as these are a favored target of sophisticated cyberattacks and due to that are crucial to monitor for intrusion and resource management. Researchers have been investigating different approaches to ensure that Intrusion Detection Systems (IDS) are more accurate, flexible and robust, while other studies have been conducted to boost cloud performance by using predictive autoscaling models. While these advances have been made, there has been little research that has considered the interaction between the two domains, leaving this area still unexamined. The existing methods in the traditional IDS, machine learning based detection, deep learning architectures, and cloud resource prediction frameworks are summarized in the following sections.

2.1 Traditional IDS and Machine Learning-Based Approaches

A traditional classification of Intrusion Detection Systems (IDS) is into two main groups: signature based and anomaly based approaches. Signature based IDS (Snort, Suricata) are based on specific signatures or patterns of attacks. They can detect patterns and attacks but not unknown ones, known malware that changes signatures, or attacks that occur in an unexpected way from a known pattern. They are also static in nature and are not very flexible when dealing with dynamic cloud environments with varying traffic patterns.

A more flexible approach was provided by Machine Learning (ML)-based IDS, which allows systems to learn statistical characteristic of abnormal traffic. The classical supervised ML algorithms that learnt discriminative features from data, like Support Vector Machines, Decision Trees, Naive Bayes, Logistic Regression, Random Forests and k-Nearest Neighbours showed better performance in detecting anomalies. But these models require a lot of feature engineering and are sensitive to noise, data imbalance and traffic distribution changes.

Furthermore, ML classifiers are often ineffective in dealing with high dimensional network traffic and temporal dependency of attack sequences. Researchers have found that ML models are aging and when an attacker changes its tactics, the accuracy drops and false positives increase. Although they were limited, traditional ML-based IDS was a stepping stone to more advanced and automated deep learning IDS models.

2.2 Deep Learning for Network ID (Intrusion Detection)

With the advantages of automatically extracting hierarchical features, deep learning (DL) has been applied as a powerful tool in network traffic analysis and has become a compelling answer to complex and high-volume network traffic. In intrusion detection, Convolutional Neural Networks (CNNs) have been extensively applied to convert network flows into structured matrices, which can represent the spatial patterns. IDS solutions based on CNN have been successful in the detection of attacks like DDoS attack, brute-force attack and port scanning. They are well suited for cloud environments where traffic varies across different input distributions, as they can be generalized to different traffic variations.

There are other variations of RNN, such as LSTM(Long Short-Term Memory) networks, which offer further benefits by considering temporal dependencies in network sequences. IDS models using LSTM can be trained to recognize the pattern of successive packets or flows, allowing the detection of staged attacks, repeated logon attempts, scanning attacks and botnet activities. LSTMs are always superior to CNNs for detecting time-dependent attacks because of its ability to hold context for long.LSTMs consistently surpass the CNNs in detecting attacks based on time because of its ability to hold the context for long.

Some hybrid architectures have also been suggested to capture both spatial and temporal dependency, such as CNN and LSTM. They typically perform better in terms of accuracy and robustness than single models. Most of the DL based IDS research has, however, been limited to solely the classification task. Few studies try to study the impact of the detected attack pattern on the utilization of resources, such as CPU load and autoscaling events for cloud systems.

2.3 Cloud Resource Prediction and Auto-Scaling Research

Auto-scaling is an important feature of cloud platforms to ensure the performance of applications in the presence of workload fluctuations. Traditional Auto-scaling: relies on triggers based on thresholds, such as CPU usage, memory usage, or request rate, to trigger scaling up (scale out) and down (scale in) operations. Simple and common, threshold-based systems are only reactive and cannot predict future load conditions. This frequently leads to under-scaling or over-provisioning of resources and/or performance problems.

Researchers have investigated the scalability, which has led to the development of predictive scaling models based on machine learning, time-series forecasting, or statistical techniques. The CPU usage, the network load and the request throughput have been predicted by means of ARIMA, Holt-Winters, regression models and LSTM-based forecasting. Additional tree-based models such as RF(Random Forest) have been found to perform well when modeling the relationship between the workload patterns and the resources consumed, but nonlinear.

But most of these predictive autoscaling methods do not take into account cybersecurity findings. Most assume that workload variations are benign, and that they are not able to differentiate between traffic increases due to actual traffic growth and traffic increases because of attack. Traditional autoscaling mechanisms are easily manipulated in cloud environments, where cyberattacks can quickly cause high CPU usage, scale-out events, and high costs of operation. A few research efforts have tried to make this connection between the output of IDS and auto-scaling logic, but no one has done so in a unified framework where accuracy of detection and resource usage predictors are both considered.

2.4 Kubernetes Security and AutoScaling

As an orchestration platform for cloud-native applications, Kubernetes has become one of the more popular options for deploying container-based applications. It handles the deployment of containers, discovery of services, allocation of resources, fault recovery and scaling in distributed computing environments. Kubernetes is dynamic and distributed, and therefore presents specific security challenges that need to be monitored and constantly addressed with intelligent threat detection mechanisms.

Modern cloud applications typically have many microservices deployed on the cluster, each running in a container. This design has a number of benefits, such as enabling better scaling and resource utilization, but it also increases the attack surface. The cluster can be compromised by exploiters when containers are mis-configured, APIs are vulnerable, privilege escalation is possible or compromised images are used. Traditional intrusion detection mechanisms are hardly effective in these environments as the workloads of containers are dynamically short-lived.

Some studies have come up with frameworks for securing the Kubernetes environment. These methods track pod behavior, system calls, communications in the network, and patterns of using resources to detect malicious activity. Given that the complex attack patterns are hard to detect with static rules, the usage of machine learning and deep learning has been growing to analyze Kubernetes telemetry data to identify these patterns.

Kubernetes also has the capability of built in auto scaling. Pod Autoscaler(HPA) automatically scales the number of running pods based on the metrics like CPU utilisation, memory usage, or custom performance metrics. This mechanism enables the application to perform even in cases of load change.

A Denial of Service(DDoS) attack may lead to high CPU consumption which may result in unnecessary scale-out operations. This can lead to an increase in infrastructure expenses and to less stability in the services.

Recent research indicates that adding intrusion detection results to the Kubernetes scaling process can enhance cloud resiliency. Cloud platforms can more effectively distinguish between workload growth and the consumption of resources due to an attack by using security analytics and autoscaling policies. The motivation for this concept is to incorporate intrusion detection and resource prediction in the proposed framework.

2.5 AWS and Cloud Protection

Amazon Web Services(AWS) is one of the biggest cloud computing platforms that are utilized by businesses around the globe. AWS offers a wide range of infrastructure, platform and software services to enable applications to scale. Security is one of the most important concerns in AWS environments, especially for those that are widely-used.

AWS has several security products that help you to secure cloud resources from cyber threats. Continuously monitoring for suspicious activity of cloud workloads and network traffic are services like Amazon GuardDuty, AWS Shield, AWS Web Application Firewall (WAF), or Amazon Inspector. These services employ threat intelligence feeds, anomaly detection and behavioral analytics to detect malicious behavior.

Amazon GuardDuty is a managed threat detection service that uses log data, DNS requests, API activity and network-flow data to detect potential attacks. AWS Shield protects against Distributed Denial of Service attacks, and AWS WAF allows organizations to create their own filtering rules for traffic that is sent to WAF. All of these services are part of a multi-layered cloud defense architecture.

Automated resource management is also possible with AWS, for example with Application Auto Scaling and EC2 Auto Scaling Groups. These services scale infrastructure capacity in response to performance thresholds that are programmed in. Auto Scaling automatically adjusts the number of computing resources allocated to ensure that your application is available and running at an optimum cost for each use.

Most of the cloud security and scaling services run independently of these capabilities. Typically, security alerts raised by GuardDuty or AWS Shield will not impact scaling decisions directly. This can cause unwanted resources to be expanded even in the event of attacks by malicious traffic.

Researchers have therefore looked into how to combine security analytics with cloud resource management (CRM). Detected patterns can be used to predict infrastructure impact and can be helpful for intelligent scaling decisions. This research direction is reflected in the proposed framework that integrates intrusion detection, CPU utilization prediction and autoscaling simulation into a single framework.

2.6 Research Gap and Motivation

The current study shows that the literature has already made significant strides in deep learning-based intrusion detection and cloud resource management. These two domains are to a large extent developed separately, however. Research in intrusion detection has been focused on enhancing classification performance, whereas research in cloud autoscaling has

been centered on forecasting the needs of the resources and optimizing their performance. There has been limited research exploring the direct correlation between the threat detected and cloud resource usage, and a gap in knowledge of the impact of cyberattacks on system performance.

In DL based IDS and CPU utilization prediction/autoscaling simulation, no unified framework is currently available that connects both areas. This disconnection makes it harder for cloud systems to intelligently react to the pressure caused by an attack. Security considerations are also a concern for cloud platforms, with unawareness of security-aware Autoscaling potentially leaving cloud platforms vulnerable to economic DDos attacks and service disruption.

2.7 Limitations of Existing Security-Aware Cloud Systems

Despite the large amount of research undertaken in cloud intrusion detection and cloud auto-scaling, there are still some unresolved limitations. Current IDS solutions only detect malicious traffic, and do not take into account the consequences of cyber attacks on cloud infrastructure. These systems alert on security but cannot help adapt the use of resources.

With respect to the benchmark data sets, the traditional ML models like Decision Trees, (SVM)Support Vector Machines and (RF)Random Forest classifiers have performed well. The high dimensionality of sequential network traffic data, though, and the lack of long-term temporal dependency are challenges that these models may have trouble with. In addition, much of the current approaches are highly manual and feature-engineered.

Although deep learning intrusion detection models have enhanced feature extraction, they are often high in computation costs and hard to implement in the real world. In some studies, evaluation of the model is limited to classification accuracy only and other crucial performance metrics like scalability, latency, false positive rates, resource overheads etc. are neglected.

Another issue that is uncovered within the current literature of Cloud resource management is the lack of security awareness in auto-scaling systems. Cloud auto-scalers are typically looking at the utilization of workloads, for example CPU, response time and memory usage, but do not differentiate between legitimate workload peaks and malicious traffic

surges. This can lead to unnecessary scaling operations by attackers, which can add infrastructure expenses and decrease availability.

Very few studies try to combine attack detection and prediction of resources. In most of these studies, the study is mostly theoretical analysis and not practical simulation. A single framework that can detect attacks, predict impact on resources and trigger intelligent scaling decisions at the same time still has a great need.

Chapter 3

METHODOLOGY

The proposed system integrates DL based intrusion detection with CPU utilization prediction and a rule-based autoscaling mechanism. This chapter outlines how the models were created, trained and tested, and how the end-to-end workflow was designed. The methodology includes four major parts namely, dataset preprocessing, deep learning model construction, resource utilization prediction and the auto-scaling simulation framework. Each component is described in the next few sections.

3.1 Data Preparation and Preprocessing Pipeline

The CICIDS2017 dataset was chosen for being a realistic mix of benign traffic and attack traffic, by having number of dimensional features of the network flows, and with the well structured labels. The preprocessing pipeline performs several tasks, including cleaning, normalizing, and transforming the data into a suitable format for DL models.

3.1.1 Dataset Cleaning and Null Handling

The raw data has missing values, duplicate data and network fields that are not defined. All non-numeric and corrupt values were deleted. For continuous attributes, missing values were imputed with the mean value.

3.1.2 Feature Selection and Numerical Encoding

Only numerical features were retained, as deep learning models perform better with continuous tensors. All categorical columns were excluded. The label column converted into the binary format:

- $0 \rightarrow$ Benign
- $1 \rightarrow$ Malicious

3.1.3 Data Normalization

In order to prevent training bias due to the magnitude differences of features, the min max normalization technique was used to normalize all the numerical features in the [0,1] range.

3.1.4 Dataset Balancing and Sampling

The data is highly imbalanced as the benign case samples outnumber the malicious case samples, a balance subset of 50,000 samples (25,000 benign and 25,000 malicious) was extracted. This will make the training of models stable and avoid the overfitting of the majority classes.

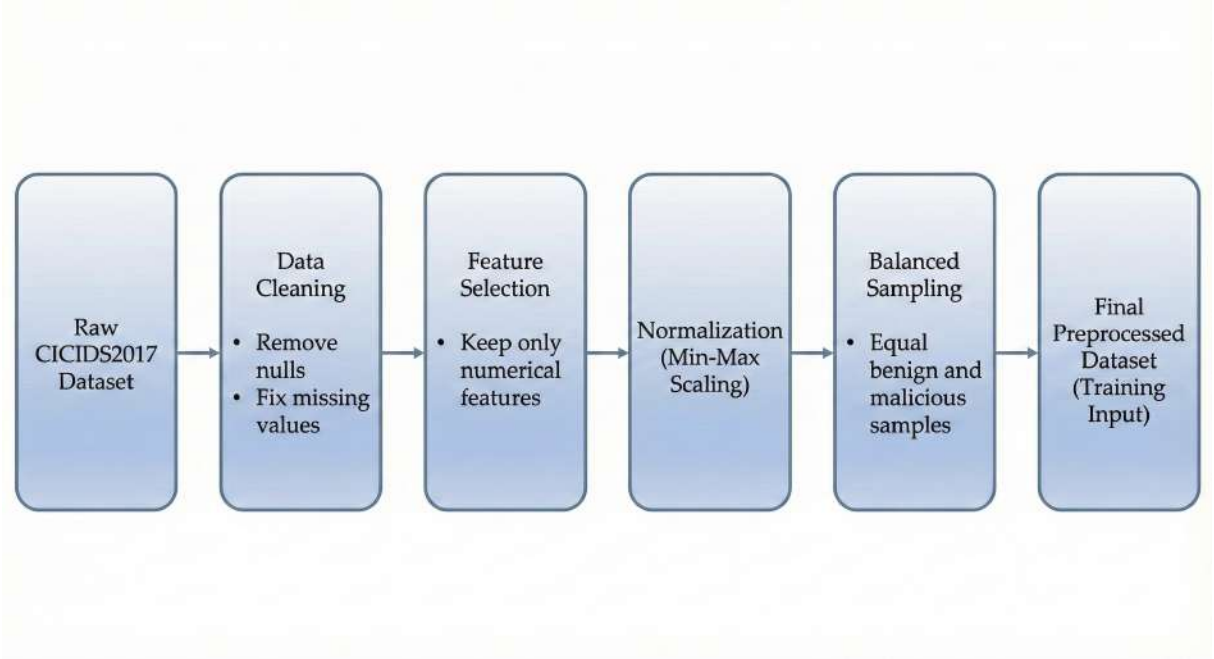

Figure 3.1: Overview of the preprocessing steps.

3.2 Architecture of Proposed Models

Two deep learning architectures were chosen: LSTM based IDS, CNN based IDS. These models complement to each other to capture temporal and spatial feature of the traffic of the cloud network.

3.2.1 LSTM Intrusion Model

Goal of LSTM networks is learn sequential feature in traffic behavior. Temporal dependencies are useful for many cyberattacks, like brute force attacks or the sequence of scans, and LSTMs can learn these patterns well.

3.2.1.1 Model Structure

- Input layer (78 normalized features)
- LSTM layer with 128 units
- Dropout layer (0.2)
- Dense layer with ReLU activation
- Output layer using Softmax for binary classification

3.2.1.2 Training Details

- Optimizer: Adam
- Loss function: Categorical Cross-Entropy
- Epochs: 10
- Batch Size: 64

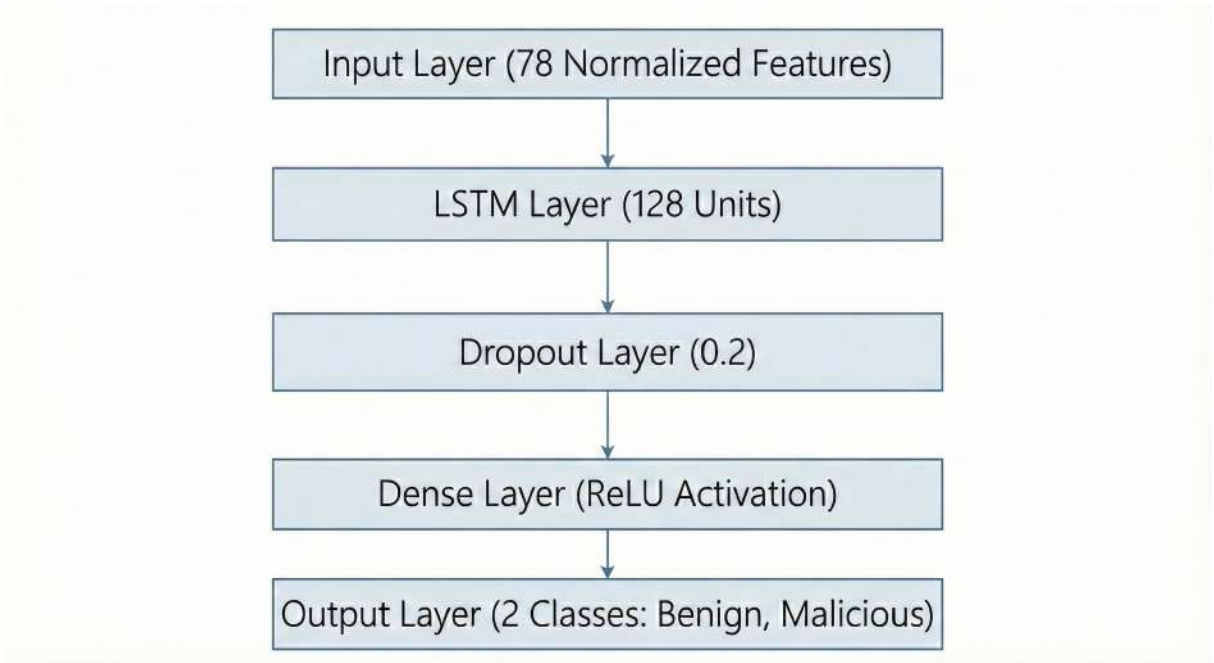

Figure 3.2: Long Short-Term Memory (LSTM) model architecture.

3.2.2 CNN Intrusion Model

CNNs treat network traffic characteristics as spatial patterns, enabling extraction of local feature correlations. This helps detect structured attack patterns such as DDoS bursts and repetitive probing cycles.

- **Reshaping of Numerical Attributes:** Numerical and encoded features are structured into 1-D or 2-D grids suitable for convolution.
- **Convolutional Feature Extraction:** Convolutional filters detect spikes, waveforms, and structural attack signatures.
- **Pooling and Batch Normalization:** Pooling layers reduce feature noise and dimensionality, while normalization stabilizes activation statistics.
- **Flattening:** Multi-dimensional feature maps are flattened into a dense vector.
- **SVM Classification Layer:** Instead of a softmax classifier, a margin-based SVM layer is used to minimize misclassification error.

3.2.2.1 Model Structure

- Input reshaped into a 1×78 tensor
- 1D Convolution layer (filters = 64, kernel size = 3)
- MaxPooling layer
- Flatten layer
- Dense layer with 64 neurons
- Output layer using Softmax

3.2.2.2 Training Details

- Optimizer: Adam
- Loss function: Categorical Cross-Entropy
- Epochs: 10
- Batch Size: 64

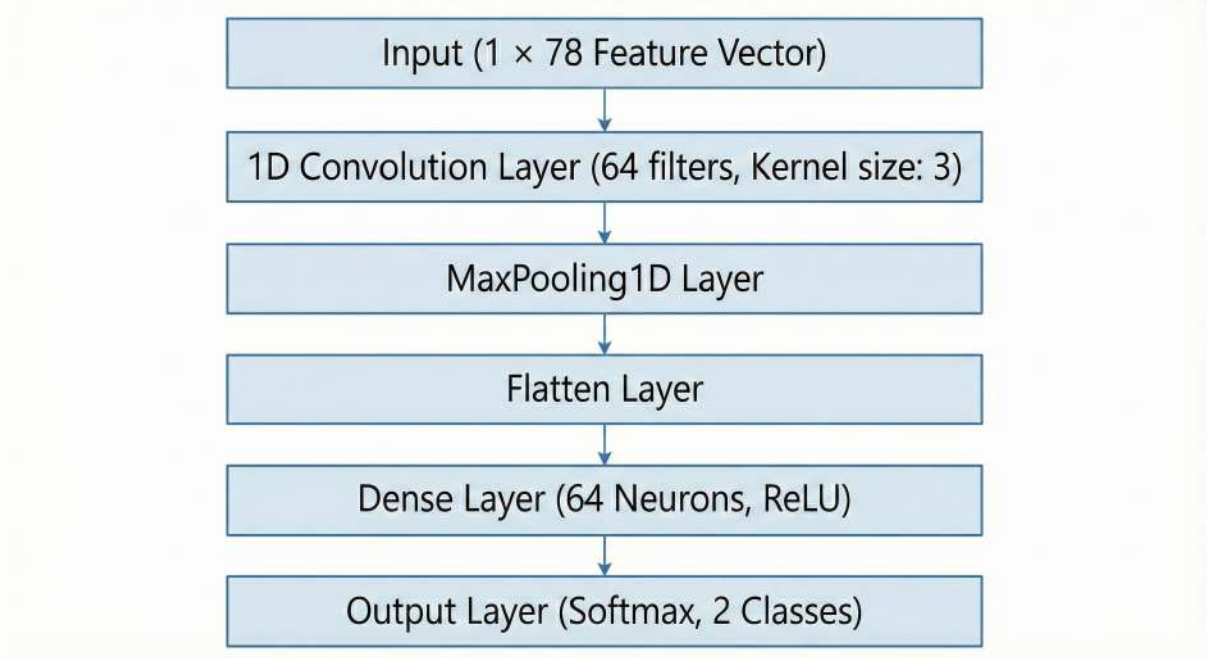

Figure 3.3: Convolutional Neural Network (CNN) architecture.

3.3 Workflow of the Framework

The framework is based on the multi stage procedure of data preprocessing, intrusion detection, CPU utilization prediction, and auto-scaling simulation. All the components help to create a secure and efficient cloud environment.

The CICIDS2017 data set is first gathered and preprocessed in the first stage. Raw network traffic is incomplete, redundant, has categorical attributes and numerical ranges that are inconsistent. So, invalid records are deleted and the quality of the data set is enhanced through data cleaning techniques.

In the second stage, numerical encoding and normalization are performed. All features are normalized to a uniform range with min max normalization as deep learning models are sensitive to the scales of the characteristics. Additionally, dataset balancing methods are provided to remove the class imbalance and promote balanced model training.

In the third stage, the intrusion detection is made using LSTM and CNN models. The LSTM model is able to learn the temporal attack behaviour and can be used to detect sequential dependencies in traffic data. CNN is able to acquire spatial patterns from transformed feature representations and efficiently detect malicious traffic signatures.

The fourth stage involves predicting CPU utilization using a Random Forest regression model, considering the intensity of the attack and the traffic-related features. This component is responsible for providing an estimation of the operational effect of cyberattacks on cloud resources.

Finally, the estimated CPU is used in an auto-scaling simulation tool based on a threshold. If CPU usage is above the desired limits, more instances are turned on to ensure stability. On the other hand, resources are minimized when not in use to help manage the operation costs.

The overall flow of the process allows intrusion detection and resource management to work seamlessly as a single intelligent cloud defence system.

3.4 CPU Utilization Prediction Model

Therefore, a Random Forest Regression created to predict CPU utilization based on the intensity of attacks to understand the impact of malicious activity on the consumption of

cloud resources.To do that, the (RF)Random Forest Regression Model was developed to predict the CPU utilization based on intensity of attacks.

3.4.1 Input and Output Mapping

- Input: Number of malicious predictions per time window
- Output: CPU utilization percentage (0–100%)

3.4.2 Model Configuration

- 200 decision trees
- Mean Squared Error(MSE) as split criterion
- By grid search depth is found

3.4.3 Rationale

The Random Forest model is capable of modelling CPU load under adversarial traffic, as it is able to account for non-linear relationships well.

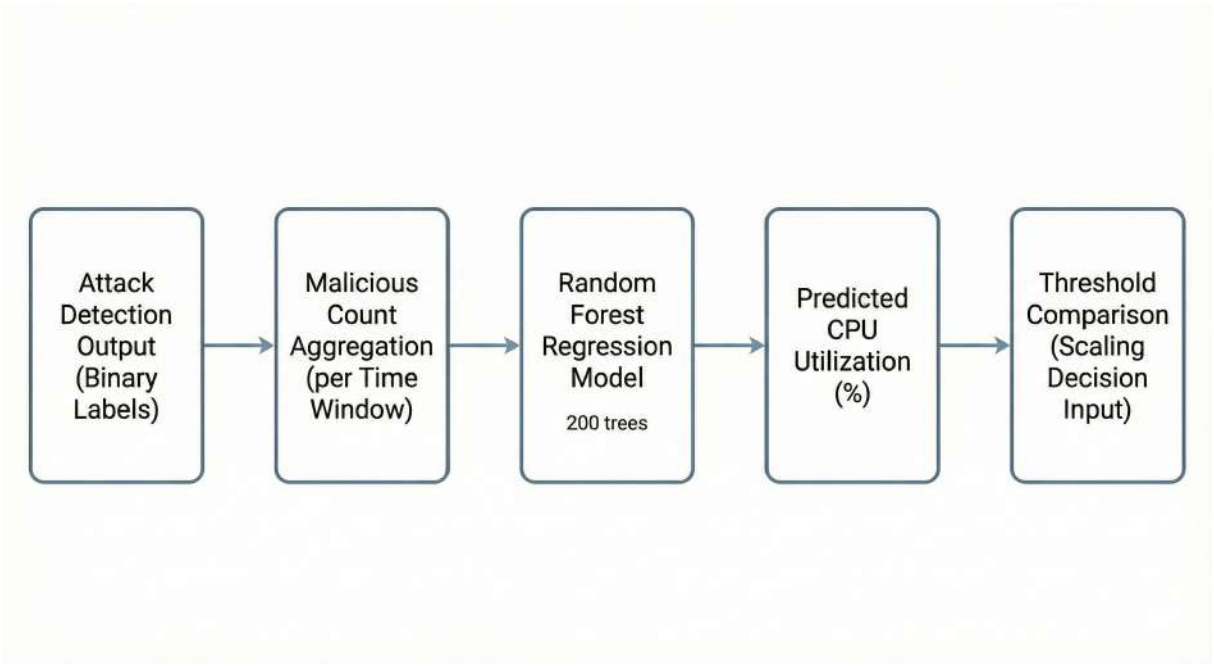

Figure 3.4: Auto-Scaling Decision Engine and Instance Management.

3.5 Auto-Scaling Simulation Framework

An autoscaling simulation was designed to show the reactions of the cloud systems to the predicted CPU load. This will give the student a realistic setting in which to observe the impact of attacks on resource scaling.

3.5.1 Autoscaling Rules

- If predicted CPU $> 70\% \rightarrow$ Scale Out (add an instance)
- If predicted CPU $< 30\% \rightarrow$ Scale In (remove an instance)
- Otherwise $\rightarrow$ No Action

3.5.2 Instance Management

It is assumed that each instance would be running a fixed amount of CPU capacity. This is because the number of running instances is updated based on scaling decisions.

3.5.3 Visualization Outputs

The simulator produces:

- CPU usage trend
- Per time step the malicious traffic count
- Number of active cloud instances
- Autoscaling actions (OUT / IN / NONE)

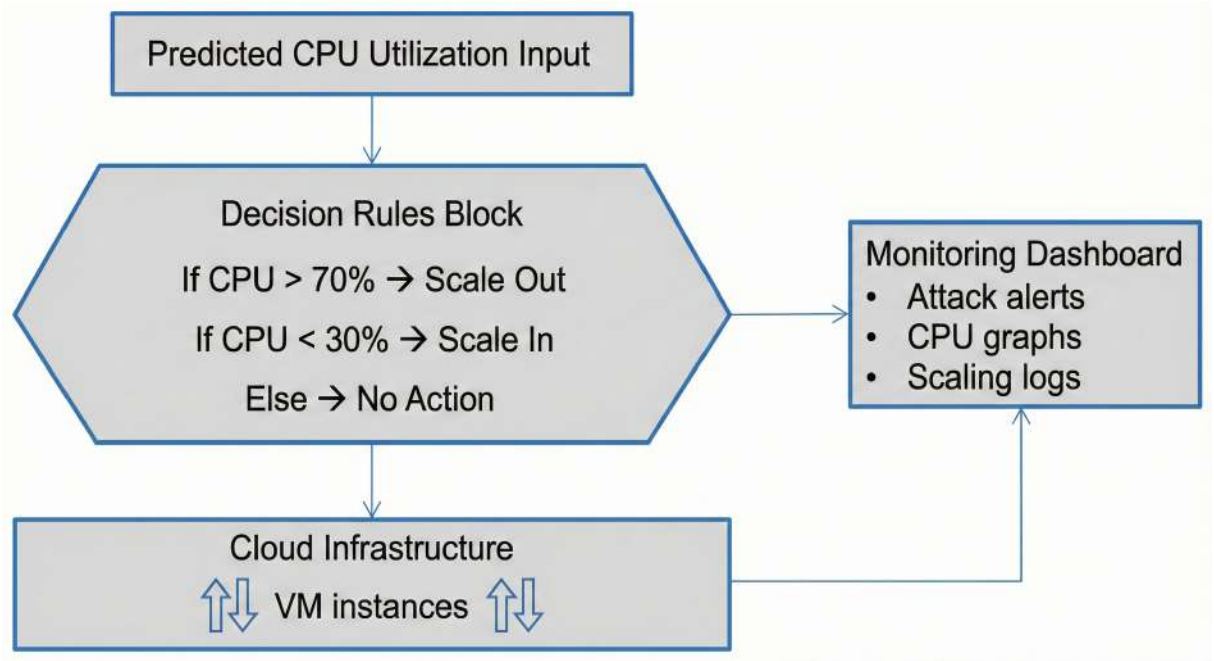

Figure 3.5: Workflow of the CPU utilization prediction model using Random Forest regression.

3.6 Integration with AWS Cloud

While the proposed framework is tested with offline data and simulation-based experiments, it has been designed with practical cloud deployment scenarios. One possible implementation could be done with the AWS infrastructure.

Network traffic with cloud workloads on Amazon EC2 instances, Elastic Kubernetes Service(EKS), or on containerized microservices can be collected. Real-time traffic and resource usage data can be obtained through monitoring services like Amazon CloudWatch, VPC Flow Logs, and Guard Duty.

The intrusion detection models created in this research can be implemented as an inference service on a continuous analysis of incoming traffic streams. The predictions from the LSTM and Convolution(CNN) models can be passed to a resource prediction part that will predict the CPU utilization in future.

Scaling recommendations can be generated and included with AWS Auto Scaling Groups based on the predicted resource consumption. This integration help make decisions about scaling, taking into account the cybersecurity environment along with the workload. As a consequence, cloud resources may be managed efficiently while having protection against

variations in workloads caused by attacks.

The architecture can be used to support future production deployments in cloud systems where security monitoring and resource management must be deployed as a single system.

3.7 Kubernetes(K8s) Deployment Strategy

The framework also works in cloud based environment with Kubernetes. This deployment model consists of the parts of the framework as separate microservices.

The preprocessing module ingests information from all the nodes of the cluster about network telemetry and traffic. The trained deep learning models are used by the intrusion detection module to make inferences, and classify traffic as benign or malicious. The detection component derives attack intensity metrics and the CPU prediction services predict future resource consumption based on these.

Services can be communicated via REST APIs or via event streaming systems like Apache Kafka. The modular design has benefits for scalability, fault tolerance, and maintainability.

The predicted CPU utilization as generated by the framework can be fed as a custom metric to the Kubernetes Horizontal Pod Autoscaler. The autoscaler can scale resources not just based on the usage it sees now, but also proactively scale resources before they become degraded.

This integration provides a cybersecurity-aware autoscaling by taking into account the demand of the workload and cybersecurity indicators when deciding on infrastructure. This way, there are fewer unnecessary scaling operations and optimized operations during attack scenarios.

3.8 Mathematical Formulation of Models

The proposed system utilizes both classification and regression models for attack detection and resource prediction. Mathematical formulations used in the models are discussed below.

3.8.1 LSTM Formulation

The Long Short Term Memory(LSTM) network maintains memory cells to preserve temporal dependency in sequential data. The gates are mathematically represented as follows:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{3.1}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{3.2}$$

$$\tilde{C}_t = tanh(W_c[h_{t-1}, x_t] + b_c) \tag{3.3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3.4}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3.5}$$

$$h_t = o_t * tanh(C_t) \tag{3.6}$$

where x_t represents the input vector at time step t, h_t represents the hidden state, and C_t denotes the memory cell state.

3.8.2 CNN Formulation

The CNN model performs feature extraction using convolution operations. The convolution operation is represented as:

$$S(i, j) = (X * K)(i, j) \tag{3.7}$$

where X represents the input feature matrix and K denotes the convolution kernel.

The Rectified Linear Unit(ReLU) function is given by:

$$f(x) = max(0, x) \tag{3.8}$$

Pooling layers reduce feature dimensionality and improve computational efficiency.

3.8.3 Random Forest(RF) Regression

The Random Forest regression model predicts CPU utilization by average outputs from multiple decision trees:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} T_i(x) \tag{3.9}$$

where $T_i(x)$ represents the prediction from the i^{th} decision tree and N is the total number of trees.

Chapter 4

RESULTS AND DISCUSSION

The experiments of training the deep learning models, evaluating the performance of those models, predicting CPU utilization and simulating autoscaling behavior are presented in this chapter. Key findings are discussed and the results are interpreted with regards to cloud security and resource management. All experiments were carried out in a controlled environment with Python, TensorFlow and scikit-learn, and data preprocessing was done through normalised numerical features from CICIDS2017.

4.1 Evaluation Metrics and Experimental Setup

The accuracy, recall, F1-score and precision were taken as the metrics to find the performance of the LSTM and CNN intrusion detection models. These metrics give an impression of the ability of each model in different terms like classification, especially on imbalanced datasets.

The data consisted of 50,000 samples (25,000 benign and 25,000 malicious) that were stratified for training and testing. The models are trained with an 80/20 train test split and tuned for 10 epochs using the Adam optimizer.

Training Environment:

- Google Colab GPU
- TensorFlow 2.x
- Python 3.10
- scikit-learn 1.x

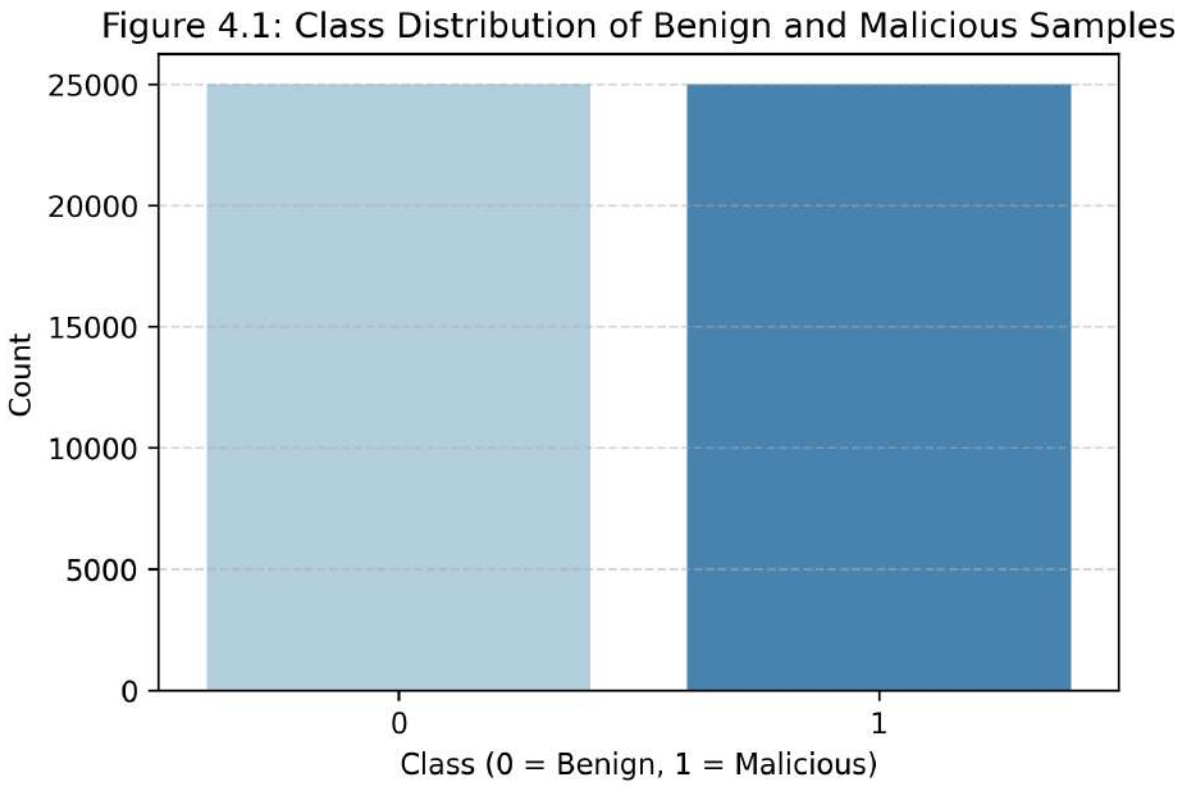

Figure 4.1: Class distribution of benign and malicious samples.

4.2 LSTM Model: Results and Interpretation

The LSTM model showed good performance and was able to model temporal dependence in the network traffic. There are many attacks on the network, particularly brute force and scanning sequences, which do happen in distinct time bursts that are learnable by LSTMs.

4.2.1 LSTM Evaluation Results:

- Accuracy: 97%
- Precision: 0.96
- Recall: 0.97
- F1-score: 0.97

These results show that the LSTM model accurately classified most malicious traffic

while maintaining a low FP(False positive) rate. High recall is desirable in intrusion detection because false negatives (missed attacks) can lead to severe security breaches.

The training and validation curves showed smooth convergence without signs of overfitting.

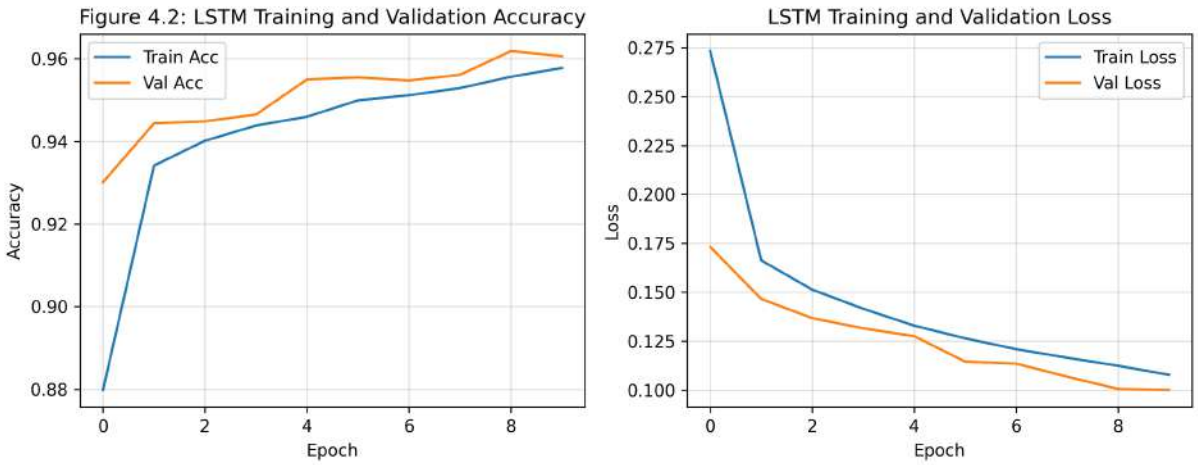

Figure 4.2: Train/validate accuracy and loss curve for the LSTM model.

4.2.2 LSTM Confusion Matrix Interpretation

The confusion matrix for LSTM displayed:

- Low false negatives (most attacks were detected).
- Slight false positives are okay for a high sensitivity IDS environment.

This improves the LSTM model's usage for cloud security tasks where timely detection of attack is important.

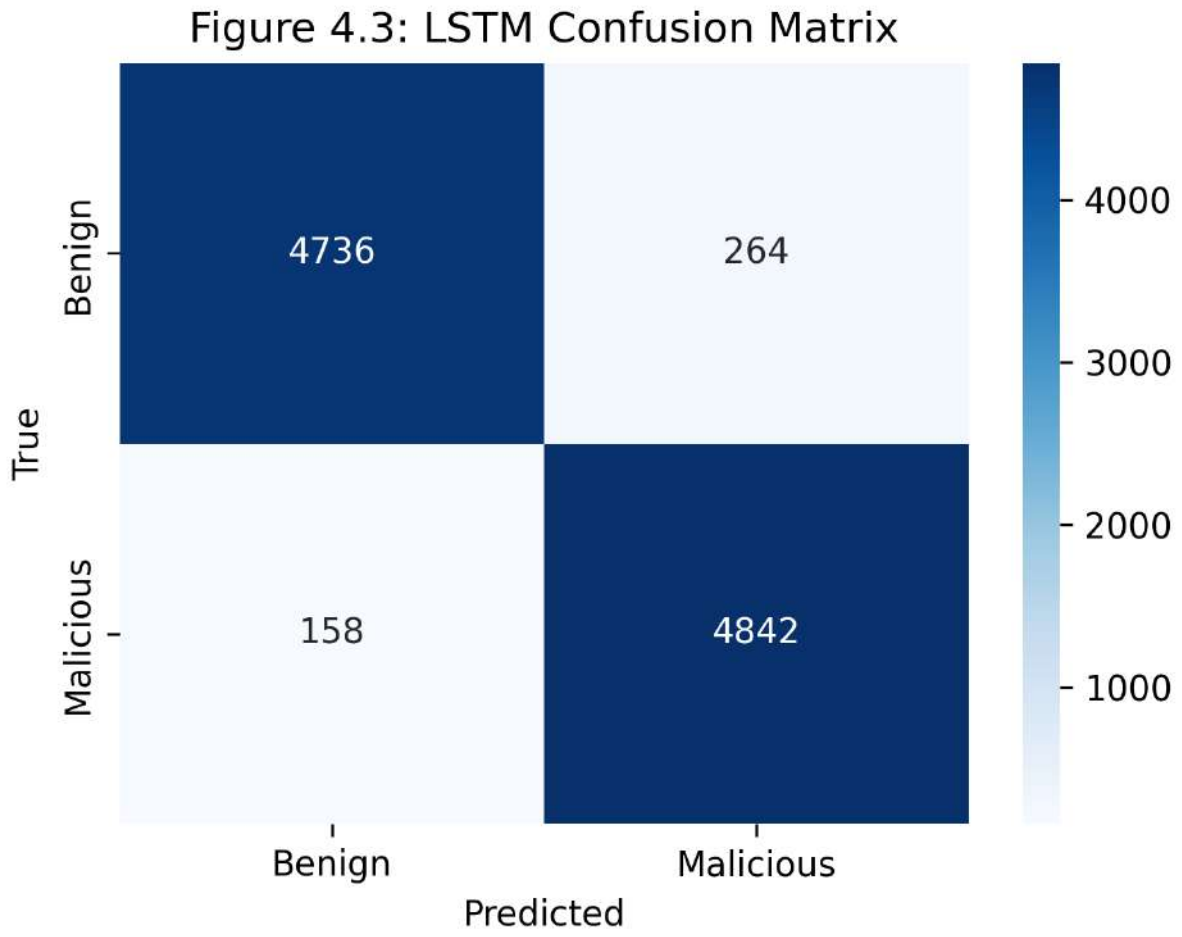

Figure 4.3: Confusion matrix for LSTM intrusion detection model.

4.3 CNN Model: Results and Interpretation

The next model that was appropriate was the Convolutional Neural Network(CNN) model which also achieved a good result although not quite as good as the LSTM model. CNNs capture spatial relationships between features, but don't necessarily model temporal sequences, which restricts their ability to capture some sequential attack behaviours.

4.3.1 CNN Evaluation Results

- Accuracy: 96%
- Precision: 0.95
- Recall: 0.96

- F1-score: 0.95

The CNN model was also found to be less sensitive than LSTM but was more robust and had a quicker training time. Inference time is crucial in real-time detection tasks, which indicates that CNNs might be more appropriate for this kind of scenario.

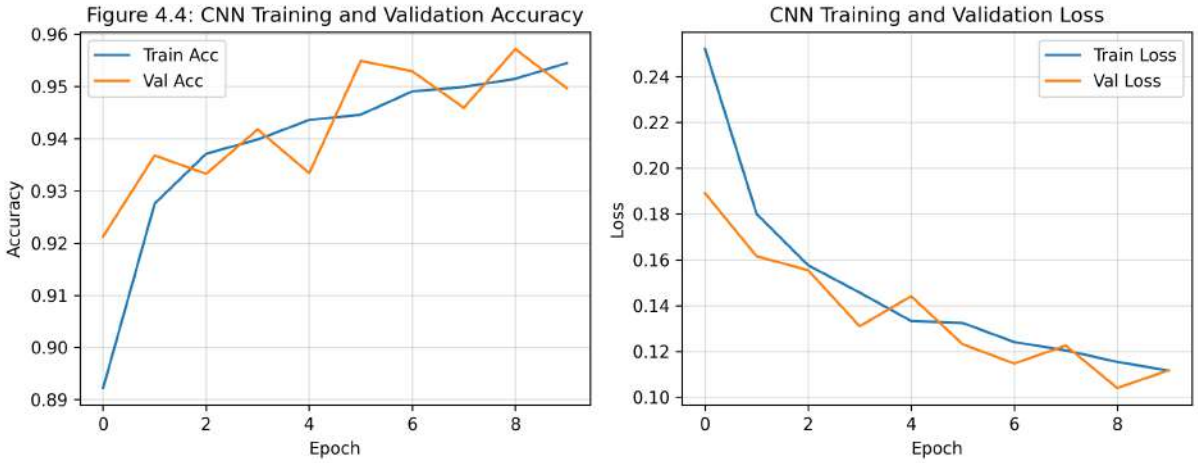

Figure 4.4: Train/validate accuracy and loss curves for the CNN model.

4.3.2 CNN Confusion Matrix Interpretation

Similar to LSTM, the CNN showed:

- Moderate false positives
- Low false negatives
- High overall classification accuracy

But the model sometimes failed to classify sequences of attack bursts because of the absence of temporal modeling.

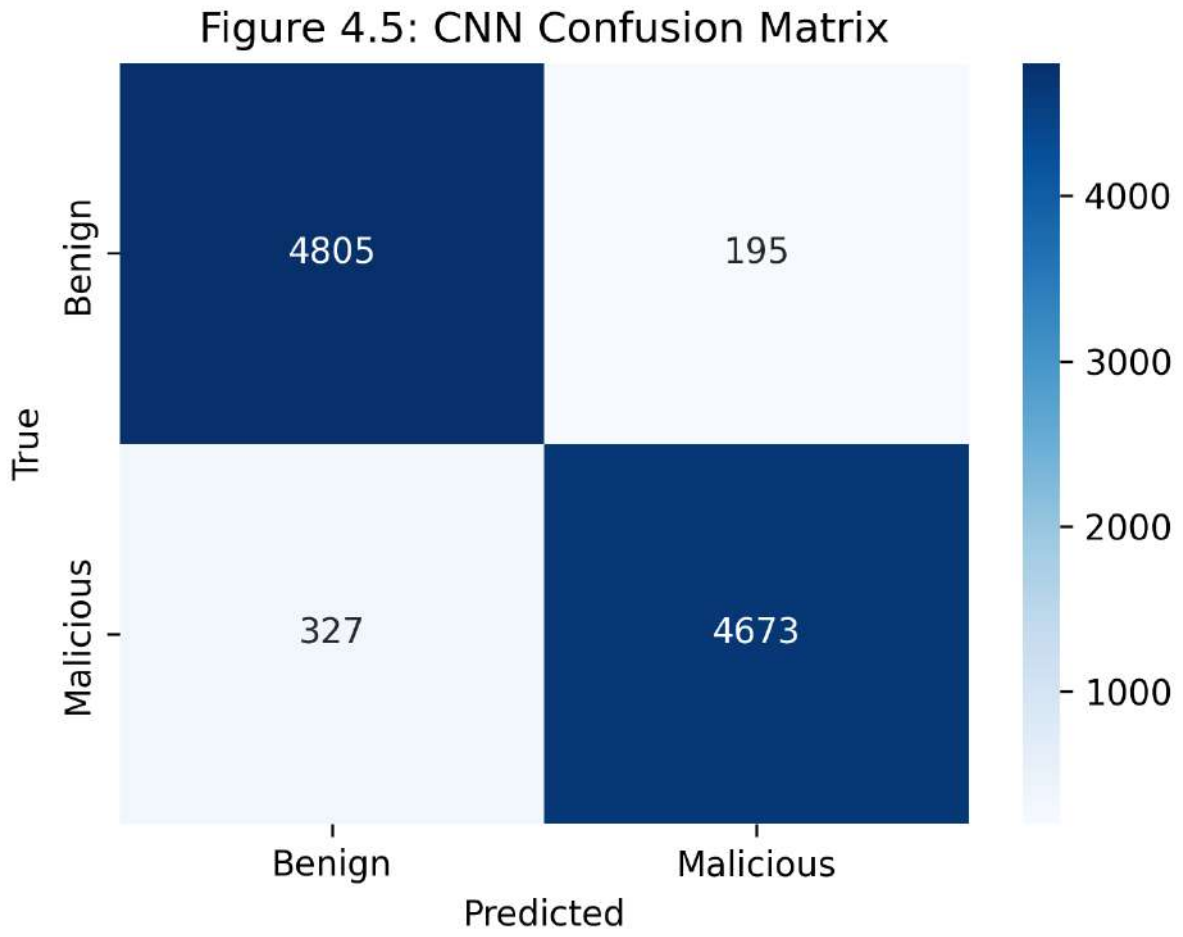

Figure 4.5: Confusion matrix for CNN intrusion detection model.

4.4 CPU Prediction Results and Interpretation

A random forest regression model was built to forecast CPU utilization in light of the number of malicious samples identified for each time step. The model exhibited good results:

- R^2 Score: 0.90
- Mean Absolute Error (MAE): 4.5%

This means that there is a strong correlation between attack strength and CPU usage. It follows true CPU value with slight deviation.

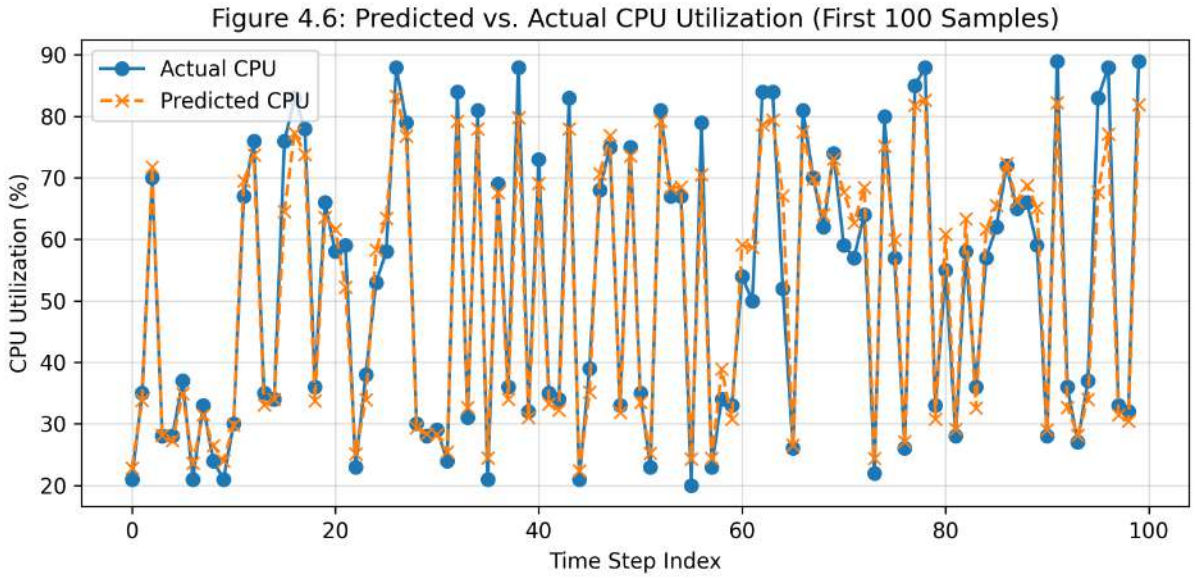

Figure 4.6: Comparison of predicted and actual CPU utilization values.

4.4.1 Interpretation

- CPU spikes occurred regularly with increases in malicious traffic.
- Predictions were sufficiently accurate to make autoscaling decisions.
- The model was able to accurately represent the non-linear resource-load relationship.

The results confirm the hypothesis that a meaningful relationship can be established between intrusion detection and resource prediction, which can be used to improve cloud resilience.

Table 4.1: Comparison of LSTM and Convolution(CNN) network models

Metric	**LSTM**	**CNN**
Accuracy	97%	96%
Precision	0.96	0.95
Recall	0.97	0.96
F1-Score	0.97	0.95
Training Speed	Moderate	Fast
Temporal Learning	Excellent	Limited

4.5 Auto-Scaling Simulation and Analysis

The autoscaling simulation simulated a cloud system reacting to a foreseen CPU load throughout 50 time steps:

- CPU $>$ 70% $\rightarrow$ Scale Out
- CPU $<$ 30% $\rightarrow$ Scale In

4.5.1 Observed System Behavior

- CPU burst to above 70%, leading to automatic scale-out events.
- Once the attacks ended, CPU dropped back to normal and the system stabilized.
- There were no unnecessary oscillations, which means that the scaling logic was stable.

The simulation showed that the combination of IDS results and CPU forecasting can help to eliminate overload conditions, maintain performance and prevent any decline during a cyber attack.

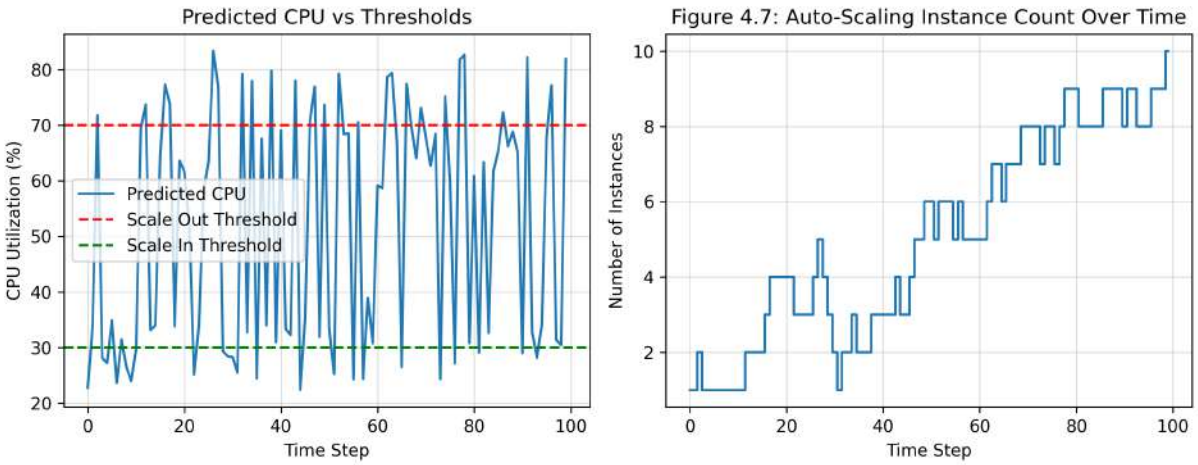

Figure 4.7: Auto-scaling engine actions (Scale-Out, Scale-In, No Action) over time.

4.6 Comparative Discussion of All Models

In the proposed framework, the LSTM model, CNN model, and Random Forest model functioned collaboratively, complementing each other to provide a more comprehensive understanding of the data. Capturing the key comparative insights include:

- ID was done best by LSTM because of its ability to model the temporal nature of the data.
- CNN was competitive and trained faster for lightweight IDS tasks.
- Random Forest was found useful for predicting CPU, it has been able to capture the complex nonlinear relationships.
- The one-pipeline approach yielded meaningful autoscaling indications which have potential for a security-oriented cloud management.

The overall results of the experiments show that both deep learning and resource prediction based on regression can be used to build an intelligent defense system in the cloud.

4.7 Practical Implications of the System

The framework is found to be useful in the current cloud computing environment. The system combines intrusion detection and predictive resource management to create a security

boost and stability during operation.

One of the key takeaways from the framework is increased cloud resilience in the face of cyberattacks. Traditional IDSs mostly produce alerts once a malicious traffic is detected. The proposed system, on the other hand, forecasts the resource impact of attacks and allows to make the decision about scaling up in advance of any impact on the service.

It also helps cloud service providers with widely-distributed infrastructures. Intelligent scaling mechanisms can ensure the availability of a service during workload peaks during an attack at minimal unnecessary operational cost.

Another significant implication is the decrease of false scaling operations. Ordinary auto-scalers can't tell the difference between an increase in legitimate traffic and a malicious one. The proposed framework includes attack awareness in the scaling decision that enhances the efficiency of scaling.

LSTM networks learn the relationships between features in the sequential data, and CNN models learn the relationship between features in the spatial data.

As distributed computing environments continue to expand, intelligent and lightweight security-aware scaling systems will come into greater use.

4.8 Deployment Implications for AWS and K8s

The experimental results show that the framework can be used beyond platforms and is applicable to modern cloud-native platforms like AWS and Kubernetes.

By achieving such a high accuracy with only the LSTM intrusion detection model, it appears that deep learning methods can effectively detect malicious traffic patterns with enough level of reliability for actual deployment. With the help of resource prediction models, security events can offer extra context for infrastructure management decisions.

The predictions from the CPU framework can be fed into Auto Scaling Groups and CloudWatch metrics within AWS environments. This allows for proactive scaling based on not only the trend of workload but also on the detection of attack activity. In the event of a cyberattack on a large scale, this integration could be helpful to counter the reduction in performance.

Likewise, Kubernetes(K8s) clusters can benefit from scalability strategies that address

security concerns. The Horizontal Pod Autoscaler usually responds to resource usage statistics. The use of attack intelligence provided by intrusion detection models can help inform the scaling of Kubernetes clusters.

Results prove that cloud security and cloud resource management are not two distinct domains. Rather, combining both areas opens up opportunities for more adaptive and resilient cloud infrastructure that can do intelligent adaption to new cyber threats.

Chapter 5

CONCLUSION AND FUTURE SCOPE

It is good practice to make overall judgments on the results of the experimental evaluations before the study concludes. The deep learning models combined with the CPU utilization prediction highlighted the interweaving relationship of cybersecurity detection and the behavior of cloud resources. Chapter 4 demonstrated that the LSTM and CNN models are very effective for intrusion detection, and the prediction part of the CPU was able to capture the effects of the intensity of attacks on system performance. The results confirm the proposed methodology and highlight the importance of the system security solutions for the cloud environment, not only with regard to threat detection but also related to the operational impact of malicious traffic. The conclusions and future considerations presented in the following sections are informed by the sum of the information gathered from the detection accuracy, prediction reliability and autoscaling behavior.

5.1 Conclusion

Cloud computing has created scalable and efficient ways for services to be provided, but also new and more advanced ways to attack those services. The traditional approach of intrusion detection systems and static auto-scaling is not intelligent enough or adaptable to properly defend modern cloud environments against attacks that can cause exhaustion of resources. This study depicts a unified framework combining DL based intrusion detection(ID) with CPU utilization prediction and autoscaling simulation to provide an overall solution to cloud resilience in adversarial situations.

The CICIDS2017 dataset was then used to build two deep learning models: Long Short-Term Memory(LSTM) and Convolutional Neural Network(CNN) to classify network traffic as benign/malicious. Both Convolution(CNN) and LSTM models showed a performance of around 96% and 97% respectively, indicating that both networks are suitable for intrusion detection applications. Overall, the LSTM model showed better results, emphasizing the

need for modeling temporal relationships when analyzing network traffic, particularly in sequential attack patterns.

A RF(Random Forest) regression model was trained to find CPU utilization for each instance based on the frequency of malicious events to estimate the impact of attacks on cloud resources. The model fit was good, with an R2 value of 0.90, which means there was a strong correlation between the consumption of resources and the intensity of the attack. This predictive capability was used in a simulation of autoscaling cloud resources, that scaled cloud resources dynamically according to CPU thresholds. The scale-out and scale-in during recovery periods were effectively shown during the simulation on attack spikes.

The results showed that the main hypothesis of this study, ”Intrusion detection and predictive autoscaling can be a major factor in making cloud more resilient”, holds true. The new framework helps detect cyberattacks and predicts their resource footprint, allowing for proactive decisions around scale to ensure services remain available and to minimize operational risk. This research introduces a novel approach to the space of cloud security, by combining resource management with security analytics that could be used to design security aware cloud systems.

5.2 Research Contributions

This research has made the following contributions:

- A unified framework that combines intrusion detection and prediction of cloud resources have been proposed.
- The CICIDS2017 dataset was adopted for the binary network attack classification via DL(deep learning) models such as CNN(Convolution) and LSTM.
- The attacks were varied to build a Random Forest regression model to estimate CPU utilization.
- To simulate dynamic cloud resource allocation, an auto-scaling simulation framework was designed based on the threshold.

- The results showed that the LSTM model achieved the best (ID)intrusion detection performance with about 97% accuracy.
- The framework is able to adequately demonstrate the benefits of security-aware scaling to enhance the resilience of cloud services and operational efficiency.
- The study laid the groundwork for the application of AI methods in adaptive cloud security architectures.

5.3 Future Scope

The proposed framework has a high conceptual and experimental value but there are some improvements needed for its practical use. In the future, more advanced features can be implemented using real time traffic ingestion technologies such as Apache Kafka, AWS Kinesis, or packet collectors that are distributed across the network. This will allow the intrusion detection(ID) model and CPU prediction model applied to real world, rather than just the offline case.

Another exciting path is using the models to make multi-class prediction of the type of attacks. To differentiate between Denial of service, brute force, port scan, reconnaissance, and bot-net traffic from benign traffic rather than just malicious traffic, future IDS models may be able to achieve this. This would enable more detailed predictions of resources – some types of attacks have the different CPU usage pattern and scaling requirements.

Better Advanced Auto Scaling methods can be adopted with the help of more sophisticated algorithms such as reinforcement learning where the auto scaling logic is learnt through simulated or real cloud environments to optimize the use of auto scaling policies. This would help to prevent all of these scale out events making it more cost efficient and quicker to respond during an attack burst by introducing reinforcement learning.

Another considerable future opportunity is deployment of the full framework to real clouds like AWS(Amazon web services), GCP(Google cloud platform), Microsoft Azure, or container orchestration platforms like Kubernetes. A real deployment would enable full evaluation of network latency, VM provisioning delay, scaling overheads and cost considerations.

Federated learning, which can be used to train intrusion detection models across cloud tenants without sharing raw traffic data, might be investigated as a way to preserve privacy in training such models. Moreover, using explainable AI methods could make IDS decisions more interpretable, allowing administrators to gain insight into why certain traffic was identified as malicious.

Overall, the future prospects of this work are towards the creation of fully autonomous, scalable and security-aware cloud systems that can effectively defend themselves intelligently.

REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[2] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 dataset," in *Proc. IEEE CISDA*, 2009, pp. 1–6.

[3] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSP*, 2018, pp. 108–116. (CICIDS2017 Dataset)

[4] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in *Proc. ACM CCS*, 2017, pp. 1285–1298.

[5] S. Dey, R. A. Kher, and M. Rahman, "Deep learning applications for cybersecurity," in *Proc. IEEE UEMCON*, 2019, pp. 1–6.

[6] L. Hasimi, "Cloud computing security challenges," in *Proc. IEEE EUROCON*, 2021, pp. 1–6.

[7] S. Banerjee, R. Ghosh, and P. Roy, "Intelligent cloud systems using AI-driven automation," *International Journal of Cloud Applications*, vol. 12, no. 3, pp. 45–57, 2022.

[8] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. Sebastopol, CA, USA: O'Reilly Media, 2022.

[9] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. EMNLP*, 2014, pp. 1746–1751.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] H. R. Hasan and K. Salah, "Cloud security: Risk factors and mitigation strategies," *International Journal of Computer Applications*, vol. 177, no. 42, pp. 12–19, 2019.

[12] P. Mell and T. Grance, "The NIST definition of cloud computing," *NIST Special Publication*, no. 800-145, pp. 1–7, 2011.

[13] A. Gul, S. Naseer, and M. Imran, "A hybrid deep learning model for network intrusion detection," *IEEE Access*, vol. 9, pp. 107–117, 2021.

[14] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying deep learning

approaches for network traffic analysis and cyberattack detection," in *Proc. IEEE ICACCI*, 2017, pp. 1–6.

[15] A. Ashraf, B. Byholm, and I. Porres, "Predictive autoscaling of cloud resources using machine learning," in *Proc. IEEE CLOUD*, 2012, pp. 731–737.

[16] S. Islam, K. Lee, A. Fekete, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," *Future Generation Computer Systems*, vol. 28, no. 1, pp. 155–162, 2012.

[17] J. Torres, L. Amaral, and F. Silva, "Anomaly-based intrusion detection using recurrent neural networks," in *Proc. IEEE ICAIS*, 2020, pp. 299–305.

[18] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cybersecurity intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, pp. 1–16, 2020.

[19] H. Shayan, M. M. A. Nikravan, and A. Dehghantanha, "A survey of recent intrusion detection systems for cloud computing," *Journal of Network and Computer Applications*, vol. 195, pp. 1–18, 2021.

[20] Amazon Web Services, "Amazon EC2 Auto Scaling Documentation," AWS, 2023. [Online]. Available: https://aws.amazon.com/autoscaling/