

Enhancing Cloud Security in the Financial Sector: An AI-Driven Threat Detection and Response Framework

MSc Research Project
MSc in Cybersecurity

Srinivas Dammu
Student ID: x23228733

School of Computing
National College of Ireland

Supervisor: Raza UI Mustafa

National College of Ireland
MSc Project Submission Sheet



School of Computing

Student Name: Srinivas Dammu
.....
Student ID: X23228733
.....
Programme: MSc in Cybersecurity
.....
Module: MSc research project
.....
Supervisor: Raza UI Mustafa
.....
Submission Due Date: 12/12/24
.....

Project Title: Enhancing Cloud Security in the Financial Sector: An AI-Driven Threat Detection and Response Framework
.....

Word Count: 6616
.....
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:

Date:

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
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.	<input type="checkbox"/>

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):	

Enhancing Cloud Security in the Financial Sector: An AI-Driven Threat Detection and Response Framework

Srinivas Dammu

X23228733

Abstract

The growing number of cyber-attacks targeting the financial sector because of expanding digital payment solutions requires improved security defenses. Researchers introduce a dual-level Intrusion Detection System (IDS) specifically tailored for protecting the cloud framework of financial industry operations. The system integrates a K-Nearest Neighbors (KNN) model for binary classification, achieving 92.79% accuracy in detecting benign and malicious traffic, and a Deep Neural Network (DNN) for multiclass classification, reaching 94.19% accuracy in identifying seven attack types: Bot, DDoS-HOIC, DoS-Hulk, SSH-Bruteforce, DoS-GoldenEye, Infiltration, and DDoS-LOIC-HTTP. The research utilized the CICIDS2018 dataset for both training machines and testing performances while preprocessing procedures such as feature selection together with scaling and balanced sampling resulted in optimal system outcomes. Our system delivers powerful security while maintaining visibility through a simple interface that tracks live threats and displays transaction progress along with attack information. The proposed solutions in this research extend Intrusion Detection Systems (IDS) to tackle essential cybersecurity threats existing within financial cloud network environments.

1 Introduction

The quick expansion of digital payment technologies by financial institutions has boosted electronic transactions which now face heightened risks from complex cyber threats. Fares and colleagues revealed in their 2023 research that artificial intelligence plays a vital security enhancement role in financial institutions. Existing intrusion detection systems encounter multiple difficulties because of data imbalance problems together with real-time inefficiencies and growing scalability challenges as cloud environments become more complex. This research aims to address these gaps by answering:

How can 2-level IDS enhance real-time threat detection and response in financial cloud infrastructure?

State-of-the-art IDS systems face difficulties when trying to perform accurate detection and classification of complex attack types including DDoS attacks along with DoS and Brute Force forces. This research develops a robust 2-level IDS leveraging ML models: KNN for binary classification and DNN for multiclass classification.

Through this development project financial institutions can achieve enhanced cloud security when utilizing advanced IDS solutions that deliver precise results alongside scalable and efficient real-time threat detection capabilities. CICIDS-2018 evaluates models based on their ability to scale and accurately detect and classify security threats in financial cloud networks.

Financial institutions will benefit from improved cloud security through this project which delivers a real-time threat detection and response IDS that is more reliable, scalable and efficient than current systems. The CICIDS-2018 dataset functions as an evaluation benchmark for network security models operating in financial cloud settings by examining their scalability alongside accuracy-based network anomaly detection capabilities.

This research presents new knowledge in the field by revealing an innovative IDS framework which together with a payment gateway incorporates multiple defense layers to prevent fraudulent transactions. The report is structured as follows: In Section 2 by analyzing related work dedicated to IDS in cybersecurity followed by research methods presentation in Section 3 system design exposition in Section 4 implementation details in Section 5 evaluation results along with inference and Front-end discussions in Section 6 concluding with major findings and research directions for the future in Section 7.

2 Related works

Developers created sophisticated IDS systems to protect critical financial sector assets because advanced cyber threats become increasingly complex. Following a review that both analyzed strengths and weaknesses this section identified gaps present in existing solutions emphasizing research requirement.

ML and DL for IDS

The deep learning-based IDS presented by Awajan, A. (2023) achieved 93.74% accuracy on IoT network data through a four-layer DNN but lacks clear path to application in financial environments. Azizan, A.H. et al. (2021) showed SVM performance at 98.18% accuracy but found its computational cost restrictive for real-time execution in high-traffic cloud landscapes. Balamurugan, E. et al. (2022) developed the IDSGT-DNN framework which merges game theory with DNNs for better detection accuracy results alongside decreasing false positive rates yet it faces practical deployment barriers for financial network real-time detection because of its system complexity. The combination of genetic algorithms and particle swarm optimization by Balyan, A.K. et al. (2022) helped establish balanced datasets which reached 98.98% accuracy while their results showed uncertain performance levels within dynamic financial systems. The LSTM model by Farhan, B.I., and Jasim, A.D. achieves 99% accuracy in IoT intrusion detection yet faces real-time deployment barriers due to computational costs. High accuracy levels were achieved in Ghanem, W.A.H. et al.'s hybrid IDS which uses artificial bee colony algorithms alongside MLP but potentially lengthens cloud infrastructure response time. Kim, J. et al. (2020) demonstrated CNN approach for DoS attack detection with high results of 99% for KDD and 91.5% for CSE-CIC-IDS2018 while facing issues of handling financial

high-dimensional data. Li, X. et al. (2022) reached improved anomaly detection results through a combination of deep learning and random forests yet displayed limitations with regards to scalability in extensive financial systems. EOS-IDS was proposed by Mayuranathan, M. et al. (2022). to protect cloud environments achieving 97.22% detection accuracy but scalability and processing speed remain unresolved problems. Ullah, S. et al. (2022) demonstrated high performance of LSTM and GRU integration in IoV networks yet showed uncertain results for financial network application.

Application and Optimization of IDS in Various Environments

Research conducted in numerous studies worked to enhance IDS exclusively for distinct operational settings. Chang, V. et al. (2022) examined cloud and fog computing security policies yet uncovered that financial sector requires its own set of techniques to protect sensitive information and deliver prompt responses. IoT devices receive security benefits from the host-based IDS suggested by Gassais, R. et al. (2020) but this solution fails to protect cloud-based financial systems effectively. Islam, U. et al. (2022) reported that SVM exceeded both KNN and RF in detecting DDoS attacks inside financial networks by reaching 99.5% accuracy yet its computational demands prevent use in real-time systems. Research from Jumabek, A. et al. (2021) supported advanced network intrusion detection through CatBoost demonstrating enhanced results compared to standard tree-based methods but the effectiveness in real-time financial operations requires further testing. Systems network design researchers Latah, M. and Toker, L. (2020) presented a hybrid SDN system that reached 84.29% accuracy yet faced difficulty adapting to financial industry infrastructures. Milosevic, M.S. and Ciric, V.M. (2022) applied DNNs to handle imbalanced data for threat detection in financial systems yet run-time implementation has yet to be fully explored. Pelletier, Z. and Abualkibash, M. (2020) demonstrated a 96% accuracy in feature selection through R programming which requires substantial modifications for dynamic financial system applications. Stiawan, D. et al. (2020) achieved 99.86% accuracy through feature selection in anomaly detection but require additional research to evaluate real-time processing effects in financial datasets.

Challenges in IDS Systems and Future Directions

Research work identified numerous technical obstacles including unbalanced data sets feature selection process plus requirements for real-time system deployment. According to Dietz, K. et al. (2024) financial sectors face major obstacles regarding accessible data availability together with transparency requirements and usability problems specifically because data privacy remains essential. Fares, Butt, and Lee (2023) review of banking AI described essential issues including data privacy and regulatory compliance as foundations for developing IDSD systems. Ghanem, W.A.H. et al. (2020) indicated that developing advanced feature selection methods is essential for enhancing IDS efficiency which represents a critical research area for sophisticated financial systems. The financial sector needs to detect subtle threats which matches Maseer Z.K. et al.'s (2021) recommendation to improve anomaly-based IDS with DL developments and enhanced feature selection approaches. The study by Thakkar, A. and Lohiya, R. (2020) showed updated datasets are crucial for detecting evolving cyberattack

patterns in the evolving financial sector. Zhang, C. et al. (2022) explored problems associated with novel data when balanced distributions are absent which poses significant issues in real-time detection systems of financial cloud infrastructures.

IoT and Smart Systems Security

The research field includes multiple studies targeting Intrusion Detection Solutions specifically for IoT technology and smart systems execution traits. Ghanem, W.A.H. et al. (2020) introduced a hybrid IDS for IoT networks but the research left unclear its deployment possibilities within financial system frameworks. Süzen, A.A. (2021) hybrid Deep Belief Network (DBN) model reached 99.72% accuracy in ICS intrusion detection yet requires refinements to operate effectively within financial cloud systems. The research by Farhan, B.I. and Jasim, A.D. (2022) showed that LSTM networks can achieve high accuracy for IoT intrusion detection under dataset imbalance conditions yet struggle with real-time large-scale financial system performance because of computational intensity.

Critical Analysis of ML and DL Techniques in IDS

Model Performance and Dataset Relevance: While the DL IDS introduced by Awajan et al. (2023) reached 93.74% accuracy levels across all communication protocols, its authors did not test its scalability for extensive IoT network deployments. The research of Azizan et al. (2021) revealed SVM achieves 98.18% accuracy while facing problems with both scalability limits and misleading false positives in complex IoT networks. LSTM delivered 99% accuracy per Farhan and Jasim (2022) but its high computational resource requirements restrict its use in low-resource environments and motivate the search for alternative methods like KNN and DNN.

Hybrid Approaches and Feature Optimization: Balyan et al. (2022) demonstrated the effectiveness of their feature engineering approach with EGA-PSO for dataset balancing together with IRF for feature selection because they reached 98.98% accuracy showing KNN benefits from well-prepared feature sets. The AE-IDS framework developed by Li et al. (2022) demonstrates strong efficiency yet shows limited ability to manage various intrusion attacks requiring DNN capabilities for improved flexibility.

Game Theory and Advanced Architectures: Balamurugan et al. (2022) reached better intrusion detection results by fusing game theory and DNN investigations in intricate environments.

Justification for KNN and DNN Models

KNN: Both Stiawan et al. (2020) and Islam et al. (2022) proved KNN works well with low-dimensional datasets because its implementation process stays straight-forward which supports feature-optimized environments.

DNN: Milosevic and Ciric (2022) demonstrated the efficiency of DNNs to handle both multidimensional and uneven datasets which leads to useful outcomes for hybrid threat detection systems as well as intrusions detection challenges.

Explainability and Transparency: The aims to improve ML-based IDS interpretability after Dietz et al. (2024) uncovered explainability issues in the field.

Scalability and Real-Time Performance: Detection models including research from Farhan and Jasic (2022) work in offline modes yet point to the necessity of real-time scalability in proposed methods.

Gap Analysis

Research demonstrates significant developments in the implementation of ML and DL algorithms within intrusion detection systems across IoT settings alongside cloud-based infrastructures and financial platforms. Research demonstrates that SVM along with RF and hybrid models achieves excellent accuracy in detecting cyber-attacks. Data imbalance alongside real-time detection requirements and scalability issues remain unresolved challenges. Present detection systems fail to provide continuous real-time threat identification alongside response activities needed to tackle dynamic security vulnerabilities in cloud-based financial systems. Existing IDS techniques demonstrate room for development because stronger solutions are required to achieve better performance and real-time response capability along with improved accuracy.

3 Research Methodology

In this work, the efficacy of machine learning models for detecting and classifying network traffic anomalies using the CICIDS-2018 dataset from the University of New Brunswick is assessed by means of a systematic approach.

The proposed methodology encompasses data collection, preprocessing, modeling, evaluation and analysis with appropriate objective function selected to achieve reproducibility and making meaningful statements about cross cuts into possible future operations.

3.1 Research Procedure

Dataset Description and Preparation:

Dataset: The study employs the CICIDS 2018 dataset because this collection offers labeled examples of both benign behaviors and multiple attack patterns such as DoS and DDoS. Network intrusion detection researchers benefit from this dataset because it provides perfect training material for machine learning models. This dataset strings together extensive modern network traffic situations together with a comprehensive selection of financial infrastructure attacks.

Data set link: <https://www.unb.ca/cic/datasets/ids-2018.html>

Data preprocessing: The Data preprocessing sequence applied feature selection based on the correlation matrix for maintaining significant attributes and performed missing value imputation and consistency checks followed by Min-Max scaling normalization methods to ensure dataset compatibility with machine learning models while additionally choosing attack-representative features for model training.

3.2 Experimental Setup:

The detection process utilized KNN binary classification to identify benign traffic while a DNN multiclass approach broke down different attack types in malicious traffic. The experimental setup was configured as follows:

Hardware: The ML models were developed and evaluated on computer hardware consisting of an Intel i7 CPU with 16GB of RAM.

Software: For software implementation researchers activated Python 3.8 together with the machine learning libraries TensorFlow and Scikit-learn along with data analysis tool Pandas and plotting library Matplotlib.

Tools: The data implementation work was conducted through Jupyter Notebook combined with Wireshark serving edge-level scrutiny in dataset evaluation.

3.3 Model Development:

Data distribution involved an 80-20 split between training models and validation purposes during model implementation.

K-Nearest Neighbors (KNN): KNN is used as the first level classifier for signing out the suspicious traffic. The model is trained with the data, and it works to tune the parameters like the number of neighbors (k) some techniques are like grid search cross section. The performance of proposed model is measured using basic measures including accuracy, precision, recall and F1 score.

Deep Neural Networks (DNN): When initial traffic classification detects potential threats a DNN model performs secondary analysis to determine suspicious traffic patterns. Reactive Unit functions drive multiple hidden layers in the DNN architecture which leads to a logistic function softmax layer for classification output. Through the combination of backpropagation and gradient descent methods, trainers process and refine model decisions on dataset information. The research improves classification effectiveness through parameter optimization which includes learning rate settings coupled with hidden layer quantity and batch dimension.

3.4 Contribution and Rationale for Methodology

Intrusion Detection Systems benefit from this research because it uses dual-layer IDS architecture that functions through K-Nearest Neighbors for binary categorization alongside

Deep Neural Networks which classify multiple attack types. Our hybrid method achieves higher detection accuracy for both untainted traffic and security threats across multiple critical attack categories including DoS and DDoS and SSH Brute Force attacks which remain fundamental to network protection today.

Why the CICIDS-2018 Dataset Was Selected: Because it provides real-world network representations through extensive feature sets and various financial institution relevant attack scenarios across 80 different dimensions the researchers selected the CICIDS-2018 dataset. The large volume of properly classified data sets available makes this source perfect for evaluating intrusion detection systems in response to current cybersecurity dangers.

3.5 Evaluation Methodology

Performance Evaluation:

Accuracy: It measures how many correctly identified instances (from both secure and attack classifications) match the overall instance count. The assessment of accuracy validates the model's performance quality when detecting traffic under normal conditions along with attack situations.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Samples}$$

Precision: how the model limits false positive detections while maintaining high identification rates for legitimate attacks.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Recall: This calculation shows how effectively the model identifies true positive intrusions. The recall metric proves essential for reducing the number of threats that the system fails to detect.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

F1-Score: The F1-Score emerges through the harmonic averages of precision and recall to measure model performance properly within unbalanced datasets.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{TPrecision \times Recall}$$

Confusion Matrix: The model's performance evaluation includes analysis from a confusion matrix which displays true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). Analyzing classification mistakes through this information reveals both model weaknesses and performance metrics.

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Figure 1 : Example Confusion Matrix

3.6 Explanation of Methodology

Inference System with IDS Response: The architecture of the inference system within IDS generates predictions about real-world behaviors through the application of raw data. The system tested the ability of certain inference pipeline elements to scale by subjecting it to different system traffic loads.

Front-End Design and Implementation: Its transaction monitoring environment delivers stable secure logins together with historical transaction tracking and real-time threat detection using class probability status updates. The design delivers smooth navigation features which allow seamless backend connectivity for threat categorization and warning generation.

Final Results and Analysis: Researchers compared two distinct models to evaluate two-level IDS performance for threat identification in financial cloud environments. Study results show that the best performing algorithm by way of performance testing charts which stored logs to enable duplicability evaluation.

4 Design Specification

This work presents Figure 2 as its system architecture which applies a multi-layered design methodology to ensure sturdy security support for financial transactions. The system begins operation by retrieving transaction logs which undergo pre-processing to manage missing data points normalize transaction values and screen features vital for precise threat detection. This process requires system analysts to use only essential data for their evaluations. Machine learning models at the first detection level recognize possible threats through analysis of historical transaction patterns. Further threat detection accuracy occurs through advanced pattern recognition techniques at the second level which improve and verify initial findings from earlier analysis. Once a threat is detected, the system triggers an automatic response protocol: The system offers automatic fraudulent transaction blocking along with instant notification emails to system administrators to enable a rapid response. Real-time protection for the payment system comes as an assured result of system operations.

The IDS system works together with the payment gateway to serve as a protective barrier between user input zones and transaction processing while scanning all activities for threatening behavior. A durable database preserves information about blocked card data for subsequent analytical review and internal examination. Web-based payment integration runs smoothly throughout the system architecture providing users with both secure transaction services and efficiency in payment handling. The IDS features a multi-layered protection

system that enables fully automated threat defense capabilities for real-time financial transactions using advanced analytics alongside live monitoring technology.

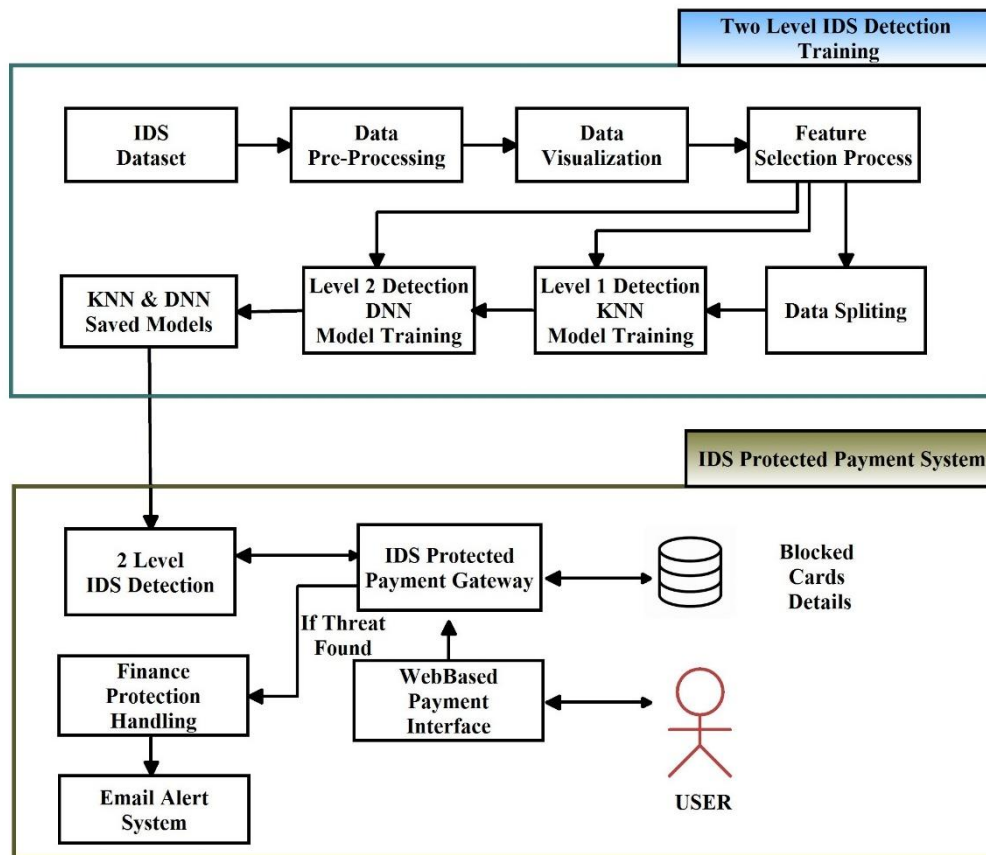


Figure 2: System Architecture

5 Implementation

This document analyzes techniques to implement detection and classification systems that recognize cyber threats. The system implementation concluded with uniting machine learning models together with both the front-end interface and backend system to allow users real-time threat monitoring.

5.1 Data Preprocessing

Researchers completed data preprocessing to maintain strong model performance combined with high operational efficiency. During the preprocessing sequence experts implemented feature selection through correlation matrices and removed redundant features while they cleaned data from missing values and normalized features via Min-Max scaling to improve modeling compatibility.

Data Loading and Analysis: Researchers analyzed uniformity by reading several parquet files for column names while studying labels and distributions of values. The system identified columns that did not match others to maintain data accuracy.

Data Sampling with Labeling: The researchers achieved equal class balance for binary classification through sampling which designated 'Benign' activities as 0 and malicious actions as 1. The sampling process incorporated exactly 20,000 records per classification label on condition that each category-maintained balance through adequate extraction from initial datasets containing no less than 10,000 records.

Feature Selection for Binary Classification: To match binary target (Category) data synthesis measured important features during selection optimization. The modeling retained features according to what emerged from a correlation matrix which analyzed significant features. The plotted graph of the selected features and their significance appears in Figure 3.

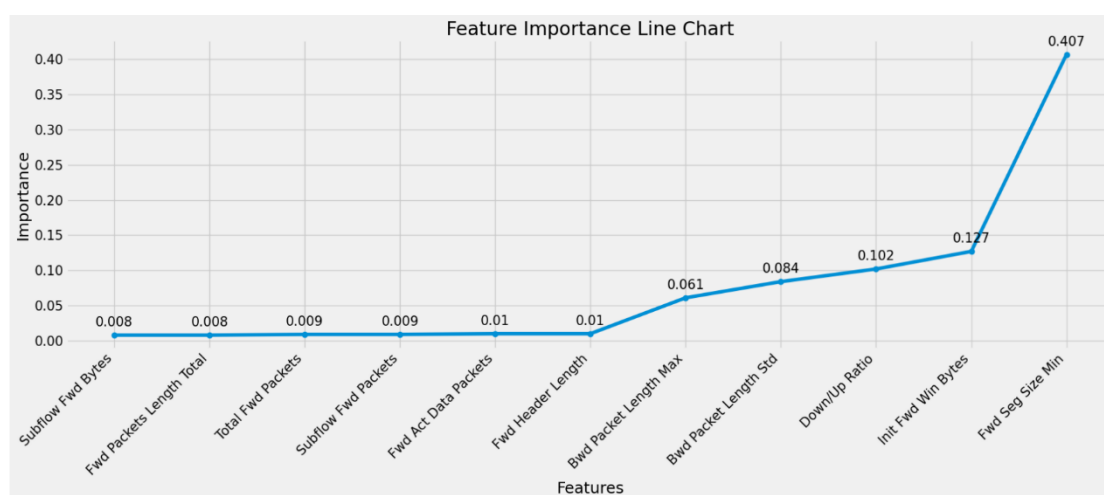


Figure 3 : Features of Boolean Classification

Feature Selection for Multiclass Classification: The classification task explored seven specific attack types including DoS – GoldenEye and Bot as well as DDoS – HOIC and DoS – Hulk paired with SSH – Bruteforce attack type alongside Infiltration and DDoS – LOIC – HTTP attacks. To train our models each attack type could have up to 20,000 records ensuring balanced dataset for model training. The importance of features in multiclass classification is shown in Figure 4.

Data Normalization: Min-Max scaling adjusted feature values of binary and multiclass datasets to lie within the range from 0 to 1. Each feature we added will contribute an equal role towards building our models.

Data Saving: The preprocessed datasets stored in Binary_df.csv and multiclass_df.csv have corresponding lists of selected features which we saved as .pkl files to enable reproducibility.

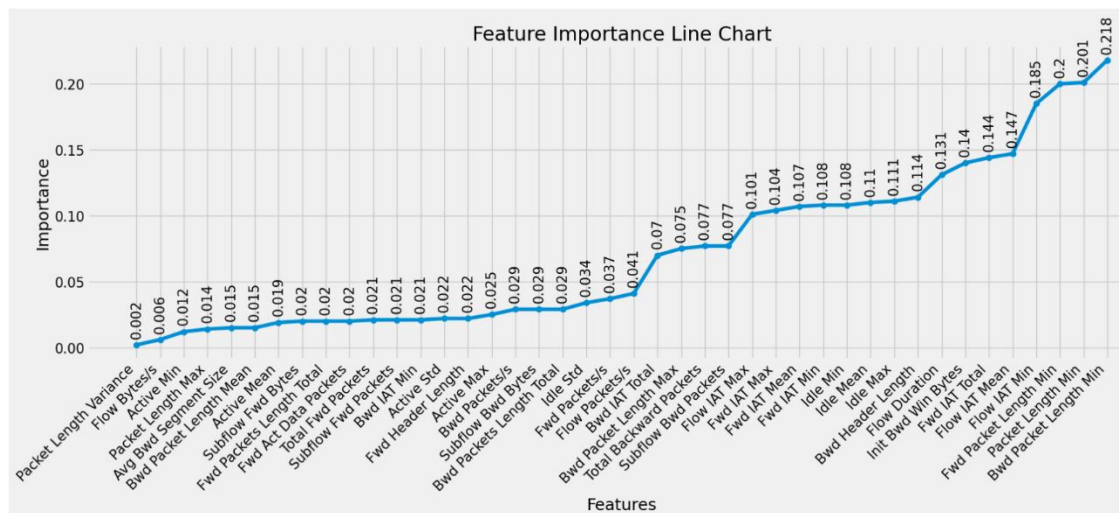


Figure 4 : Features of Multiple class Classification

5.2 Model Development

By Developing two machine learning systems to conduct traffic classification which consisted of the K-Nearest Neighbors and a Deep Neural Network. The KNN model manages traffic to binary classifications between benign and malicious whereas the DNN model examines each specific type of attack it encounters. The cyber-attacks in our full-scale trained both models used the CICIDS 2018 dataset to test their detection strength against attack variety.

Model Training

Multiclass Model Training: DNN received training on 140,000 rows of data containing 43 features through an 80% training split of 112,000 samples paired with a 20% testing split of 28,000 samples. Batch Normalization and L2 regularization were combined with multiple layers in a deep neural network which optimized its model using the Adam optimizer and measured errors through categorical cross entropy loss.

Binary Model Training: Binary classification tests applied a KNN algorithm with 280,000 rows of data plus 12 features separating training (224,000 samples) from testing (56,000 samples) at 80/20. The KNN model applied 11 neighbors.

5.3 Front-End Development

By developing an accessible front-end interface which gave live transaction status reports and threat detection alerts. Users remain both informed and secure since the dashboard live displays attack types next to their probabilities alongside information about transaction results. The system offers secure login capabilities together with intricate representations of its detection mechanism. HTML, CSS and JavaScript composed the front-end development that features backend integration to enable uninterrupted exchange with the trained models.

5.4 Backend Integration

A backend architecture built from Python and Flask functions to manage how the front-end talks to the ML models. This backend setup processes incoming data requests before conducting real-time classification with KNN and DNN methods then sends classification findings directly to the front end for user viewing. The backend system controls security alerts and records data logs to detect threats both promptly and precisely.

5.5 Output and Results

Multiple output results come from this implementation including Transformed data in Figure 2 & Figure 3 together with transaction security logs alongside real-time attack classification and probability score analysis. The final UI with threat detection feedback. This architecture delivers a scalable and secure processing method for surveillance of digital threats within the cloud environments of financial institutions.

6 Evaluation

The detection capabilities of classifiers for malicious and benign network data receive evaluation analysis here before this section presents information about inference steps alongside front-end interactions and user interface elements. This section shows classification accuracy metrics together with reports on class distribution while explaining login and transaction security systems process flow.

6.1 Result analysis

Evaluation of Binary Model classifier: This stage shows that the KNN Binary classifier model offers good classification results supported by 92.79% validation accuracy and strong precision and recall metrics for both classes which results in minimal misclassification.

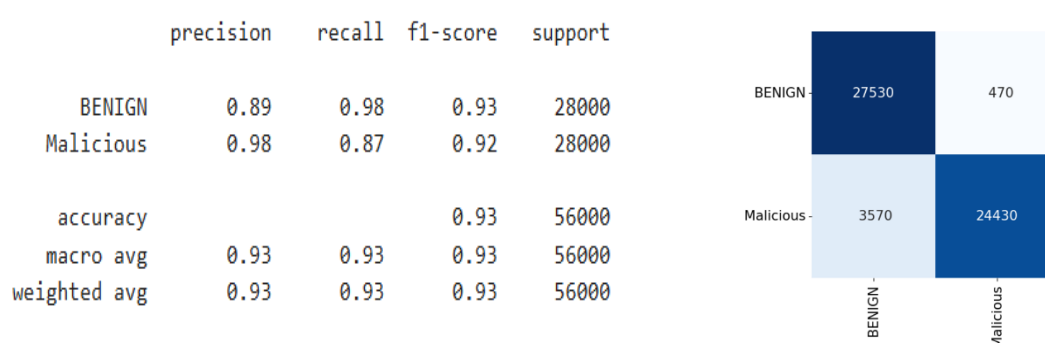


Figure 5 : Classification Report & Confusion matrix of KNN

Classification Report: The KNN model displayed in Figure 5 reached image classification accuracy of 93% while producing "BENIGN" metrics of 0.89 precision and 0.98 recall and "Malicious" metrics showing 0.98 precision and 0.87 recall. The model demonstrates balanced performance which surpasses the baseline benign traffic detection according to its macro average and weighted average values of 0.93.

Confusion matrix: Figure 5 right illustrates model results that provided correct classifications for 27,530 benign samples and 24,430 of malicious samples. The system incorrectly labeled 470 benign samples as malicious and 3,570 malicious samples as benign which revealed a larger number of false negatives of "Malicious" as an indication for enhancement need in cyberattack detection.

Evaluation of Multiclass model classifier: I achieved a validation accuracy rate of 94.19% in a DNN multiclass classifier for DoS attack detection system following some model parameter tuning. The classifier performance maintains consistency with new data which is supported by both training and validation graph analysis.

	precision	recall	f1-score	support
DoS_attacks_GoldenEye	0.96	0.90	0.93	4000
Bot	1.00	0.99	0.99	4000
DDoS_attack_HoIC	0.87	0.93	0.90	4000
DoS_attacks_Hulk	0.92	0.86	0.89	4000
SSH_Bruteforce	1.00	1.00	1.00	4000
Infiltration	0.95	0.98	0.97	4000
DDoS_attacks_LOIC_HTTP	0.90	0.94	0.92	4000
accuracy			0.94	28000
macro avg	0.94	0.94	0.94	28000
weighted avg	0.94	0.94	0.94	28000

Confusion Matrix							
DoS_attacks_GoldenEye	3585	0	0	0	0	0	415
Bot	0	3960	1	5	0	34	0
DDoS_attack_HoIC	0	1	3807	176	0	16	0
DoS_attacks_Hulk	2	2	665	3281	0	49	1
SSH_Bruteforce	0	0	0	0	3993	7	0
Infiltration	5	4	53	18	0	3880	40
DDoS_attacks_LOIC_HTTP	130	0	4	8	0	44	3814
	DoS_attacks_GoldenEye	Bot	DDoS_attack_HoIC	DoS_attacks_Hulk	SSH_Bruteforce	Infiltration	DDoS_attacks_LOIC_HTTP

Figure 6 : Classification Report & Confusion matrix of DNN

Classification Report: The DNN model displayed in figure 6 reaches a 94% detection accuracy for DoS attacks and records 0.96 precision and 0.93 F1-score with a recall rate of 0.90 for DDoS_attacks_GoldenEye from a sample of 4000 instances. The evaluation results reflect high model performance throughout all patterns which appear with macro and weighted averages reaching 0.94.

Confusion matrix: The model presented in right side of figure 6 demonstrates excellent prediction accuracy through 3960 precise results for "DDoS_attacks_HoIC" and 3993 accurate outcomes for "SSH_Bruteforce." Misclassifications consist of 415 false positive predictions where "DDoS_attacks_LOIC_HTTP" appeared as "DDoS_attacks_GoldenEye" and 3880 false negative predictions where "DDoS_attacks_LOIC_HTTP" was identified instead as "SSH_Bruteforce".

Accuracy Plot: Figure 7 reveals training accuracy began with approximately 0.92 before reaching 0.95 after training and demonstrated effective learning through convergence. When validation accuracy (green line) begins to fluctuate before stabilizing at approximately 0.90 the model demonstrates a level of overfitting but shows poor performance on unseen data.

Loss Plot: The training loss profile displayed in Figure 8 demonstrates effective learning as it begins high before spiking to reach 2.5 then dropping below 1 before achieving stabilization. Generalization quality remains high because validation error maintains low and stable levels. Since both training and validation losses show a point of convergence, I observe neither overfitting nor underfitting in the model behavior.

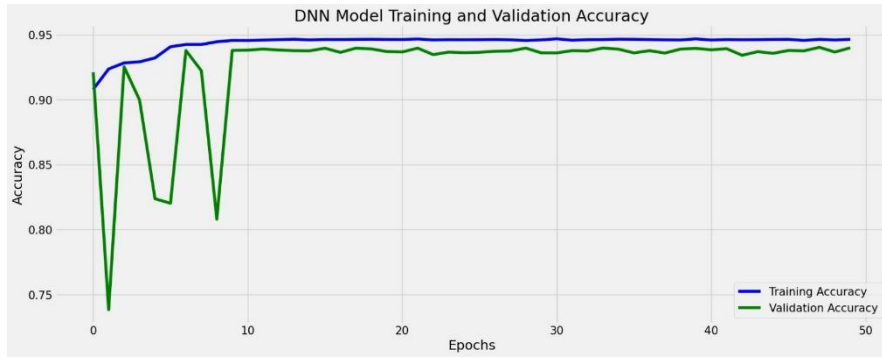


Figure 7 : Accuracy Plot

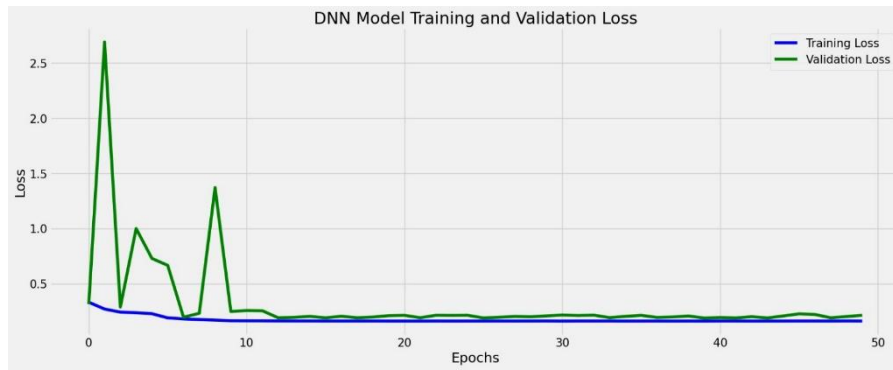


Figure 8 : Accuracy Plot

Baseline Metrics: Baseline performance standards were computed for modeling comparison purposes. The binary classification baseline measured performance through random guessing which proved capable of approximately 50% accuracy because of the binary task structure. The baseline for the multiclass classification resulted from class label distribution proportions which led the predominant category to produce a roughly 33% baseline accuracy rate. The improved scores for the KNN Binary Classifier which stands at 92.79% and for the DNN Multiclass Classifier sitting at 94.19% demonstrate the trained models' successful performance against baseline measures.

6.2 Inference

Two classifiers manage input samples detection such that the Knn binary classifier identifies attacks and the DNN multiclass classifier determines the attack type. The security system outputs "BENIGN" when the attack detection fails to find threats but identifies the attack category if one exists. The following flowchart illustrates when the inference steps will occur which is complemented by code samples in Figure 11.

Load Models and Feature Scalers: The system loads both pre-trained binary and multiclass models together with dedicated scalers for the binary detection of benign or malicious data and the multiclass distinction between negative attack types. Feature selectors work with scalers to pick appropriate features which they simultaneously normalize.

Define Class Labels: The outputs from the model correlate to easy-to-understand textual descriptions which we refer to as class labels. The binary classifier assigns "BENIGN" or

"Malicious" as classes whereas the multiclass classifier provides attack type details for malicious classifications.

Prediction Function: The prediction function transforms input data through one hot encoding together with normalization. Within the system binary classification selects outputs "BENIGN" and "Malicious". Users get an identified attack type for malicious traffic through the multiclass model's output. The prediction algorithm shows the most probable label as its result details.

Explanation of Output: A user provides CSV files like (Benign_test (2).csv) to the prediction function for classification while displaying results. A benign case equals authentic requests as indicated in Figure 10. When a network activity is categorized as malicious within Figure 9 it shows the exact attack type such as "DDOS_attack_HOIC".

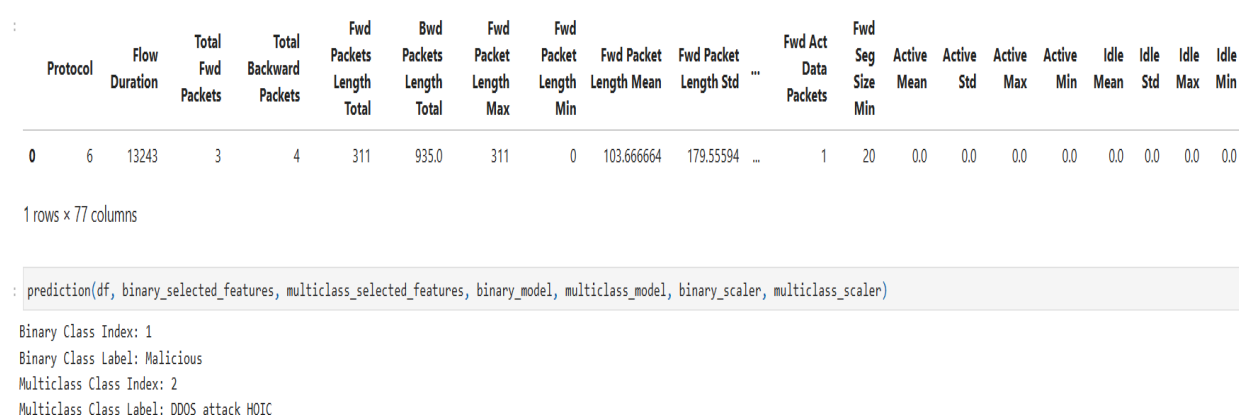


Figure 9: Output for Malicious

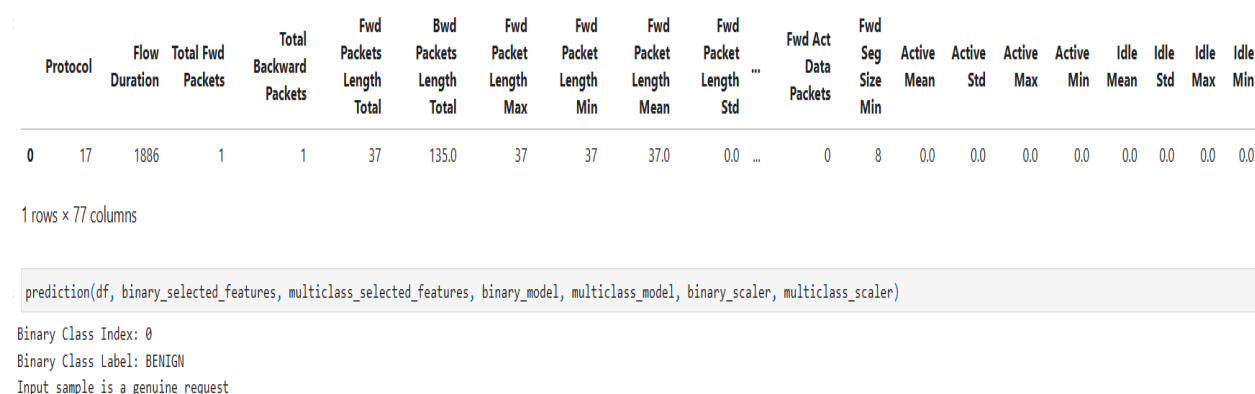


Figure 10: Output for Benign

```

import warnings
warnings.filterwarnings("ignore")

import pandas as pd
import numpy as np
from tensorflow.keras.models import load_model
import pickle
import os

with open("models/KNeighborsClassifier.pkl", "rb") as file:
    binary_model = pickle.load(file)

multiclass_model = load_model("models/DeepNeuralNetwork_model.h5", compile=False)

with open(file="models/binary_selected_features.pkl", mode='rb') as file:
    binary_selected_features = pickle.load(file=file)

with open(file="models/multiclass_selected_features.pkl", mode='rb') as file:
    multiclass_selected_features = pickle.load(file=file)

with open(file="models/binary_scaler.pkl", mode='rb') as file:
    binary_scaler = pickle.load(file=file)

with open(file="models/multiclass_scaler.pkl", mode='rb') as file:
    multiclass_scaler = pickle.load(file=file)

binary_class_labels = ['BENIGN', 'Malicious']

multiclass_class_labels = ['DoS_attacks_GoldenEye', 'Bot', 'DDoS_attack_HOIC', 'DoS_attacks_Hulk', 'SSH_Bruteforce',
                           'Infiltration', 'DDoS_attacks_LOIC_HTTP']

def prediction(input_data, b_features, m_features, b_model, m_model, b_scaler, m_scaler):
    # One-hot encode categorical features
    input_data = pd.get_dummies(input_data)

    # Normalize the data for both binary and multiclass models
    input_data_binary = input_data[b_features]
    input_data_multiclass = input_data[m_features]

    # Apply binary scaler
    input_data_binary_scaled = b_scaler.transform(input_data_binary.values)
    input_data_binary = pd.DataFrame(input_data_binary_scaled, columns=input_data_binary.columns)

    # Predict binary class
    binary_prediction = b_model.predict(input_data_binary.values)

    # Check if the prediction is 1D or 2D
    if binary_prediction.ndim == 1:
        binary_index = binary_prediction[0]
    else:
        binary_index = np.argmax(binary_prediction, axis=1)[0]

    binary_label = binary_class_labels[binary_index]

    print(f"Binary Class Index: {binary_index}")
    print(f"Binary Class Label: {binary_label}")

    if binary_label != "BENIGN":
        # Apply multiclass scaler
        input_data_multiclass_scaled = m_scaler.transform(input_data_multiclass.values)
        input_data_multiclass = pd.DataFrame(input_data_multiclass_scaled, columns=input_data_multiclass.columns)

        # Predict multiclass label
        multiclass_prediction = m_model.predict(input_data_multiclass.values)

        # Check if the prediction is 1D or 2D
        if multiclass_prediction.ndim == 1:
            multiclass_index = multiclass_prediction[0]
        else:
            multiclass_index = np.argmax(multiclass_prediction, axis=1)[0]

        multiclass_label = multiclass_class_labels[multiclass_index]

        print(f"Multiclass Class Index: {multiclass_index}")
        print(f"Multiclass Class Label: {multiclass_label}")
    else:
        print(f"Input sample is a genuine request")

user_input_filepath = "user_input/DDoS attack-HOIC_test (1).csv"

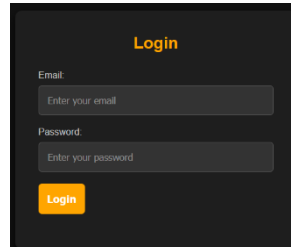
df = pd.read_csv(user_input_filepath)
df.head()

```

Figure 11: Code Snippets for Inference

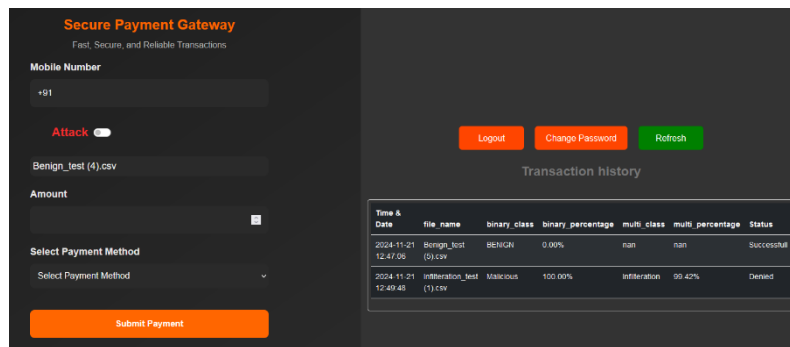
6.3 Front-End Interaction

User authentication occurs through email address alongside their password. Access to further services becomes available with correct credential entry while incorrect entries cause an error message according to Figure 11.



The login page features a dark background with the title 'Login' in orange. It includes two input fields: 'Email' with the placeholder 'Enter your email' and 'Password' with the placeholder 'Enter your password'. A yellow 'Login' button is positioned below the password field.

Figure 12: Login page

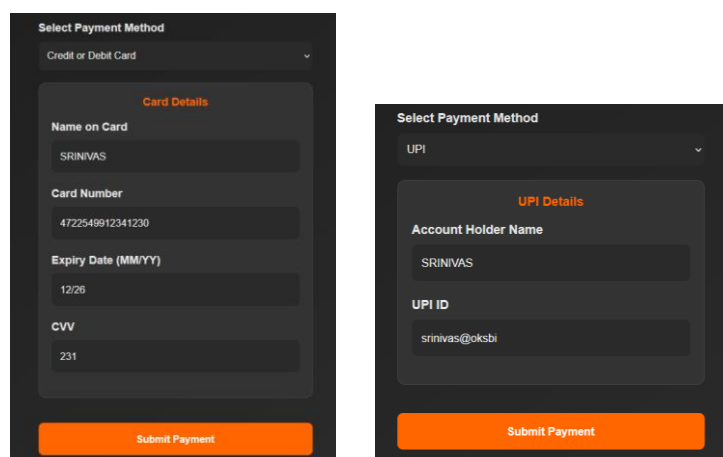


The Payment Gateway page is split into two panels. The left panel, titled 'Secure Payment Gateway' with the tagline 'Fast, Secure, and Reliable Transactions', contains a 'Mobile Number' field with '+91' as a prefix, an 'Attack' toggle switch, a file upload field for 'Benign_test (4).csv', an 'Amount' field, a 'Select Payment Method' dropdown, and a 'Submit Payment' button. The right panel includes 'Logout', 'Change Password', and 'Refresh' buttons, followed by a 'Transaction history' section with a table.

Time & Date	file_name	binary_class	binary_percentage	multi_class	multi_percentage	Status
2024-11-21 12:47:05	Benign_test (4).csv	BENIGN	0.00%	nan	nan	Successful
2024-11-21 12:49:45	infiltration_test (1).csv	Malicious	100.00%	Infiltration	99.42%	Denied

Figure 13: Payment Gateway page

Secure transaction capability resides in Figure 12 left while its right side offers transaction details plus account management features such as logout and password alterations.



Two side-by-side panels show payment method details. The left panel, titled 'Select Payment Method' with a dropdown set to 'Credit or Debit Card', shows 'Card Details' with fields for 'Name on Card' (SRINIVAS), 'Card Number' (472549912341230), 'Expiry Date (MM/YY)' (12/26), and 'CVV' (231), followed by a 'Submit Payment' button. The right panel, also titled 'Select Payment Method' with a dropdown set to 'UPI', shows 'UPI Details' with fields for 'Account Holder Name' (SRINIVAS) and 'UPI ID' (srinivas@oksbi), followed by a 'Submit Payment' button.

Figure 14: Payment Gateway methods

The payment gateway on figure 13 presents payment method options that users can choose between card payment and UPI option. The authorization system will verify all credentials before completing any transaction process.

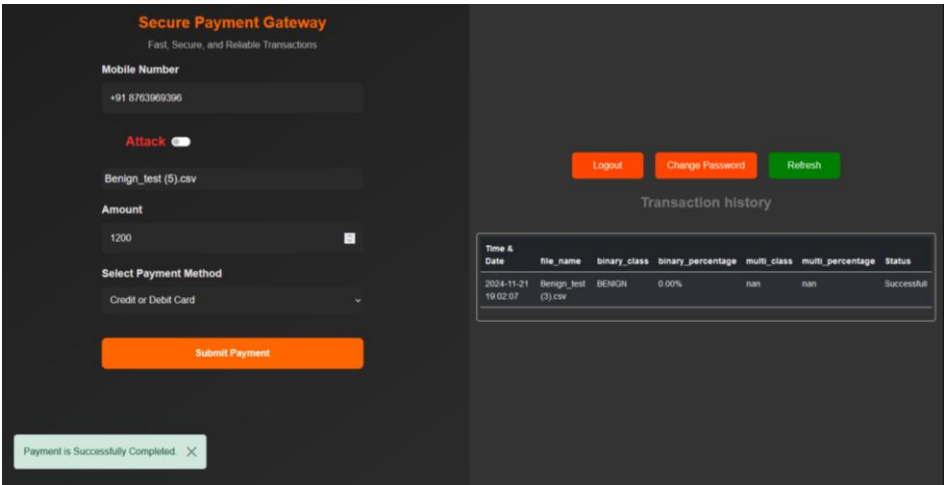


Figure 15:Successful Payment

When a payment is processed the transaction enters a dual-layer detection system. Binary classification models work fast to determine traffic either benign or malicious. Figure 14 demonstrates transaction success because of benign detection.

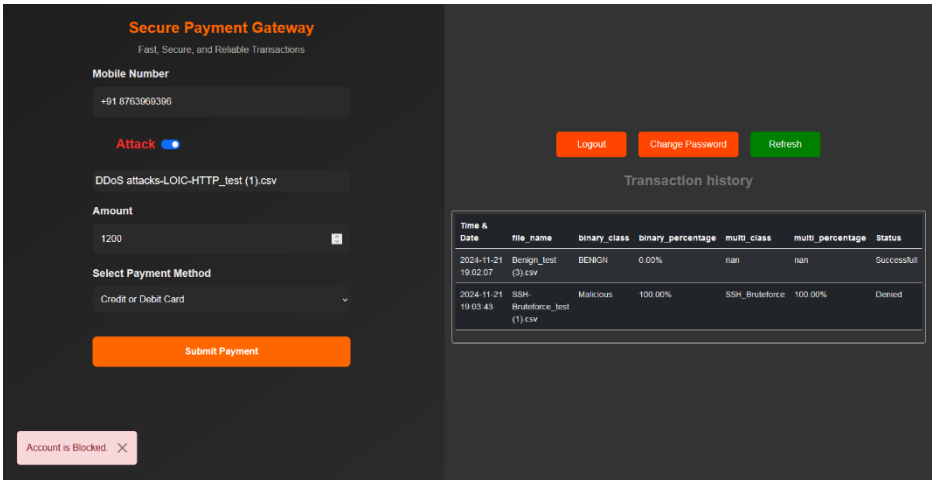


Figure 16:Denial of Payment

A binary classification model identifies benign transactions with malicious traffic first, then a multiclass detection system processes the threat along with its type before blocking requests as referenced by Figure 15 and flags the account for no further transactions pending resolution.

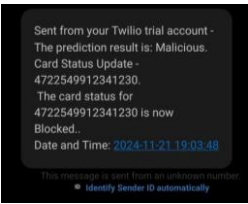


Figure 17:SMS Notification

Users receive through text messaging details of the suspicious attack and reason behind their account blockade features found in Figure 16. Through this step users remain informed and protected against potential threats.

6.4 Discussion

Binary classifier demonstrated superior performance for separating benign network traffic from malicious payloads as its resulting accuracy and precision metrics along with recall rating confirmed effectiveness. The evaluation of malicious traffic detection methods has revealed a vulnerability to false negatives that demand improvement. Our higher accuracy Multiclass model classifier managed to identify numerous cyberattacks including DoS and DDoS attacks but suffered from sporadic misclassification due to class imbalance or insufficient training data for rare attack types. The performance issues this system demonstrates can be effectively addressed through oversampling data and conducting fine-tuning architecture. While the DNN demonstrated strong performance in both learning and generalization its early-stage results exhibited slight overfitting due to the lack of regularization. While the integrated KNN-DNN inference system effectively accurately identifies benign and malicious network traffic together with attack subtype classification existing performance optimization opportunities can advance system real-time capability. Users receive a secure and intuitive front-end interface experience through display of attack type and class probability together with transaction status information which ensures transparency and builds trust. User progress updates during threat-based transactions work to build greater system trustfulness.

7 Conclusion and Future Work

The research evaluates a two-level Intrusion Detection System for detecting cyber threats in the financial industry while testing K-Nearest Neighbors for binary classification alongside Deep Neural Networks for multiclass classification. This research developed an IDS to process traffic distinguishing benign activities from harmful streams identify different assault patterns and alert users when threats emerge. The project included steps of data preprocessing and model training together with creating a user-friendly front-end interface to provide real-time threat detection. The proposed IDS reached 92.79% accuracy with the binary model and achieved 94.19% accuracy with the multiclass model confirming its ability to detect multiple cyber threats. The proposed models demonstrate strong performance yet show deficiencies through false positive and false negative results which appear more frequently with certain attack varieties making them imperfect solutions. The research makes a constructive impact on financial technologies' protection by delivering a functional IDS that delivers outstanding performance even with limited specific attack detection capabilities and class imbalances.

Subsequent works will target research constraints by enhancing detection capabilities for different attack types coupled with improved class imbalance management. We managed to lower false positive rates together with false negative rates through the application of sophisticated ensemble strategies. The detection capabilities for unknown attack patterns

would strengthen if models incorporated diverse data features and real time network traffic information. The achievement of improved system accuracy and adaptability emerges through hybrid model research which combines machine learning strengths with conventional rule-based methods. The solution which has been developed presents an opportunity for businesses to commercialize a fully deployable real-time cyber threat monitoring system for financial institutions. Future research should investigate the application of the IDS within cloud computing contexts because this opens new opportunities for both scalability optimization and performance enhancement.

Reference

Awajan, A., 2023. A novel deep learning-based intrusion detection system for IOT networks. *Computers*, 12(2), p.34.

Azizan, A.H., Mostafa, S.A., Mustapha, A., Foozy, C.F.M., Wahab, M.H.A., Mohammed, M.A. and Khalaf, B.A., 2021. A machine learning approach for improving the performance of network intrusion detection systems. *Annals of Emerging Technologies in Computing (AETiC)*, 5(5), pp.201-208.

Balamurugan, E., Mehbodniya, A., Kariri, E., Yadav, K., Kumar, A. and Haq, M.A., 2022. Network optimization using defender system in cloud computing security-based intrusion detection system with game theory deep neural network (IDSGT-DNN). *Pattern recognition letters*, 156, pp.142-151.

Balyan, A.K., Ahuja, S., Lilhore, U.K., Sharma, S.K., Manoharan, P., Algarni, A.D., Elmannai, H. and Raahemifar, K., 2022. A hybrid intrusion detection model using ega-pso and improved random forest method. *Sensors*, 22(16), p.5986.

Chang, V., Golightly, L., Modesti, P., Xu, Q.A., Doan, L.M.T., Hall, K., Boddu, S. and Kobusińska, A., 2022. A survey on intrusion detection systems for fog and cloud computing. *Future Internet*, 14(3), p.89.

Dietz, K., Mühlhauser, M., Kögel, J., Schwinger, S., Sichermann, M., Seufert, M., Herrmann, D. and Hoßfeld, T., 2024. The Missing Link in Network Intrusion Detection: Taking AI/ML Research Efforts to Users. *IEEE Access*.

Fares, O.H., Butt, I. and Lee, S.H.M., 2022. Utilization of artificial intelligence in the banking sector: A systematic literature review. *Journal of Financial Services Marketing*, p.1.

Farhan, B.I. and Jasim, A.D., 2022. Performance analysis of intrusion detection for deep learning model based on CSE-CIC-IDS2018 dataset. *Indonesian Journal of Electrical Engineering and Computer Science*, 26(2), pp.1165-1172.

Gassais, R., Ezzati-Jivan, N., Fernandez, J.M., Aloise, D. and Dagenais, M.R., 2020. Multi-level host-based intrusion detection system for Internet of things. *Journal of Cloud Computing*, 9(1), p.62.

Ghanem, W.A.H., Jantan, A., Ghaleb, S.A.A. and Nasser, A.B., 2020. An efficient intrusion detection model based on hybridization of artificial bee colony and dragonfly algorithms for training multilayer perceptrons. *IEEE Access*, 8, pp.130452-130475.

Islam, U., Muhammad, A., Mansoor, R., Hossain, M.S., Ahmad, I., Eldin, E.T., Khan, J.A., Rehman, A.U. and Shafiq, M., 2022. Detection of distributed denial of service (DDoS) attacks in IOT based monitoring system of banking sector using machine learning models. *Sustainability*, 14(14), p.8374.

Jumabek, A., Yang, S. and Noh, Y., 2021. CatBoost-based network intrusion detection on imbalanced CIC-IDS-2018 dataset. *한국통신학회논문지*, 46(12), pp.2191-2197.

Kim, J., Kim, J., Kim, H., Shim, M. and Choi, E., 2020. CNN-based network intrusion detection against denial-of-service attacks. *Electronics*, 9(6), p.916.

Latah, M. and Toker, L., 2020. An efficient flow-based multi-level hybrid intrusion detection system for software-defined networks. *CCF Transactions on Networking*, 3(3), pp.261-271.

Li X, Chen W, Zhang Q, Wu L. Building auto-encoder intrusion detection system based on random forest feature selection. *Computers & Security*. 2020 Aug 1;95:101851.

Maseer, Z.K., Yusof, R., Bahaman, N., Mostafa, S.A. and Foozy, C.F.M., 2021. Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. *IEEE access*, 9, pp.22351-22370.

Mayuranathan, M., Saravanan, S.K., Muthusenthil, B. and Samydurai, A., 2022. An efficient optimal security system for intrusion detection in cloud computing environment using hybrid deep learning technique. *Advances in Engineering Software*, 173, p.103236.

Milosevic, M.S. and Ciric, V.M., 2022. Extreme minority class detection in imbalanced data for network intrusion. *Computers & Security*, 123, p.102940.

Pelletier, Z. and Abualkibash, M., 2020. Evaluating the CIC IDS-2017 dataset using machine learning methods and creating multiple predictive models in the statistical computing language R. *Science*, 5(2), pp.187-191.

Stiawan, D., Idris, M.Y.B., Bamhdi, A.M. and Budiarto, R., 2020. CICIDS-2017 dataset feature analysis with information gain for anomaly detection. *IEEE Access*, 8, pp.132911-132921.

Süzen, A.A., 2021. Developing a multi-level intrusion detection system using hybrid-DBN. *Journal of Ambient Intelligence and Humanized Computing*, 12(2), pp.1913-1923.

Tay, L.Y., Tai, H.T. and Tan, G.S., 2022. Digital financial inclusion: A gateway to sustainable development. *Heliyon*, 8(6).

Thakkar, A. and Lohiya, R., 2020. A review of the advancement in intrusion detection datasets. *Procedia Computer Science*, 167, pp.636-645.

Ullah, S., Khan, M.A., Ahmad, J., Jamal, S.S., e Huma, Z., Hassan, M.T., Pitropakis, N., Arshad and Buchanan, W.J., 2022. HDL-IDS: a hybrid deep learning architecture for intrusion detection in the Internet of Vehicles. *Sensors*, 22(4), p.1340.

Zhang, C., Jia, D., Wang, L., Wang, W., Liu, F. and Yang, A., 2022. Comparative research on network intrusion detection methods based on machine learning. *Computers & Security*, 121, p.102861.