National College of Ireland

Safeguarding Financial Transactions: A Machine Learning Perspective on online payment network security

MSc Research Project
MSc Cybersecurity

Shiron Shine Yesudas
Student ID: 22175504

School of Computing
National College of Ireland

Supervisor: MICHAEL PANTRIDGE

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	SHIRON SHINE YESUDAS
Student ID:	22175504
Programme:	MASTER'S OF SCIENCE IN CYBERSECURITY **Year:** 2024
Module:	MSC RESEARCH PROJECT
Supervisor:	MICHAEL PANTRIDGE
Submission Due Date:	25/04/2024
Project Title:	SAFEGUARDING FINANCIAL TRANSACTIONS: A MACHINE LEARNING PERSPECTIVE ON ONLINE PAYMENT NETWORK SECURITY
Word Count: 7622	**Page Count 20**

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: SHIRON SHINE YESUDAS

Date: 25/04/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

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

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

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

Abstract

The objective of this study is to describe the development of an advanced Intrusion Detection System (IDS) specifically meant for securing online payment transactions. The proposed work seeks to employthe most advanced modern machine learning techniques and comprehensively analyze threats. This system is aimed at real-time intrusion detection and prevention. In that regard, we will collect a lot of information using CICIDS2017 dataset and thereafter apply different machine learning algorithms such as Support Vector Classifier (SVC), Convolutional Gated Recurrent Unit (CGRU), Artificial Neural Network (ANN) for fraud detection. Furthermore, the suggestion has in mind making it user-friendly so that people can use it without any glitches. The overall goal of this system is improved security in online payments through AI-based intrusion detection which helps in mitigating risks and countering fraudulent activities. This research assesses three various machine learning techniques on their efficiency to detect fraudulent transactions. CGRU had the highest accuracy rate among them all with almost perfect 99.61%. SVC and ANN recorded slightly lower levels of accuracy at 94.59% and 97.97% respectively as compared to CGRU's calculations accuracy rate value, thus implying its superior performance in accurately detecting fraudulent deals than others put together for analysis purposes or considerations made by these algorithms. These two findings underscore the importance of complex machine learning approaches to ensure reliability and security in digital financial transactions by providing a strong defense against frauds too.

Keywords: *Machine Learning, Intrusion Detection System, Online Payment Transactions, Support Vector Classifier (SVC), Convolutional Gated Recurrent Unit (CGRU), Artificial Neural Network ANN)*

1. Introduction

Online transactions have become widespread in this era of digital technology, transforming the way people and companies deal with monetary transactions. Digital payments are convenient but also come with new problems especially on security issues. The number of cyber threats such as fraud and intrusion attempts is growing at a significant rate, which poses serious risks to the trustworthiness of online payment networks. Consequently, stringent security measures are necessary to safeguard users and institutions. Consequently, it is recommended that this study suggests creating an advanced Intrusion Detection System (IDS) specifically for protecting the online payment system.

This research seeks to design a system capable of detecting and preventing intrusions in real-time within online payment networks through machine learning and comprehensive threat analysis. In doing so, the study proposes that the dataset, which contains vast amounts of cyber security data including scenarios surrounding online payments can be utilized. This dataset will help train and test the IDS by providing insights into fraudulent behavior patterns.

The models employed in this approach include Support Vector Classifier (SVC), Convolutional Gated Recurrent Unit (CGRU), Artificial Neural Network (ANN) for fraud detection in online payments. These particular models have been chosen because they can handle complex data well while also effectively detecting anomalies indicative of frauds taking place. Besides developing sophisticated detection algorithms, this study also emphasizes on creation of user-friendly interface that makes interaction with IDS

straightforward. Thus, stakeholders can effectively monitor and respond to security events as appropriate.

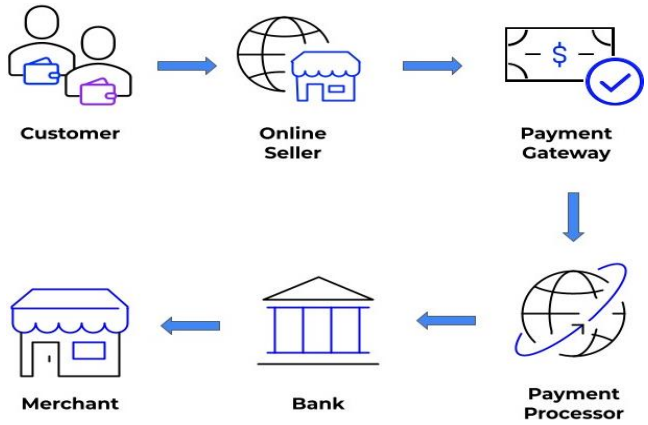

Fig 1: Online Payment Network

1.1 Background Information

Cyber threats are targeting financial transactions as cyber security is developing. With online payment systems becoming more common, adequate security measures must be put in place to prevent fraud and breaches. Transaction confidentiality is at risk due to problems such as unauthorized access and data breaches. Such threats are not effectively countered by traditional security approaches most of the time. This research has been motivated by successes in cyber-security from machine learning with an aim on designing a unique intrusion detection system for payment security. The objective is to come up with an advanced algorithm that can rapidly detect and stop abnormalities on online payment systems.

1.2 Research Problem

The emergence of online payment systems has brought us convenience but also exposed us to hacking and fraud. Our focus, therefore, lies in building an intelligent system using machine learning. This system is meant to prevent unauthorized access as well as fraud in online transactions. Our major target is the development of a good model capable of distinguishing genuine transactions from suspected ones accurately. By this, we hope to improve the safety of digital financial interactions thereby ensuring more secure internet payments for all parties involved.

2. Related Work

This study centers around online payment systems and delves into the evaluation, comparison, and contrast of existing literature on intrusion detection methods based on machine learning. It meticulously scrutinizes each piece of literature, highlighting their main claims, goals, and impacts in relation to the main question. Goal of analyzing this literature is to pinpoint main area of the study and outline how this study will contribute to the scholarly field.

Machine Learning Applications in Security Frameworks

Researchers have recently made significant progress in addressing electronic payment systems, mobile transactions. In the Machine Learning-Assisted Secure Mobile Electronic Payment Framework (ML-SMEPF) by Wang et al., 2021. This framework uses multiple approaches including Efficient Random Oracle Model to detect malware and multi-factor authentication and fraud detection for assessing security of mobile network payment gateways. Different types of frauds can also be detected through the Mutual Mobile Authentication model that they included. From an extensive simulation analysis, Wang et al. confirmed that this framework is good in terms of accuracy, security performance and cost-

effectiveness. They highlighted their flexibility, usability and safety of use with respect to mobile money payments. A similar study was conducted by Hajek et al., 2023, which addressed the problem of detecting fraudulent activities in mobile payment systems using a framework. The latter encompasses unsupervised outlier detection for dealing with data scarcity and class imbalance issues inherent in financial fraud datasets. Their semi-supervised ensemble model has overperformed existing methods; thus putting emphasis on its financial implications for organizations as well as offering practical solutions for tackling fraud transactions.They have contributed to enhancing EPS' security by suggesting a novel method for testing software during its development life cycle (Behera et al., 2021). Protection against attacks, prioritization of test cases, project management; testing tools; metrics are among those various safeguards aimed at personal information protection & user confidence improvement across multiple payment channels that were put into place in his approach.Their goal is to ensure a reduction in duration, costs and efforts while eventually increasing online sales within the payments industry as soon as possible.Security vulnerabilities need to be discovered at early stages because Behera et al., seek to achieve higher internet sales volumes by reducing duration, cost effort that are spent online.In these articles combined together it is shown how machine learning techniques such Support Vector Machines (SVM) when coupled with strong security measures can make electronic payment systems more reliable and safe for mobile banking transactions. Also, they underscore the importance of intrusion detection and prevention in maintaining network security.

Network Intrusion Detection Systems using Machine Learning

In the recent past, there have been remarkable changes happening in the area of network security through machine learning as can be seen from some significant studies. Machine learning methods were used by (Almutairi et al., 2022) to investigate Network Intrusion Detection Systems (NIDS) and focused on NSL-KDD dataset. According to their examination of various algorithms such as Support Vector Machine, J48, Random Forest and Naïve Bayes – Random Forest was seen to be superior to others based on recall, accuracy and Matthews correlation coefficient which surpasses current intrusion detection systems. Despite challenges such as imbalanced class, synthetic malicious traffic, etc., these methods had potential in detecting network intrusions which demonstrated the effectiveness of machine learning for security enhancement in communication networks. The AB-TRAP framework is an example of a complete solution for NIDS this utilize machine learning that was introduced by (Bertoli et al., 2021). It was intentionally designed to overcome the limitations inherent in existing datasets regarding new attack types and outdated ones thereby involving several stages: Dataset generation; Model training; Performance evaluation; Deployment. The best f1-scores came from AB-TRAP against port scanning attacks (TCP) at internet environments, LANs with minimal resources consumption making it the right choice for developing protection modules (NIDS) under dynamic network conditions. Data quality became even more important regarding intrusion detection systems based on machine learning in Cybersecurity after (Tran et al., 2022). In their study they showed that pretrained models perform much better than traditional ones because of less vulnerability towards problems including duplication or overlapped data within them. They tried out their proposed cleaning method using 11 datasets to prove its effectiveness on improving models' performance prompting them to suggest that creating a data curation framework needs careful consideration so that high-quality intrusion detection systems can be built. Finally, Lirim Ashiku et al.'s (2021) work evolved around the development of an adaptive and resilient network intrusion detection system (IDS) using deep learning architectures. This improvement for multiclass models involved recommended machine learning categorization

architecture as well as tractor-trailer max parameter tuning using the UNSW-NB15 dataset that covers both artificially generated attack scenarios and real-world network communication behavior. Their proposed approaches achieved 95.4% overall accuracy levels for Warburtons against a user-defined classification of 95.6%, signifying a significant step towards efficient recognition and classification of network threats from the standpoint of machine learning in boosting network security.

Advanced Analytics and Fraud Detection in Financial Systems

In 2023, two very interesting academic papers in the area of financial systems can give us a lot of insight on important security matters and analytics. More significantly, (Patel et al., 2023) elaborates on the role of data analytics in credit card business focusing majorly on identifying frauds, evaluating risks and managing information. It argues for adoption of advances in machine learning models and asserts that strong data analytics frameworks are significant for proper risk assessment. Further, it explores how to incorporate advanced technologies like augmented reality or quantum computing into data analytics and their potential uses in credit card analysis as technology crosses path with finance. Secondly, (Sangeetha 2023) deals with increasing cybersecurity threats faced by banks as well as other financial institutions proposing a comprehensive cyber security framework. This involves taking proactive measures such as use of firewalls, monitoring tools and training employees besides reactive strategies as incident response plans and digital forensics. They also recommend that the current state of affairs requires one to have an updated knowledge about cyber security trends so they will be able to protect their accounts and data from more complex forms of cyber attacks thereby showing why constant change is necessary to continuously keep guard over the safety of our financial systems' securities.

Deep Learning and Intrusion Detection Systems (IDS)

The field of cybersecurity is evolving, and three academic papers demonstrate this transformation in the field of intrusion detection systems (IDS). Lansky (Lansky, et al.2021) focuses on how deep learning can be applied to IDS in order to increase network security against more sophisticated online threats. This paper accomplishes this objective by carefully classifying and analyzing several deep learning based IDS options; it explores their use of deep neural networks for detecting intrusions, discussing their benefits and shortcomings as well as claiming them. (Halbouni, et al.2022) contribute to this discussion by presenting a comprehensive review of machine learning and deep learning applications in cyber security with an emphasis on IDS. The study underscores the importance these technologies play in making information secure from ever changing cyber-attacks across different network implementations, methods of training, algorithms used among others. Thus, they underscore the IRA (Importance Rate Analysis) IDS relevance in the cybersecurity landscape accentuating its significance towards dataset selection while at the same time highlighting transition from classical M.L approaches to D.L ones for better precision and efficiency in addressing intricate cyber security challenges. In keeping with these discussions, (Shafique, et al., 2022) provide a strong intrusion detection system that uses a multilayer perceptron (MLP) classifier which has an accuracy level of up to 99.98%. Through extensive testing across a variety of situations, their research ensures the trustworthiness and consistency of MLP classifier utilizing metrics such as Sensitivity or Specificity or Matthew's Correlation Coefficient validating its competence to separate various types of network behaviors. All these papers together show how intrusion detection system space is changing due to the inclusion of deep learning techniques coupled with improvements made on machine learning along with empirical validation that have been thorough thus enhancing modern IT environment's cybersecurity measures.

Research Niche and Expected Contribution

This research project has identified a large void in the existing literature about a proactive and complete Machine Learning based Intrusion Detection System (IDS) that can be specifically designed for the purpose of improving the security of online payment systems. Although research has moved ahead in this area, most of it has focused on various aspects of the problem without providing integrated solutions to this issue. Thus, this study aims at bridging that gap by providing a comprehensive technique which involves using machine learning algorithms such as Support Vector Classification (SVC), Artificial Neural Networks (ANN) and Convolutional Gated Recurrent Unit and implementing them through Scikit-learn and Tensorflow programming packages. This will enable us to evaluate the overall performance accuracy between these Machine Learning models hence enabling us to choose which will be more effective for ensuring secure online transaction payments. The broad objective of this study is to develop an advanced system that can provide secure payment measures for transactions conducted via the internet.

3 Methodology

The project aims to develop an advanced intrusion detection system for ensuring the security of online payment transactions. The methodology entails a thorough integration of machine learning techniques with a strong emphasis on analyzing potential threats. This approach is designed to create a robust and adaptive system capable of effectively detecting and thwarting intrusions in real-time. By combining machine learning with intrusion detection, the proposed system aims to enhance user confidence by providing a secure environment for conducting online financial transactions. The focus is on building a system that not only identifies existing threats but also adapts to new and evolving risks, thereby ensuring the safety and integrity of online payment networks.

Model Evaluation

This system design details the structured process of designing and executing a system for detecting credit card fraud. It covers the essential steps involved, starting from loading the data, preparing it through preprocessing techniques, visualizing relevant information, addressing class imbalance through oversampling methods, encoding categorical variables, and finally, standardizing the data through normalization techniques.

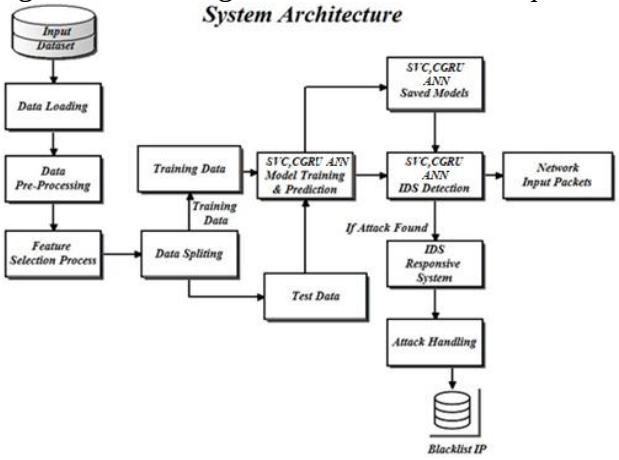

Figure 2: Model Evaluation Steps

The model evaluation illustrates a setup for a credit card fraud detection system that relies on machine learning.

1. Data Input: It deals with the historical data of network traffic transactions, which serves as the basis for training the fraud detection model.

2. Data Pre-processing: This phase involves preparing the historical transaction data for machine learning model training. It includes loading the data into the system and cleaning it up by handling missing values, outliers, and scaling.
3. Feature Selection Process: This step focuses on identifying the most relevant attributes from the pre-processed data, which will be used by the machine learning models to detect fraud.
4. Model Training and Prediction: This part involves dividing the pre-processed data into training and testing sets. The training set is used to train various machine learning models like ANN, SVC and CGRU to recognize fraudulent patterns in transactions. The trained models are then saved for later use.
5. IDS Detection and Input Packets: In this, the system monitors live credit card transactions for fraudulent activities using an Intrusion Detection System (IDS). It analyzes incoming transaction data in real-time.
6. Attack Handling: If the IDS system identifies a potentially fraudulent transaction, it triggers actions such as blocking the transaction, notifying the card issuer, and possibly adding the associated IP address to a blacklist to prevent future fraudulent activities.
In summary, the architecture outlines a system that employs machine learning to detect fraudulent credit card transactions. It trains models using historical data and applies them to live transactions, taking appropriate actions when fraud is suspected.

4 Design Specification

This section explores the detailed specifications of three important elements in machine learning: the Support Vector Classifier (SVC), the Convolutional Gated Recurrent Unit (CGRU), and Artificial Neural Networks (ANN). Furthermore, it emphasizes the importance of using the CICIDS2017 dataset for assessing intrusion detection systems.

Support Vector Classifier (SVC)

The Support Vector Classifier (SVC) is a widely used machine learning algorithm for classification tasks. It identifies a hyperplane to separate data classes, maximizing the margin between them to prevent overfitting. Key to its performance is the regularization parameter (C), balancing margin maximization and error minimization. SVC employs various kernels like linear, polynomial, RBF, and sigmoid to handle non-linear data by mapping it into higher dimensions. However, its efficacy hinges on proper kernel selection and parameter tuning. Despite this, SVC remains a powerful and adaptable classifier for diverse classification challenges in machine learning.

Convolutional Gated Recurrent Unit (CGRU)

The Convolutional Gated Recurrent Unit (CGRU) is a neural network that merges Convolutional Neural Networks (CNNs) and Gated Recurrent Units (GRUs), tailored for handling sequential data like time-series or spatial-temporal data. CGRU uses convolutional layers to efficiently extract features, capturing local patterns and spatial dependencies, then employs gated recurrent units to manage temporal dependencies across sequential inputs. These gates regulate information flow, determining what to retain or discard from past steps. By combining both convolutional and recurrent operations, CGRU maximizes advantages from both architectures, enabling parallel feature processing and capturing long-term dependencies. It's particularly useful for tasks like video processing, action recognition, and time-series forecasting.

Artificial Neural Networks (ANN)

Artificial Neural Networks (ANNs) mimic the human brain's structure, comprising interconnected nodes or "neurons" arranged in layers. Information flows from the input layer through hidden layers to the output layer, with weights on connections adjusting during

training. ANNs excel at discerning intricate patterns in data, finding applications in image recognition, language processing, and regression analysis. They can generalize from training to predict unseen data but demand significant computational resources and ample data for effective training. Despite complexity, ANNs drive modern machine learning, fueling advancements across diverse domains.

Data Collection

The CICIDS2017 dataset is a valuable resource for evaluating intrusion detection and prevention systems. It offers a wide range of network traffic data, covering both benign activities and various types of attacks commonly encountered in real-world scenarios. These attacks include Brute Force FTP, Brute Force SSH, Denial of Service (DoS), Distributed Denial of Service (DDoS), Heartbleed, Web Attacks, Infiltration, Botnet activity, and DDoS attacks. The dataset addresses the limitations of previous datasets by providing diverse and realistic traffic samples, comprehensive network configurations, labeled datasets, various interaction scenarios, thorough capture methods, and extensive metadata.

Data source: https://www.kaggle.com/datasets/cicdataset/cicids2017

Data Loading

The dataset being loaded, information extracted from the CSV file related to the CICIDS2017 dataset, which is widely known in the field of network intrusion detection. It consists of 24,689 rows and 79 columns. Each row represents a distinct observation or instance, while the columns denote various features or attributes associated with these observations

5 Implementation

This implementation section details the thorough steps involved in preparing and analyzing a dataset specifically for ensuring the security of online payment transactions. It covers various essential tasks such as cleaning up the data, creating visual representations, adjusting the data balance, converting categorical data into numerical format, choosing relevant features, standardizing the data, dividing it into subsets, and storing it for future use. Additionally, Chapter 7 introduces a prototype model implementation for further exploration.

Data Preprocessing

The dataset used to ensure the security of online payment transactions undergoes thorough preparation to make sure it's accurate and suitable for analysis and training models. This involves several important steps. First, the column names are standardized by removing any extra spaces, making everything consistent and avoiding problems later on. Then, we check for any missing values in the data to make sure it's complete.

Bar Chart with 'Label' Feature

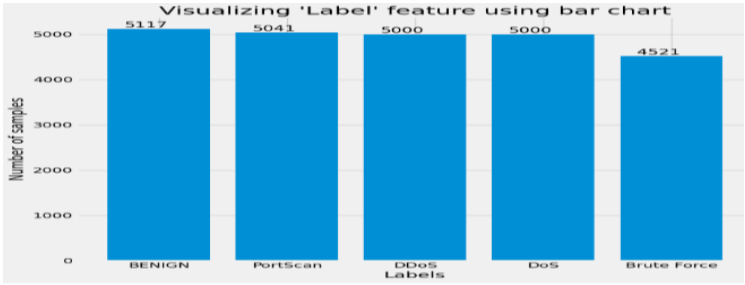

Fig 3: Visualizing 'label' feature using bar chart

The bar chart depicted in Figure 3 displays how samples in a dataset are distributed based on different attack types. On the x-axis, there are labels representing five specific fraud categories: benign, PortScan, DDOS, DoS, and Brute Force. The vertical axis y-axis shows the number of samples corresponding to each fraud type. According to the data presented, the

dataset consists of 5117 samples labeled as benign, making it the most prevalent category. Following closely behind is PortScan, with 5041 samples.

Pie Chart with 'Label' Feature

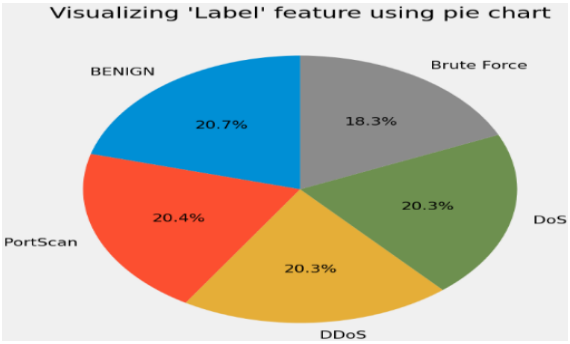

Fig 4: Visualizing 'label' feature using pie chart

The pie chart as in Fig 4 presents the distribution of the label feature within a dataset related to network related attacks, highlighting potential fraudulent activities. Benign transactions represent 20.7% of the dataset, likely indicating regular, non-fraudulent transactions. In contrast, PortScan and Brute Force transactions each constitute 18.3% and 20.4% respectively, suggesting potential attempts to breach systems. DoS and DDoS attacks collectively make up 40.6% of the dataset, with each accounting for 20.3%, indicating efforts to disrupt services by flooding servers with traffic. The class of Benign transactions performs relatively well, representing regular, non-fraudulent activities with a notable presence in the dataset.

Data Oversampling

Data Oversampling

```
oversample = SMOTE()

X_smote, y_smote = oversample.fit_resample(df.drop(labels='Label', axis=1), df['Label'].values)

df = pd.DataFrame(data=X_smote,columns=df.drop(labels='Label', axis=1).columns)
df['Label'] = y_smote
```

Fig 5: Over Sampling

In oversampling as in Fig 5 a key focus is addressing class imbalance, a common issue where one class (like fraudulent transactions) is significantly less represented compared to another class (like non-fraudulent transactions). To tackle this, the Synthetic Minority Over-sampling Technique (SMOTE) is utilized. This technique generates synthetic samples for the minority class, ensuring a more balanced distribution between classes. After applying SMOTE, the dataset is reconstructed with the oversampled features and corresponding labels. This process enhances the training of machine learning models, improving their ability to recognize patterns and make accurate predictions.

Bar Chart with 'Label' Feature after Data Balancing

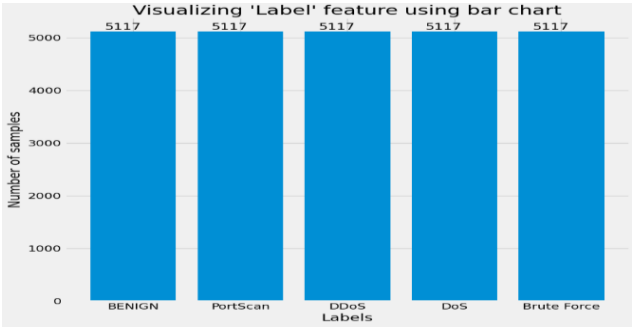

Fig 6: Visualizing 'label' feature using bar chart

The bar chart as in Fig 6 displays the distribution of samples across different classes, with each class containing 5117 samples. This suggests a balanced dataset where the number of data points for each class is equal. In the context of credit card fraud detection, this implies an absence of oversampling, a technique utilized to mitigate imbalanced datasets where one class significantly outnumbers the others. Oversampling typically involves generating additional data points for the minority class to address bias in machine learning algorithms.Pie Chart with ‘Label’ Feature after Data Balancing

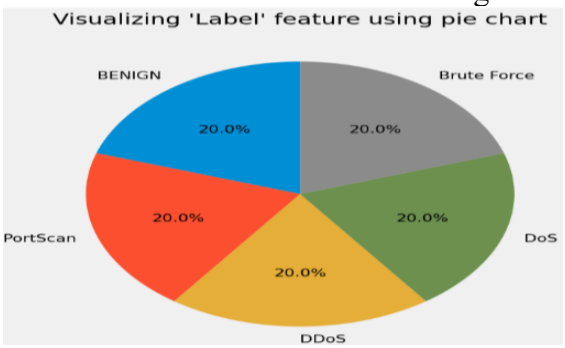

Fig 7: Visualizing ‘label’ feature using pie chart

The pie chart provided in Fig 7 depicts the distribution of the target variable, Label, within machine learning. The data demonstrates a balanced distribution, indicating roughly equal proportions between attacks, such as Brute Force, PortScan, DoS, and DDOS, and legitimate transactions categorized as BENIGN. This balance is critical for machine learning models as it helps mitigate bias towards the majority class. In datasets where one class predominates, models might incorrectly prioritize the majority class, leading to poor performance in identifying fraudulent transactions, which are typically less frequent.

Label Encoding

Label Encoding

```
class_dict={}
for idx, label in enumerate(class_labels):
    class_dict[label] = idx
print(class_dict)
```

```
{'BENIGN': 0, 'Brute Force': 1, 'DDoS': 2, 'DoS': 3, 'PortScan': 4}
```

Fig 8: Label Encoding

Label Encoding as in Fig 8 technique applied to a dataset, specifically a DataFrame containing a column named 'Label'. This dataset seems to comprise categorical data, with unique labels like 'BENIGN', 'Brute Force', 'DDoS', 'DoS', and 'PortScan'. The label encoding procedure involves converting these categorical labels into numerical values. Initially, the distinct labels are extracted from the DataFrame and sorted alphabetically. Then, a dictionary is created where each unique label is mapped to its corresponding index in the sorted list of labels. This dictionary effectively assigns a numerical value to each label. Once the encoding process is executed, the resulting dictionary offers a mapping from categorical labels to numerical values, enabling the representation of categorical data in a numerical format suitable for utilization in machine learning algorithms.

Feature Selection

Feature Selection

```
target_feature = 'Label'
all_features = df.columns.tolist()
all_features.remove(target_feature)
corr = df[all_features].corrwith(df[target_feature])
```

```
corr_df = pd.DataFrame(corr).reset_index()
corr_df.columns = ['Features', 'Importance']
corr_df.head(10)
```

Fig 9: Feature Selection

Feature selection as in Fig 9 based on correlation with a target feature labeled 'Label'. It begins by calculating the correlation coefficients between each feature in a DataFrame and the target feature. After excluding the target feature from the list of all features, it constructs a DataFrame to store the correlation results. Each row in this DataFrame represents a feature, along with its correlation coefficient with the target feature. This methodology allows for the identification of features that demonstrate stronger correlations with the target, aiding in feature selection for predictive modeling. By examining the top correlated features, one can discern which attributes might have the most impact on predicting or explaining the target variable, thus streamlining the model-building process.

Feature Importance

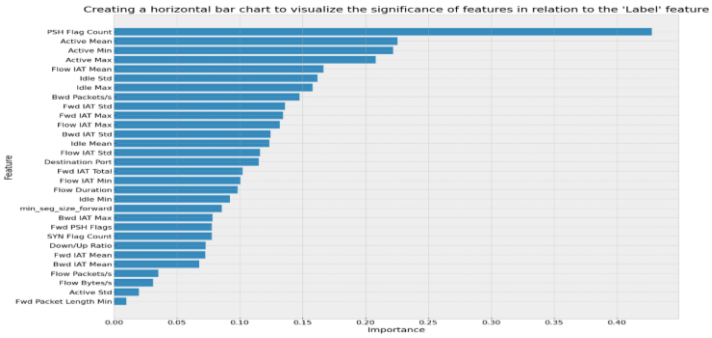

Fig 10: Feature Importance

Feature importance as in Fig 10 process using a horizontal bar chart to visualize the importance of features with respect to a specified label feature. Feature selection holds significant importance in machine learning, as it helps in identifying the most relevant features from a dataset, thereby improving model performance and reducing the risk of overfitting. In the context of credit card fraud detection, such selection can highlight key factors contributing to fraudulent activities, aiding in the development of more accurate fraud detection models. Notable features depicted in the chart include metrics like PSH Flag Count, Active Mean, Flow IAT Mean, among others, with each feature's importance represented by the length of its corresponding bar.

Data Normalization

Data Normalization

```
scaler = MinMaxScaler()
scaler = scaler.fit(X.values)
scaled_X = scaler.transform(X.values)

df = pd.DataFrame(data=scaled_X,columns=X.columns)
df['Label'] = y.values.ravel()
df.head()
```

Fig 11: Data Normalization

Data Normalization as shown in Fig 11 is a crucial preprocessing step. It ensures that numeric features associated with each payment are scaled to a standard range, typically between 0 and 1, to address potential discrepancies in feature scales. This process becomes particularly

important when dealing with diverse features as it enhances the performance of machine learning algorithms by making the features comparable. The `MinMaxScaler` method is used to calculate the minimum and maximum values of the features, which are then utilized to scale the data accordingly. Once normalized, the dataset is structured into a DataFrame where each column represents a feature from the original dataset, including labels indicating the payment status. This preparatory step establishes the groundwork for subsequent analysis or model training, ensuring that the data is appropriately scaled and organized for optimal performance.

Data Splitting

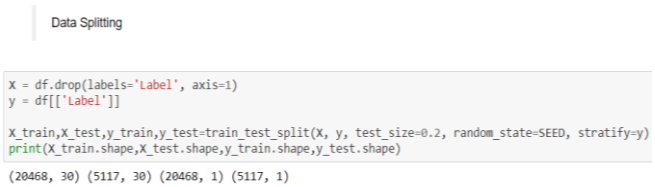

Fig 12: Data splitting

The dataset appears to encompass details concerning network attacks payments, likely encompassing various features as in Fig 12. It's evident that the data has been divided into training and testing sets, with 80% of the data designated for training and 20% for testing. Furthermore, it seems that the data splitting process has been executed while maintaining the class distribution of the target variable, which is essential in scenarios involving imbalanced datasets. The training set comprises 20468 samples, each containing 30 features, while the testing set includes 5117 samples, with the same number of features. Such meticulous data preparation serves as the groundwork for constructing and evaluating predictive models aimed at identifying fraudulent credit card transactions.

Save Data

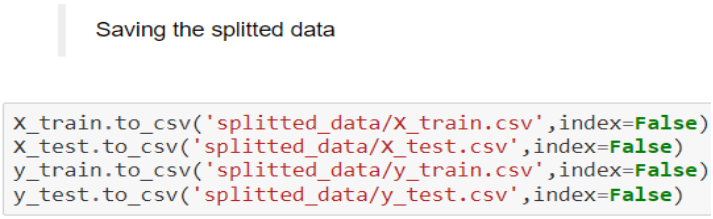

Fig 13: Save Data

As in Fig 13 The dataset has been divided into training and testing sets, designated as X_train, X_test, y_train, and y_test, respectively. This partitioning is essential for effectively training and evaluating machine learning models. Saving the split data into CSV files aids in data management, reproducibility, and ensures that the datasets remain readily accessible for future use in model training and evaluation endeavors. This process is integral for the development of robust and accurate predictive models tailored for credit card payment identification tasks.

6 Evaluation

It evaluates the performance of three machine learning models—Support Vector Classifier (SVC), Artificial Neural Network (ANN), and Convolutional Gated Recurrent Unit (CGRU)—for detecting fraudulent credit card transactions. Evaluation metrics, including precision, recall, and accuracy, are analyzed to compare the effectiveness of each model.

Support Vector Classifier

The Support Vector Classifier (SVC) is a popular machine learning algorithm used for classification tasks. It works by finding the optimal hyperplane that separates different classes in the data space. SVC aims to maximize the margin between classes, making it effective for various applications.

Classification Report– SVC

	precision	recall	f1-score	support
BENIGN	0.88	0.92	0.90	1024
Brute Force	0.93	0.91	0.92	1023
DDoS	0.94	1.00	0.97	1023
DoS	1.00	0.96	0.98	1024
PortScan	1.00	0.94	0.97	1023
accuracy			0.95	5117
macro avg	0.95	0.95	0.95	5117
weighted avg	0.95	0.95	0.95	5117

Fig 14: Classification Report for SVC Model

The classification report as in Fig 14 evaluates the performance of a machine learning model across five classes: BENIGN, Brute Force, DDoS, DoS, and PortScan. Overall, the model exhibits strong performance with an accuracy of 95%. Notably, it achieves high precision scores for most classes, ranging from 88% for BENIGN to 100% for DoS and PortScan, indicating a low rate of false positives. The model also demonstrates high recall rates, correctly identifying a significant portion of instances for each class, particularly noteworthy with a perfect recall score for DDoS. Additionally, the F1-scores, which strike a balance between precision and recall, are consistently high, ranging from 0.90 for BENIGN to 0.98 for DoS. These scores suggest that the model maintains a good balance between minimizing false positives and false negatives across different classes. With support values ranging from 1023 to 1024 instances per class, the dataset appears to be relatively balanced.

Confusion Matrix – SVC

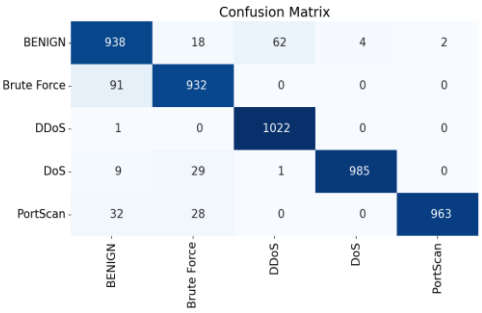

Fig 15: Confusion Matrix for SVC Model

The confusion matrix depicted in Figure 15 illustrates the performance of the SVC model in classifying various types of credit card payment events. It evaluates the accuracy across different event categories: Benign, Brute Force, DDoS, DoS, and PortScan. The model accurately identifies 938 instances of Benign events but makes 86 incorrect predictions. For Brute Force attacks, it accurately identifies 932 cases but misclassifies 91 instances. In DDoS attacks, the model achieves 1022 accurate predictions with only 1 misclassification. Similarly, it correctly identifies 985 instances of DoS attacks but misclassifies 39. Regarding PortScan, the model correctly classifies 963 instances but misclassifies 60 attacks. Overall, the model excels in identifying DDoS attacks.

Artificial Neural Network

Classification Report– ANN

	precision	recall	f1-score	support
BENIGN	0.96	0.95	0.95	1024
Brute Force	0.99	1.00	1.00	1023
DDoS	0.97	0.99	0.98	1023
DoS	1.00	0.99	0.99	1024
PortScan	0.97	0.97	0.97	1023
accuracy			0.98	5117
macro avg	0.98	0.98	0.98	5117
weighted avg	0.98	0.98	0.98	5117

Fig 16: Classification Report for ANN Model

The classification report presented in Figure 16 demonstrate that ANN model performs exceptionally well in accurately identifying and classifying various types of security threats. Across different attack categories, we achieved high precision, recall, and F1-score values, indicating the effectiveness of our model. The precision scores ranged from 0.96 to 1.00, indicating the model's ability to minimize false positives, which is crucial for ensuring the integrity of financial transactions. Additionally, the recall values exceeded 0.95 for all classes, indicating that the model can identify the majority of instances belonging to each attack type, thus ensuring thorough threat detection. The weighted average F1-score of 0.98 further highlights the model's overall reliability in protecting online payment networks from malicious activities. The evaluation shows that all classes perform well, with high precision, recall, and F1-score values across different attack types, indicating effective threat detection.

Confusion Matrix – ANN

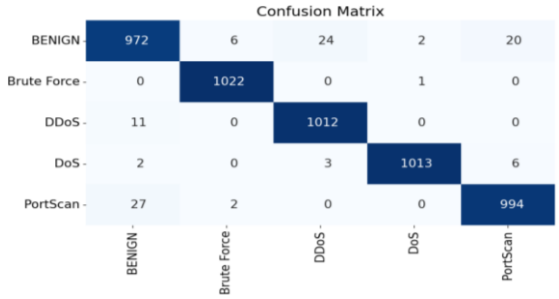

Fig 17: Confusion Matrix for ANN Model

The confusion matrix in Figure 17 shows how well a model artificial neural network (ANN) can figure out different kinds of credit card payment events. It looks at five categories: Benign, Brute Force, DDoS, DoS, and PortScan. Starting with Benign events, the model correctly spots 972 out of 1,024 cases but gets 52 wrong. For Brute Force attacks, it gets 1,022 correctly predicted with only 1 mistake. In DDoS attacks, it gets 1,012 correctly predicted with 11 mistakes. It identifies 1,013 DoS attacks correctly but misclassifies 11 attacks. For PortScan, it gets 994 correctly predicted but misses 29. The ANN model performs well overall, particularly excelling in accurately identifying Brute Force attacks among credit payment events.

Accuracy plot Graph

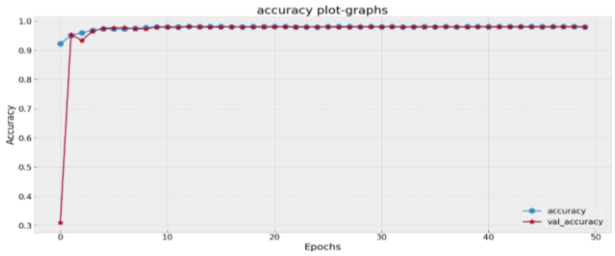

Fig 18: Accuracy Plot graph

The graph provided in Fig 18 shows how well a machine learning model detects credit card fraud over several rounds of training, called epochs. On the graph, the x-axis represents the epochs. The y-axis represents the accuracy of the model. There are two lines on the graph are accuracy and val_accuracy. These likely represent the training and validation accuracies, respectively. Training accuracy shows how well the model performs on the dataset it's being trained on, while validation accuracy shows its performance on a separate dataset that wasn't used for training. This helps understand how well the model generalizes to new data. Ideally, both lines should be close together, indicating the model is learning effectively without overfitting. So, this graph helps assess how well the model is performing and can guide improvements in how it's trained or designed.

Loss plot Graph

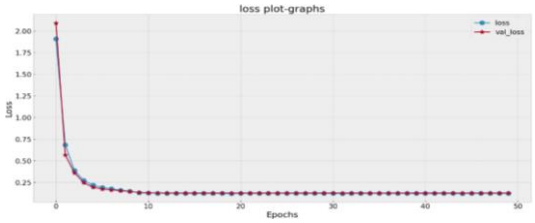

Fig 19: Loss plot graph

The graph displays in Fig 19 is the performance of a machine learning model built for spotting credit card fraud, showing the training and validation loss. The x-axis represents epochs, which are complete passes through the training data, while the y-axis represents loss. The red line shows the training loss, which consistently decreases as the model gets better at fitting the training data. The blue line represents the validation loss, indicating how well the model performs on a separate validation set. Despite some ups and downs, the validation loss also decreases over time, showing that the model is improving at recognizing unseen data. It's important to keep an eye on the validation loss to prevent overfitting, where the model becomes too specialized to the training data and performs poorly on new data. Overall, the graph indicates good progress in training the model for credit card fraud detection.

Convolutional Gated Recurrent Unit

The Convolutional Gated Recurrent Unit (C-GRU) combines the strengths of convolutional neural networks (CNNs) and gated recurrent units (GRUs). It employs convolutional operations to capture spatial features and GRUs for sequential data processing.

Classification Report– CGRU

	precision	recall	f1-score	support
BENIGN	0.993144	0.990234	0.991687	1024
Brute Force	0.995131	0.999022	0.997073	1023
DDoS	0.994158	0.998045	0.996098	1023
DoS	0.999022	0.997070	0.998045	1024
PortScan	0.999020	0.996090	0.997553	1023

Fig 20: Classification Report for CGRU Model

The classification report shown in Fig 20 provides evaluation metrics for a model trained using the CGRU algorithm across five distinct classes: "BENIGN," "Brute Force," "DDoS," "DoS," and "PortScan." The model showcases outstanding performance with consistently high precision, recall, and F1-score values across all classes. Specifically, the "BENIGN" class achieves a precision of 0.99, recall of 0.99, and F1-score of 0.99 with support for 1024 instances. Likewise, the "Brute Force," "DDoS," "DoS," and "PortScan" classes demonstrate precision scores of 0.99, 0.99, 0.99, and 0.99, respectively, along with perfect recall and F1-scores. The overall accuracy of the model is an impressive 99.61%, indicating its capability to accurately classify instances into their respective categories. This comprehensive assessment highlights the effectiveness and reliability of the CGRU algorithm in multi-class classification tasks, especially in identifying various types of network intrusions.

Confusion Matrix – CGRU

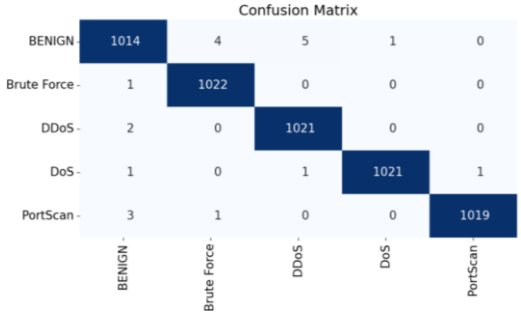

Fig 21: Confusion Matrix for CGRU Model

The confusion matrix shown in Fig 21 depicts the performance of the CGRU model in classifying various types of credit card payment events. It assesses accuracy across different event categories: Benign, Brute Force, DDoS, DoS, and PortScan. Primarily, the model accurately identifies 1014 instances of Benign events but makes 10 incorrect predictions. Remarkably, for Brute Force attacks, it accurately identifies 1022 cases with only 1 misclassification. In DDoS attacks, the model achieves 1021 accurate predictions with only 2 misclassifications. Similarly, It correctly identifies 1021 instances of DoS attacks but misclassifies 3. Regarding PortScan, the model correctly classifies 1019 instances but misclassifies 4 attacks. Overall, the CGRU model excels in accurately identifying Brute Force and DDoS classes.

Accuracy plot Graph

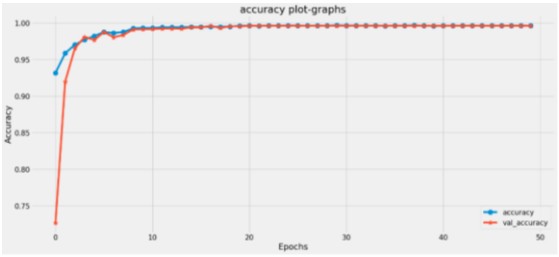

Fig 22: Accuracy Plot graph

The graph as in Fig 22 illustrates the performance of a machine learning model trained for a credit card payment classification task. The x-axis represents these epochs, which are the iterations over the training data, while the y-axis represents the accuracy of the model's predictions. There are two lines on the graph: a blue line labeled "val_accuracy," likely representing accuracy on a separate validation set, and a red line labeled "accuracy," likely representing accuracy on the training data. Generally, we want to see higher and increasing accuracy over epochs, indicating the model is learning from the training data. Monitoring validation accuracy helps prevent overfitting, where a model gets really good at the training data but struggles with new data. In this case, the fact that both training and validation accuracies show similar trends suggests there might not be a significant concern for overfitting, though we would need exact accuracy values to draw definite conclusions.

Loss plot Graph

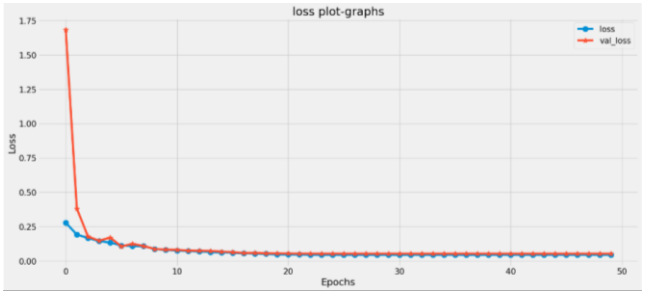

Fig 23: Loss plot graph

The loss plot graph provided in Fig 23 illustrates the training and validation loss of a machine learning model designed for credit card payment fraud detection. The x-axis represents epochs, while the y-axis represents loss. Two lines are depicted: "loss" (blue), indicating the model's loss on the training data, and "val_loss" (orange), representing loss on the validation data. Ideally, both lines should decrease and get close to each other at low values, showing that the model is learning well without overfitting. Additionally, the validation loss shows a slight increase after around 20 epochs, which could mean the model is starting to overfit, becoming too specialized to the training data and not performing as well on new data.

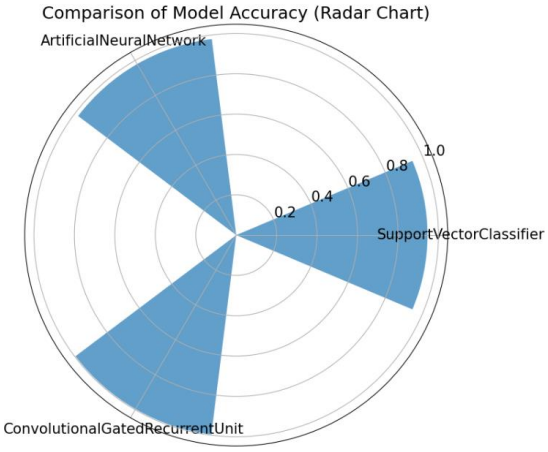

Fig 24: Comparison Graph

The radar chart as shown in Fig 24 compares the accuracy of three machine learning algorithms, Artificial Neural Network (ANN), SVC (Support Vector Classifier) and CGRU (Convolutional Gated Recurrent Unit), on a scale of 0 to 1, regarding their performance in identifying fraudulent credit card transactions. A radar chart illustrates the performance of each algorithm across various metrics, with a focus on accuracy. CGRU notably achieves a perfect score, indicating superior overall accuracy, while SVC and ANN achieve slightly lower accuracy levels, around 0.9 and 0.95, respectively. Overall, CGRU demonstrates the best accuracy performance, surpassing both ANN and SVC.

7 Model Implementation

It outlines a project focusing on securing online financial transactions through Python socket programming, machine learning, and real-time monitoring.

Python socket programming

Python socket programming enables communication between two entities over a network by creating endpoints for sending and receiving data. It allows developers to build client-server applications, facilitating data exchange and interaction across devices or systems. Sockets provide a versatile and efficient means of network communication in Python.

Protecting Online Financial Transactions

Our project focuses on safeguarding online financial transactions to ensure the security of users' sensitive information during payment processing. In today's digital age, online payments have become increasingly common, making it crucial to implement robust security measures to prevent fraud and unauthorized access.

How It Works:

Our system consists of two main components: a server-side application and a client-side application. The server application is responsible for receiving payment information from clients, analyzing it for potential threats, and taking appropriate actions based on the analysis. On the other hand, the client application allows users to initiate payment transactions securely.

Server-Side Application:

The server-side application is developed using Python and relies on various libraries and modules such as Twilio for communication, Pandas for data manipulation, and Flask for creating a web server. The core functionality of the server involves receiving payment data from clients, analyzing it using machine learning models, and updating the status of payment methods (such as credit cards or UPI IDs) based on the analysis results.

Client-Side Application:

The client-side application is a web-based interface built using Flask, a lightweight web framework. Users interact with the client application to initiate payment transactions by

providing necessary details such as payment amount, method, and payment credentials. The client application then communicates with the server to process the transaction securely.

Machine Learning for Threat Detection:

One of the key features of our system is its ability to predict potential threats during payment transactions using machine learning algorithms. We have trained a Convolutional Gated Recurrent Unit (CGRU) model using historical payment data to classify transactions into different categories such as benign, brute force, DDoS, DoS, or PortScan attacks. This prediction helps in identifying suspicious activities and taking proactive measures to mitigate risks.

Enhancing Security:

To enhance security, our system implements several measures such as real-time monitoring of payment transactions, validation of payment credentials, and updating the status of payment methods based on analysis results. Additionally, users are notified of transaction outcomes via SMS, providing transparency and accountability.

Client page: This webpage is specifically designed for our clients, allowing them to provide file details and choose their preferred payment method. Two payment options are offered: credit/debit card payment and UPI transaction.

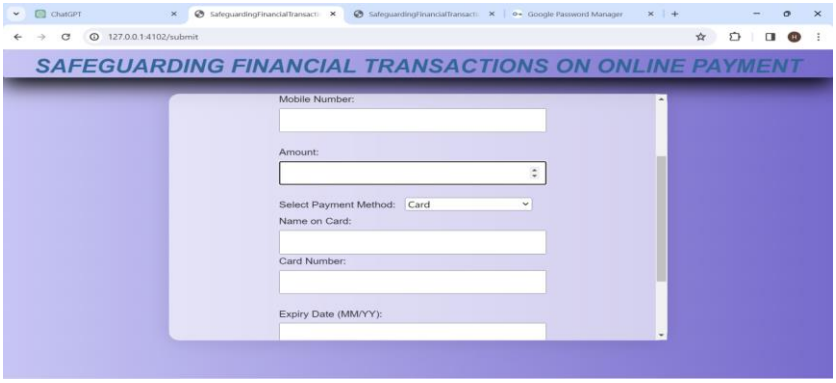

Fig 25: User interface

Client Card Page: When opting for 'card' as the payment method on this page, users are required to input all the relevant details of their card.

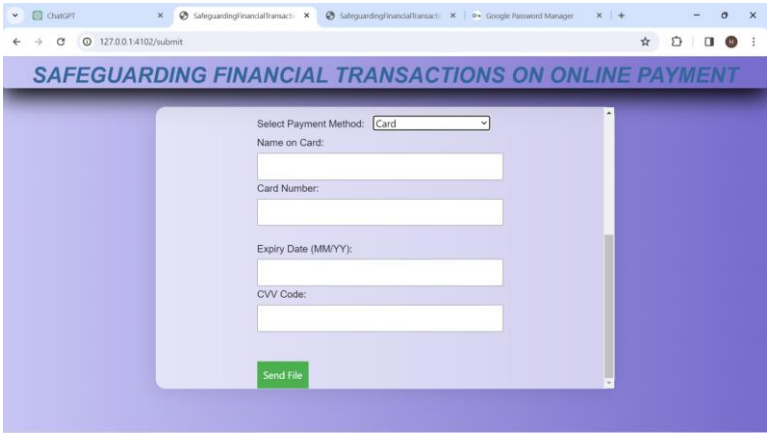

Fig 26: Payment method selection

Client UPI Page: When choosing UPI as the payment method on this page, users must provide their name and UPI ID.

Home Page: This acts as the central hub where we showcase various client details, including the date and time they accessed the system displayed under a timestamp. Furthermore, it presents the IP address of the client accessing the system, while the 'attack feature' provides insight into the type of file attack encountered.

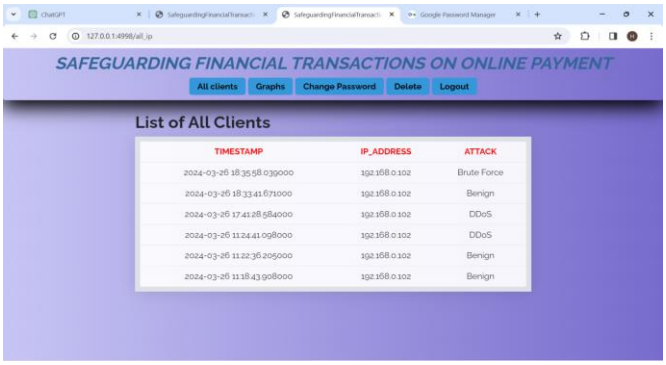

Fig 27: Listing all clients

Both of these are server pages.
Regarding benign sharing: This server page involves a normal transaction carried out by an active client.

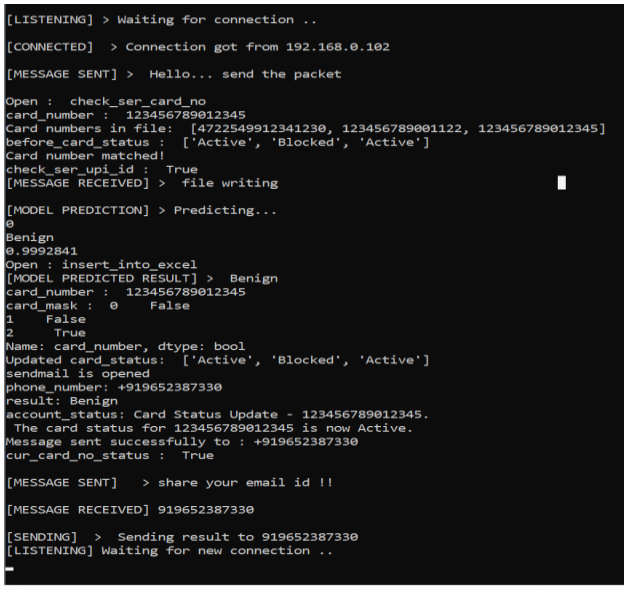

Fig 28: Client transaction logs

File-sharing attack: This server page indicates that a client conducted a transaction, resulting in a brute force attack and subsequent account is blocked.

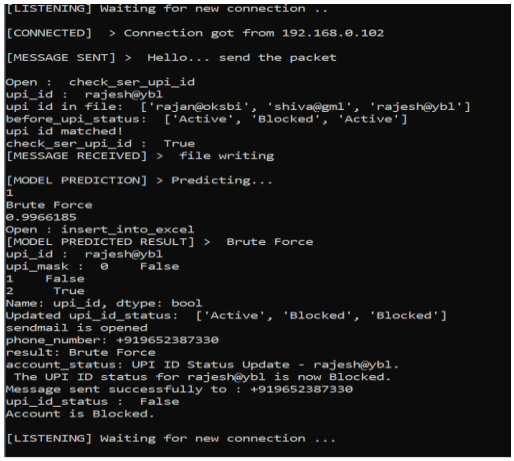

Fig 1: Online Payment Network

8. Conclusion & Future Work

In conclusion, the development of an advanced Intrusion Detection System (IDS) customized for protecting online payments, utilizing cutting-edge machine learning methods, represents a significant advancement in strengthening the security of digital financial transactions. This system aims to analyze potential threats carefully and quickly detect and prevent unauthorized access in real-time to reduce risks and stop fraudulent activities effectively. This research thoroughly assesses the effectiveness of three different machine learning approaches – Artificial Neural Network (ANN), Support Vector Classifier (SVC), and CGRU (Convolutional Gated Recurrent Unit) – in identifying fraudulent credit card transactions. Notably, CGRU stood out as the top performer, achieving an impressive accuracy rate of 99.61%. In comparison, SVC and ANN produced slightly lower accuracy rates of 94.59% and 97.97%, respectively. This highlights the significant superiority of CGRU in accurately detecting fraudulent transactions compared to the other methods. These findings emphasize the critical importance of integrating AI-based intrusion detection mechanisms to ensure the reliability and security of online payment networks. By offering strong defense against fraudulent activities, the proposed system greatly contributes to building trust and confidence in digital financial transactions, thereby creating a more resilient and secure online payment environment. The future scope encompasses integrating block chain technology to heighten transaction security, rolling out the system across diverse financial institutions on a global scale, and continually enhancing machine learning algorithms to adjust to evolving fraud tactics.

9 References

1. Wang, Fei, et al. "Machine learning for mobile network payment security evaluation system." *Transactions on Emerging Telecommunications Technologies* (2021): e4226.
2. Hajek, Petr, Mohammad Zoynul Abedin, and Uthayasankar Sivarajah. "Fraud detection in mobile payment systems using an XGBoost-based framework." *Information Systems Frontiers* 25.5 (2023): 1985-2003.
3. Behera, Rajat Kumar, Abhaya Kumar Sahoo, and Ajay Jena. "A resourceful approach in security testing to protect electronic payment system against unforeseen attack." *Research Anthology on Artificial Intelligence Applications in Security*. IGI Global, 2021. 1279-1302.
4. Halimaa, Anish, and K. Sundarakantham. "Machine learning based intrusion detection system." 2019 3rd International conference on trends in electronics and informatics (ICOEI). IEEE, 2019. https://ieeexplore.ieee.org/document/8862784
5. Almutairi, Yasmeen S., Bader Alhazmi, and Amr A. Munshi. "Network intrusion detection using machine learning techniques." *Advances in Science and Technology Research Journal* 16.3 (2022): 193-206.
6. Bertoli, Gustavo De Carvalho, et al. "An end-to-end framework for machine learning-based network intrusion detection system." *IEEE Access* 9 (2021): 106790-106805.
7. Tran, Ngan, et al. "Data Curation and Quality Evaluation for Machine Learning-Based Cyber Intrusion Detection." *IEEE Access* 10 (2022): 121900-121923.
8. Lirim Ashiku, Cihan Dagli, Network Intrusion Detection System using Deep Learning, Procedia Computer Science, Volume 185, 2021,Pages 239-247,ISSN 1877-0509, https://doi.org/10.1016/j.procs.2021.05.025.(https://www.sciencedirect.com/science/article/pii/S1877050921011078)
9. Patel, Kaushikkumar. "Credit Card Analytics: A Review of Fraud Detection and Risk Assessment Techniques." *International Journal of Computer Trends and Technology* 71.10 (2023): 69-79.
10. Sangeetha, R., et al. "An Innovation Detection of Vulnerabilities for Digital Transactions in Financial Institutions Using Cyber Security Framework." *International Journal of Intelligent Systems and Applications in Engineering* 11.3 (2023): 70-76.
11. Lansky, Jan, et al. "Deep learning-based intrusion detection systems: a systematic review." *IEEE Access* 9 (2021): 101574-101599.
12. Halbouni, Asmaa, et al. "Machine learning and deep learning approaches for cybersecurity: A review." *IEEE Access* 10 (2022): 19572-19585.
13. Shafique, Hassan, et al. "Machine learning empowered efficient intrusion detection framework." (2022). https://vfast.org/journals/index.php/VTSE/article/view/1017