IDENTIFICATION OF CREDIT CARD FRAUDS USING MACHINE LEARNING AND DEEP LEARNING

A Project Report submitted to
Jawaharlal Nehru Technological University
in partial fulfillment of the requirements for the award of Degree of

BACHELOR OF TECHNOLOGY
in
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Submitted by

ARAVIND.M	**21831A7203**
C.S ABHINAY	**21831A7214**
D.S PRATHAMESH	**21831A7217**

Under the Guidance of

Dr. Sheetal Kundra
Professor & Head

Department of Artificial Intelligence and Data Science
Guru Nanak Institute of Technology
Ibrahimpatnam, Hyderabad, R.R. District – 501506

May, 2025

GURU NANAK INSTITUTE OF TECHNOLOGY
(Autonomous)
Ibrahimpatnam, Ranga Reddy District-501506

CERTIFICATE

This is to certify that the project entitled IDENTIFICATION OF CREDIT CARD FRAUDS USING MACHINE LEARNING AND DEEP LEARNING is being submitted by Mr. ARAVIND.M, C.S ABHINAY, D.S PRATHAMESH, bearing Roll No.21831A7203, 21831A7214, 21831A7217, in partial fulfilment for the award of the Degree of Bachelor of the Technology in ARTIFICIAL INTELLIGENCE AND DATA SCIENCE to the Jawaharlal Nehru Technological University is a record of Bonafide work carried out by him/her/them under my guidance and supervision.

The results embodied in this Major-project/project report have not been submitted to any other University or Institute for the award of any Degree or Diploma

Internal Guide

Dr. Sheetal Kundra

Professor & Head

Head of the Department

External Examiner

GURU NANAK INSTITUTE OF TECHNOLOGY
(Autonomous)
Ibrahimpatnam, Ranga Reddy District-501506

DECLARATION OF STUDENT

We, ARAVIND.M (21831A7203), C.S ABHINAY (21831A7214), D.S PRATHAMESH (21831A7217), hereby declare that the major project titled "IDENTIFICATION OF CREDIT CARD FRAUDS USING MACHINE LEARNING AND DEEP LEARNING" has been carried out by us as part of the requirements for the award of the Degree of Bachelor of Technology in the Department of Artificial Intelligence and Data Science at Guru Nanak Institute of Technology.

We confirm the following:

1. The project was undertaken by us under the supervision of our guide, Dr. Sheetal Kundra, from the selection of the topic to the completion of the final report.
2. We have ensured that the results presented in the report are accurate and based on our original work.
3. To the best of our knowledge, the content of this report is free from plagiarism and adheres to ethical standards.
4. Each member of the team has contributed significantly and appropriately to the project work.
5. The project report has been prepared with diligence, ensuring clarity, accuracy, and adherence to academic standards.

We further declare that this report has not been submitted, in part or full, to any other institution or university for the award of any degree or diploma.

Aravind. M Student Signature

21831A7203

C.S Abhinay Student Signature

21831A7214

D.S Prathamesh Student Signature

21831A7217

Date:

Place: Ibrahimpatnam.

GURU NANAK INSTITUTE OF TECHNOLOGY
(Autonomous)
Ibrahimpatnam, Ranga Reddy District-501506

DECLARATION OF GUIDE

I, Dr. Sheetal Kundra, hereby declare that I have guided the major project titled "IDENTIFICATION OF CREDIT CARD FRAUDS USING MACHINE LEARNING AND DEEP LEARNING" undertaken by Aravind. M, C.S Abhinay, D.S Prathamesh (21831A7203,21831A7214,21831A7217). This project was carried out towards the fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Artificial Intelligence and Data Science at Guru Nanak Institute Of Technology.

As the guide, I confirm the following:

1. I have overseen the entire project process, from the selection of the project title to the submission of the final report.
2. I have reviewed and certified the accuracy and relevance of the results presented in the report.
3. The work carried out is original, free from plagiarism, and adheres to ethical guidelines.
4. The contributions of each student have been appropriately recognized and assessed.
5. The project report has been prepared under my supervision, ensuring adherence to high standards of quality, clarity, and structure.

I further certify that this project report has not been previously submitted in part or full for the award of any degree or diploma by any institution or university.

Name of Guide: Dr. Sheetal Kundra Signature of the Guide

Date:

Place: Ibrahimpatnam

Name of HOD: Dr. Sheetal Kundra Signature of the HOD

Department: Artificial Intelligence & Data Science

Date:

Place: Ibrahimpatnam

GURU NANAK INSTITUTE OF TECHNOLOGY
(Autonomous)
Ibrahimpatnam, Ranga Reddy District-501506

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to **Dr. Sheetal Kundra**, Professor and Head of Artificial Intelligence and Data Science, for her invaluable guidance, expert supervision, and continuous support throughout the duration of this project. Her encouragement and insightful suggestions played a significant role in the successful completion of our work.

We would also like to thank the faculty members of the **Artificial Intelligence and Data Science** and the Lab Technicians for their assistance and cooperation during the practical work of our project.

We are grateful to our friends and well-wishers for their encouragement, collaboration, and useful feedback throughout the project journey.

Lastly, we sincerely thank our parents for their constant support, patience, and motivation, which helped us complete this project successfully.

Aravind. M	21831A7203
C.S Abhinay	21831A7214
D.S Prathamesh	21831A7217

LIST OF CONTENT

ABSTRACT

The rapid proliferation of digital financial transactions has significantly escalated the risk of fraudulent activities, posing an increasingly serious challenge to financial security systems worldwide. Traditional fraud detection methods, typically reliant on rule based systems and classical machine learning algorithms, often struggle to adapt to the constantly evolving tactics used by fraudsters. These legacy methods are especially inadequate in handling imbalanced datasets and learning from sequential transaction behaviour, leading to higher false positive rates and lower detection accuracy. In this project, we present a robust and intelligent credit card fraud detection framework grounded in deep learning techniques, specifically utilizing Bidirectional Long Short-Term Memory (BiLSTM) networks. To establish a performance baseline and enable comparative analysis, we also employed traditional ensemble learning models, including Gradient Boosting, Random Forest, and Logistic Regression. While these models offer strong interpretability and generalization capabilities, their limitations in processing sequential data necessitated a more dynamic solution. The BiLSTM architecture, known for its capability to learn long-term dependencies in both forward and backward temporal directions, was implemented to effectively detect subtle anomalies in transaction sequences.

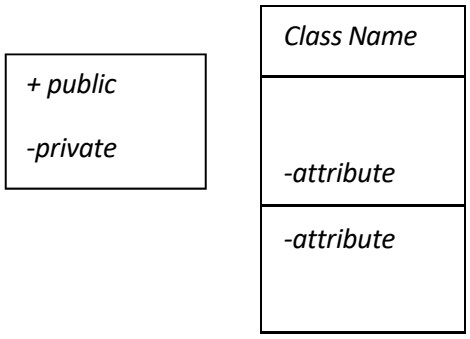
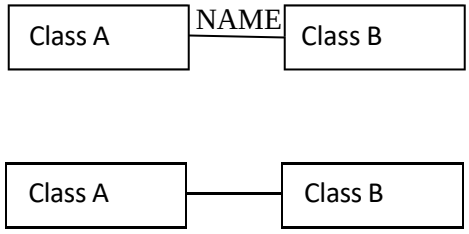
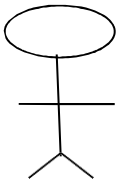
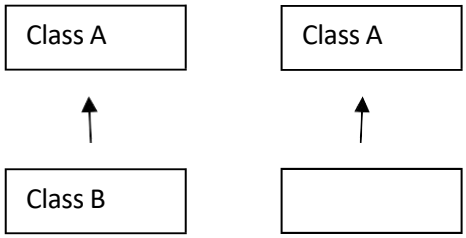
Preprocessing included data cleaning, feature scaling using Standard Scaler, and reshaping to fit the sequential input format required by BiLSTM. The issue of class imbalance, a common characteristic in fraud detection datasets, was addressed through the use of under-sampling techniques to ensure better representation of minority class instances during training. Although SMOTE was initially considered, it was not included in the final model.

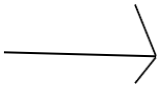
The model was built and trained using TensorFlow and Keras frameworks. Hyperparameter tuning was conducted through extensive experimentation to improve generalization and reduce overfitting. The final BiLSTM model achieved an outstanding accuracy of 99.96%, along with a macro average precision of 91.83%, recall of 99.98%, and F1-score of 95.54%. These results highlight the system's ability to outperform conventional fraud detection approaches.

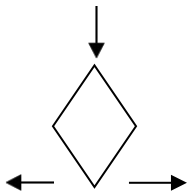
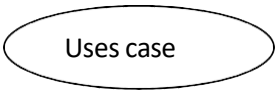
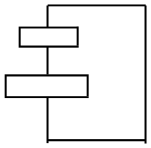
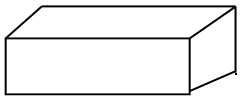
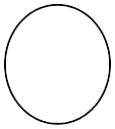
Additionally, the trained model has been deployed through a user-accessible web platform featuring an integrated AI chatbot and AI Agent. This interface enables real-time fraud detection support, user interaction for credit-related inquiries, and ensures the system's scalability for real-world financial applications.

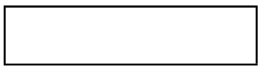
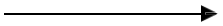
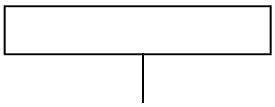
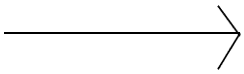
LIST OF FIGURES

LIST OF NOTATIONS

S.NO	NAME	NOTATION	DESCRIPTION
1.	Class		Represents a collection of similar entities grouped together.
2.	Association		Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single class.
4.	Aggregation		Interaction between the system and external environment

5.	*Relation* (uses)	*Uses*	Used for additional process communication.
6.	Relation (extends)		Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.
8.	State		State of the process.
9.	Initial State		Initial state of the object
10.	Final state		Final state of the object
11.	Control flow		Represents various control flow between the states.

12.	Decision box		Represents decision making process from a constraint
13.	Use-case		Interaction between the system and external environment.
14.	Component		Represents physical modules which is a collection of components.
15.	Node		Represents physical modules which are a collection of components.
16.	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or action.

17.	External entity		Represents external entities such as keyboard, sensors, etc.
18.	Transition		Represents communication that occurs between processes.
19.	Object Lifeline		Represents the vertical dimensions that the object communications.
20.	Message	Message 	Represents the message exchanged.

LIST OF ABBREVIATIONS

S.NO	ABBREVATION	EXPANSION
1.	ML	Machine Learning
2.	AI	Artificial Intelligence
3.	RF	Random Forest
4.	GBM	Gradient Boosting Machine
5.	LR	Logistic Regression
6.	Bi LSTM	Bidirectional Long Short-Term Memory
7.	SGD	Stochastic Gradient Descent

CHAPTER 1

INTRODUCTION

1.1 General

In recent years, the rapid growth of digital financial transactions has greatly enhanced convenience for consumers and businesses. However, this increase has also led to a significant rise in fraudulent activities, particularly in the context of credit card transactions. Detecting and preventing fraud in real time has become an essential challenge for financial institutions and organizations. Traditional fraud detection systems rely heavily on predefined rules and statistical methods. While effective to some extent, these techniques often fail to adapt to evolving fraud patterns and result in a high number of false positives, which can frustrate legitimate users. To address these challenges, this project explores the development of a deep learning-based model for credit card fraud detection. The proposed system utilizes advanced machine learning techniques, particularly ensemble learning and data balancing strategies, to improve detection accuracy and reduce false positives.

The credit card fraud detection system aims to revolutionize how financial institutions combat fraudulent transactions by leveraging deep learning and ensemble machine learning techniques. Unlike traditional fraud detection methods, which rely on static rules and basic statistical analysis, this system offers a dynamic and adaptive approach capable of identifying evolving fraud patterns. By integrating Gradient Boosting, Random Forest, and Logistic Regression models, the system capitalizes on the complementary strengths of each algorithm, ensuring more accurate, robust, and generalized fraud prediction outcomes.[2]

A significant aspect of this project is its ability to address the issue of class imbalance, a common challenge in fraud detection, where fraudulent transactions make up a very small fraction of the overall data. By applying a combination of SMOTE (Synthetic Minority Over-sampling Technique) and under-sampling strategies, the system ensures balanced training data, which reduces model bias and improves detection sensitivity. This preprocessing step is essential in enhancing the system’s ability to correctly identify fraudulent activities without increasing false positives.

Built using TensorFlow and Keras, the deep learning framework provides the flexibility needed for experimentation and hyperparameter tuning, resulting in a model that achieves an accuracy of 99.59%. This high performance underscores the system's readiness for real-world deployment, providing financial institutions with a scalable and efficient solution to a growing cybersecurity threat. Furthermore, the model's architecture allows for future enhancements, including real-time transaction monitoring and the incorporation of more sophisticated neural networks.[2]

Beyond its technical merits, the proposed fraud detection system contributes to financial security by minimizing potential losses and protecting consumer trust. Its scalability ensures that the model can be adapted for different financial institutions, transaction types, and evolving fraud tactics. By automating complex fraud detection processes and reducing reliance on manual rule creation, the system significantly enhances operational efficiency while maintaining high levels of accuracy and reliability.

1.2 Objective

- To explore and implement advanced machine learning and deep learning models, including sequential and ensemble techniques, for effective and robust fraud detection in online financial transactions.
- To enhance model performance through feature engineering and optimization for real-time detection under high-load scenarios.

1.3 Existing System

Current fraud detection mechanisms rely on rule-based systems and basic machine learning algorithms, such as decision trees and logistic regression. However, these methods struggle with imbalanced datasets and evolving fraud patterns. Some of the limitations of the existing system include:[2]

1.3.1 Existing System Disadvantages

- ❖ High false positive rates.
- ❖ Ineffective handling of imbalanced datasets.
- ❖ Limited adaptability to new fraud patterns.

1.4 Proposed System

The proposed system introduces a deep learning-based sequential model for credit card fraud detection, incorporating ensemble learning techniques to improve prediction accuracy and robustness. Addressing the challenges of imbalanced datasets, the system applies Synthetic Minority Over-sampling Technique (SMOTE) and under-sampling to ensure a balanced training dataset and reduce bias toward non-fraudulent transactions.

Key components of the system include the integration of ensemble learning algorithms such as Gradient Boosting, Random Forest, and Voting Classifiers. These models work together to combine their individual strengths, enhancing the model's ability to detect fraudulent activities while minimizing false positives. Additionally, the system leverages temporal data and transaction patterns for feature extraction, enabling the model to identify subtle behavioral changes indicative of fraud.

Hyperparameter tuning was performed to optimize the model's performance, reducing overfitting and improving generalization on new data. The final deep learning model, developed using TensorFlow and Keras frameworks, achieves high accuracy, demonstrating its effectiveness in detecting fraud with minimal false positives.

This research highlights the potential of deep learning and ensemble learning in financial security applications, offering a scalable and reliable solution to the growing threat of fraud in digital financial transactions.

CHAPTER 2
LITERATURE SURVEY

Title: Credit card fraud detection using artificial neural network

Author: Asha RB, Suresh Kumar KR

Year: 2021

Description:

Frauds in credit card transactions are common today as most of us are using the credit card payment methods more frequently. This is due to the advancement of Technology and increase in online transaction resulting in frauds causing huge financial loss. Therefore, there is need for effective methods to reduce the loss. In addition, fraudsters find ways to steal the credit card information of the user by sending fake SMS and calls, also through masquerading attack, phishing attack and so on. This paper aims in using the multiple algorithms of Machine learning such as support vector machine (SVM), k-nearest neighbor (KNN) and artificial neural network (ANN) in predicting the occurrence of the fraud. Further, we conduct a differentiation of the accomplished supervised machine learning and deep learning techniques to differentiate between fraud and non-fraud transactions.

Title: Intelligent mechanical systems and its applications on online fraud detection analysis using pattern recognition K-nearest neighbor algorithm for cloud security applications

Author: A. Kannagi, J. Gori Mohammed, S. Sabari Giri Murugan, and M. Varsha

Year: 2023

Description:

Banks and financial institutions use systematic business transaction fraud, individual analytics. Distrust analysis definition is based on a customer history of abnormal behaviors that indicate normal activities and fraud, manipulation or application analysis. The challenge of facing the banking and finance sector is in the form of fraudulent transactions. Combined with the explosive development of data and information, it has urged great advances in technology and communication, global consumer awareness, and capabilities. A comprehensive customer engagement strategy of the banking sector should establish opportunities through dynamic consumer estimation. Security is therefore essential for all banks to implement fraud detection methods to minimize their losses. The regulatory sequence based on different methods of machine learning has evolved to detect various fraudulent transactions. The effective way to detect the fraud detection mechanism is far more essential than the standard classification technology. Pattern Recognition K-Nearest Neighbor [PR-KNN] is developed for such a progressive search and non-parametric technique that evaluates the pseudo-nature to find the best solution to the problem. Regression and classification problems of fraudulent transactions mean the spending is reducing the number of false alarms. The detection systems are placed in bank fraud or fraudulent transactions after the transaction are expected to expectation.

Title: Deep convolution neural network model for credit-card fraud detection and alert

Author: J. I.-Z. Chen and K.-L. La

Year: 2021

Description:

With the exponential increase in the usage of the internet, numerous organizations, including the financial industry, have operationalized online services. The massive financial losses occur as a result of the global growth in financial fraud. Henceforth, devising advanced financial fraud detection systems can actively detect the risks such as illegal transactions and irregular attacks. Over the recent years, these issues are tackled to a larger extent by means of data mining and machine learning techniques. However, in terms of unknown attack pattern identification, big data analytics and speed computation, several improvements must be performed in these techniques. The Deep Convolution Neural Network (DCNN) scheme based financial fraud detection scheme using deep learning algorithm is proposed in this paper. When large volume of data is involved, the detection accuracy can be enhanced by using this technique. The existing machine learning models, auto-encoder model and other deep learning models are compared with the proposed model to evaluate the performance by using a real-time credit card fraud dataset. Over a time duration of 45 seconds, a detection accuracy of 99% has been obtained by using the proposed model as observed in the experimental results.

Title: Application of deep learning for credit card approval: A comparison with two machine learning techniques

Author: M. G. Kibria and M. Sevkli

Year: 2021

Description:

The increased credit card defaulters have forced the companies to think carefully before the approval of credit applications. Credit card companies usually use their judgment to determine whether a credit card should be issued to the customer satisfying certain criteria. Some machine learning algorithms have also been used to support the decision. The main objective of this paper is to build a deep learning model based on the UCI (University of California, Irvine) data sets, which can support the credit card approval decision. Secondly, the performance of the built model is compared with the other two traditional machine learning algorithms: logistic regression (LR) and support vector machine (SVM). Our results show that the overall performance of our deep learning model is slightly better than that of the other two models.

Title: Comparison and analysis of logistic regression, naive Bayes and KNN machine learning algorithms for credit card fraud detection

Author: F. Itoo, Meenakshi, and S. Singh

Year: 2021

Description:

Financial fraud is a threat which is increasing on a greater pace and has a very bad impact over the economy, collaborative institutions and administration. Credit card transactions are increasing faster because of the advancement in internet technology which leads to high dependence over internet. With the up-gradation of technology and increase in usage of credit cards, fraud rates become challenge for economy. With inclusion of new security features in credit card transactions the fraudsters are also developing new patterns or loopholes to chase the transactions. As a result of which behavior of frauds and normal transactions change constantly. Also the problem with the credit card data is that it is highly skewed which leads to inefficient prediction of fraudulent transactions. In order to achieve the better result, imbalanced or skewed data is pre-processed with the re-sampling (over-sampling or under sampling) technique for better results. The three different proportions of datasets were used in this study and random under-sampling technique was used for skewed dataset. This work uses the three machine learning algorithms namely: logistic regression, Naïve Bayes and K-nearest neighbor. The performance of these algorithms is recorded with their comparative analysis. The work is implemented in python and the performance of the algorithms is measured based on accuracy, sensitivity, specificity, precision, F-measure and area under curve. On the basis these measurements logistic regression based model for prediction of fraudulent was found to be a better in comparison to other prediction models developed from Naïve Bayes and K-nearest neighbor. Better results are also seen by applying under sampling techniques over the data before developing the prediction model.

CHAPTER 3

DESIGN AND DEVELOPMENT

3.1 General

DL is at the heart of a slew of high-tech breakthroughs. Deep understanding is being used in various industries to ensure its success. Because most DL approaches rely on neural network topologies, DL models are also defined as deep neural networks. With a DL model, an algorithm can use its neural network to decide whether a prediction is correct.[2]

3.2 System Architecture

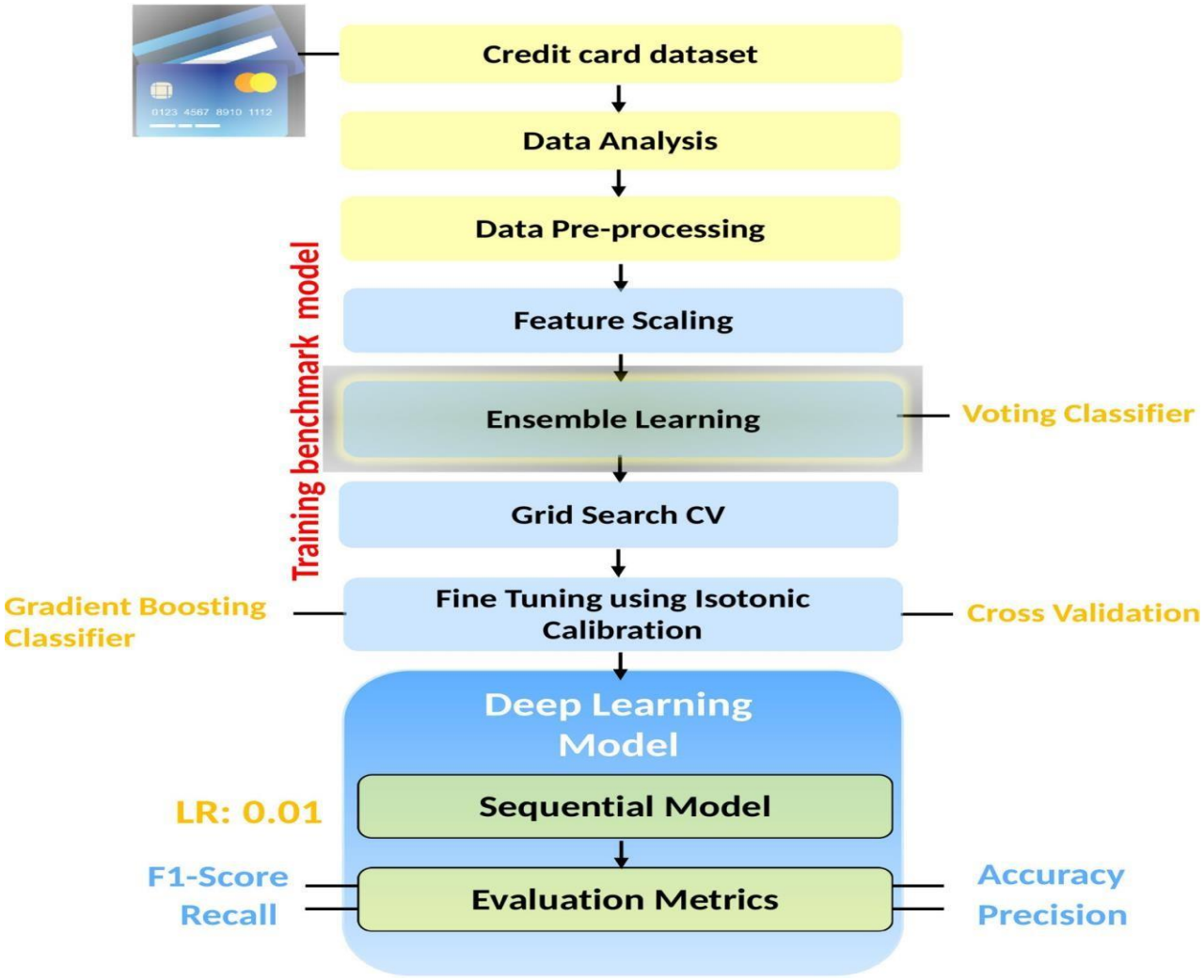


Fig 3.2: System Architecture

3.3 Design Methodology

1. Data set

2. Analyzing

3. Preprocessing

4. Data sampling

5. Model

6. Prediction

1. Data set

We utilized the Credit Card Fraud Detection dataset, comprising 284,807 transactions with 492 fraud cases (0.172%). It contains 28 anonymized features using PCA, alongside 'Time' and 'Amount'. Preprocessing included handling missing values, standardizing 'Amount', removing 'Time' due to minimal relevance, and detecting and eliminating duplicate entries.

P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
1.46818	-0.4704	0.20797	0.02579	0.40399	0.25141	-0.01831	0.27784	-0.11047	0.06693	0.12854	-0.18911	0.13356	-0.02105	149.62	0
0.63556	0.46392	-0.1148	-0.18336	-0.14578	-0.06908	-0.22578	-0.63867	0.10129	-0.33985	0.16717	0.12589	-0.00898	0.01472	2.69	0
2.34586	-2.89008	1.10997	-0.12136	-2.26186	0.52498	0.248	0.77168	0.90941	-0.68928	-0.32764	-0.1391	-0.05535	-0.05975	378.66	0
-0.63142	-1.05965	-0.68409	1.96578	-1.23262	-0.20804	-0.1083	0.00527	-0.19032	-1.17558	0.64738	-0.22193	0.06272	0.06146	123.5	0
0.17512	-0.45145	-0.23703	-0.03819	0.80349	0.40854	-0.00943	0.79828	-0.13746	0.14127	-0.20601	0.50229	0.21942	0.21515	69.99	0
0.51762	0.40173	-0.05813	0.06865	-0.03319	0.08497	-0.20825	-0.55982	-0.0264	-0.37143	-0.23279	0.10591	0.25384	0.08108	3.67	0
0.05014	-0.44359	0.00282	-0.61199	-0.04558	-0.21963	-0.16772	-0.27071	-0.1541	-0.78006	0.75014	-0.25724	0.03451	0.00517	4.99	0
0.68613	-0.07613	-1.22213	-0.35822	0.3245	-0.15674	1.94347	-1.01545	0.0575	-0.64971	-0.41527	-0.05163	-1.20692	-1.08534	40.8	0
-0.32878	-0.21008	-0.49977	0.11876	0.57033	0.05274	-0.07343	-0.26809	-0.20423	1.01159	0.3732	-0.38416	0.01175	0.1424	93.2	0
0.15022	0.73945	-0.54098	0.47668	0.45177	0.20371	-0.24691	-0.63375	-0.12079	-0.38505	-0.06973	0.0942	0.24622	0.08308	3.68	0
0.23093	0.03197	0.25341	0.85434	-0.22137	-0.38723	-0.0093	0.31389	0.02774	0.50051	0.25137	-0.12948	0.04285	0.01625	7.8	0
0.9289	-0.12949	-0.80998	0.35999	-0.70766	0.12599	0.04992	0.23842	0.00913	0.99671	-0.76731	-0.49221	0.04247	-0.05434	9.99	0
0.72567	-0.81561	0.87394	-0.84779	-0.68319	-0.10276	-0.23181	-0.48329	0.08467	0.39283	0.16113	-0.35499	0.02642	0.04242	121.5	0
-0.65556	-0.19993	0.12401	-0.9805	-0.98292	-0.1532	-0.03688	0.07441	-0.07141	0.10474	0.54826	0.10409	0.02149	0.02129	27.5	0
0.4068	-0.30306	-0.15587	0.77827	2.22187	-1.58212	1.15166	0.22218	1.02059	0.02832	-0.23275	-0.23556	-0.16478	-0.03015	58.8	0
1.10695	1.66011	-0.27927	-0.41999	0.43254	0.26345	0.49962	1.35365	-0.25657	-0.06508	-0.03912	-0.08709	-0.181	0.12939	15.99	0
0.35457	-0.24663	-0.00921	-0.59591	-0.57568	-0.11391	-0.02461	0.196	0.0138	0.10376	0.3643	-0.38226	0.09281	0.03705	12.99	0
-0.18452	1.14317	-0.92871	0.68047	0.02544	-0.04702	-0.1948	-0.67264	-0.15686	-0.88839	-0.34241	-0.04903	0.07969	0.13102	0.89	0
-0.52661	0.472	-0.72548	0.07508	-0.40687	-2.19685	-0.5036	0.98446	2.45859	0.04212	-0.48163	-0.62127	0.39205	0.94959	46.8	0
-0.04298	-0.16643	0.30424	0.55443	0.05423	-0.38791	-0.17765	-0.17507	0.04	0.29581	0.33293	-0.22038	0.0223	0.0076	5	0
-0.80798	-2.04456	0.51566	0.62585	-1.30041	-0.13833	-0.29558	-0.57196	-0.05088	-0.30421	0.072	-0.42223	0.08655	0.0635	231.71	0
0.71091	-0.60223	0.40248	-1.73716	-2.02761	-0.26932	0.144	0.40249	-0.04851	-1.37187	0.39081	0.19996	0.01637	-0.01461	34.09	0
0.00308	0.42442	-0.45448	-0.09887	-0.8166	-0.30717	0.0187	-0.06197	-0.10385	-0.37042	0.6032	0.10856	-0.04052	-0.01142	2.28	0
1.5552	-1.39689	0.78313	0.43662	2.17781	-0.23098	1.65018	0.20045	-0.18535	0.42307	0.82059	-0.22763	0.33663	0.25048	22.75	0
1.19072	0.57884	-0.97567	0.04406	0.4886	-0.21672	-0.57953	-0.79923	0.8703	0.98342	0.3212	0.14965	0.70752	0.0146	0.89	0
-0.21167	-0.33332	0.01075	-0.48847	0.50575	-0.38669	-0.40364	-0.2274	0.74243	0.39853	0.24921	0.2744	0.35997	0.24323	26.43	0
0.76148	-0.01458	-0.51164	-0.32506	-0.39093	0.02788	0.067	0.22781	-0.15049	0.43505	0.72482	-0.33708	0.01637	0.03004	41.88	0
0.78217	-1.35587	-0.21694	1.27177	-1.24062	-0.52295	-0.28438	-0.32336	-0.03771	0.34715	0.55964	-0.28016	0.04234	0.02882	16	0
0.0768	-1.40592	0.77559	-0.94289	0.54397	0.09731	0.07724	0.45733	-0.0385	0.64252	-0.18389	-0.27746	0.18269	0.15266	33	0
-0.21868	0.00448	-0.19355	0.04239	-0.27783	-0.17802	0.01368	0.21373	0.01446	0.00295	0.29464	-0.39507	0.08146	0.02422	12.99	0
-0.21839	-0.19155	-0.11658	-0.63379	0.34842	-0.06635	-0.24568	-0.5309	-0.04427	0.07917	0.50914	0.28886	-0.0227	0.01184	17.28	0
-0.67459	-0.52911	0.15826	-0.39875	-0.14571	-0.27383	-0.05323	-0.00476	-0.03147	0.19805	0.56501	-0.33772	0.02906	0.00445	4.45	0
-1.51977	-0.28438	-0.31052	-0.40425	-0.82337	-0.29035	0.04695	0.2081	-0.18555	0.00103	0.09882	-0.5529	-0.07329	0.02331	6.14	0
-1.51977	-0.28438	-0.31052	-0.40425	-0.82337	-0.29035	0.04695	0.2081	-0.18555	0.00103	0.09882	-0.5529	-0.07329	0.02331	6.14	0

Fig 3.3: Data Set Image

2. Analyzing

The Credit Card Fraud Detection dataset contains 284,807 transactions, with only 492 (0.17%) classified as fraud, indicating severe class imbalance. It includes 28 anonymized PCA-applied features, along with 'Time' and 'Amount'. Due to limited feature descriptions, feature selection was challenging, and the 'Time' feature was excluded for minimal relevance.

3. Preprocessing

Data preprocessing prepares raw data for machine learning by handling missing values, encoding data, standardization, PCA transformation, and feature scaling. Min-Max Normalization scaled features between 0 and 1 using Sklearn's Min Max Scaler. Given the dataset's severe class imbalance (99.83% legitimate vs. 0.17% fraud), resampling strategies were essential

4. Data sampling

Pre-processing, data imbalance was addressed using under-sampling and SMOTE. Under-sampling reduced the majority class to match the minority class size, while SMOTE synthetically generated new samples for the minority class. This resulted in a balanced dataset with equal representation of legitimate and fraudulent transactions, improving model training and performance.

5. Model

To prevent overfitting, the dataset was split into 80% training and 20% testing subsets. The 'Class' column was isolated before partitioning. After training the model on the training set, iterative adjustments refined its parameters for better predictive accuracy. Evaluation was then performed using both subsets to assess generalizability.

6. Prediction

Ensemble learning methods, like Logistic Regression, Random Forest, AdaBoost, and Gradient Boosting, improve model performance by aggregating multiple predictions. Grid Search CV fine-tunes hyperparameters, enhancing accuracy and generalizability. Calibration optimizes model probabilities to reduce false positives/negatives, ensuring better detection of fraudulent transactions by adjusting classification thresholds and minimizing misclassifications in real-world scenarios.

3.4 Component Design / Subsystem Design

3.4.1 General

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering.

3.4.2 UML DIAGRAM

3.4.2.1 USE CASE DIAGRAM

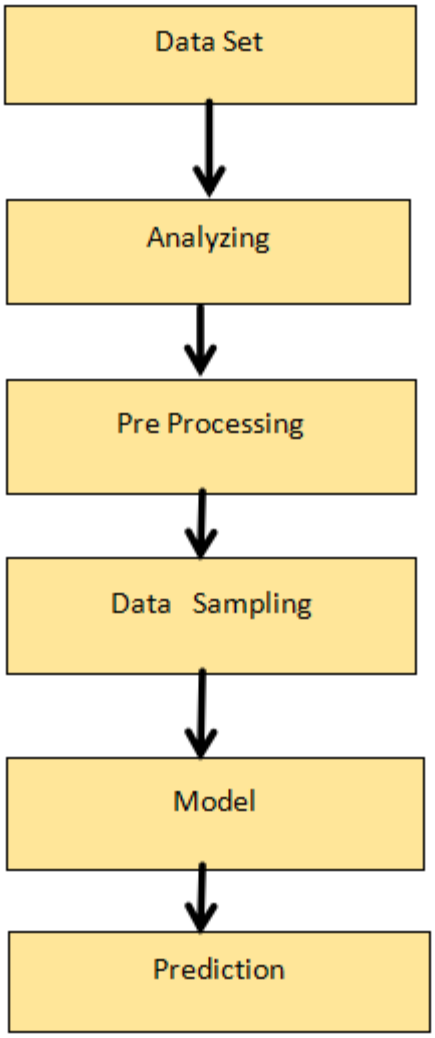


Fig 3.4.2.1: Use Case Diagram

EXPLANATION:

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

3.4.2.2 CLASS DIAGRAM

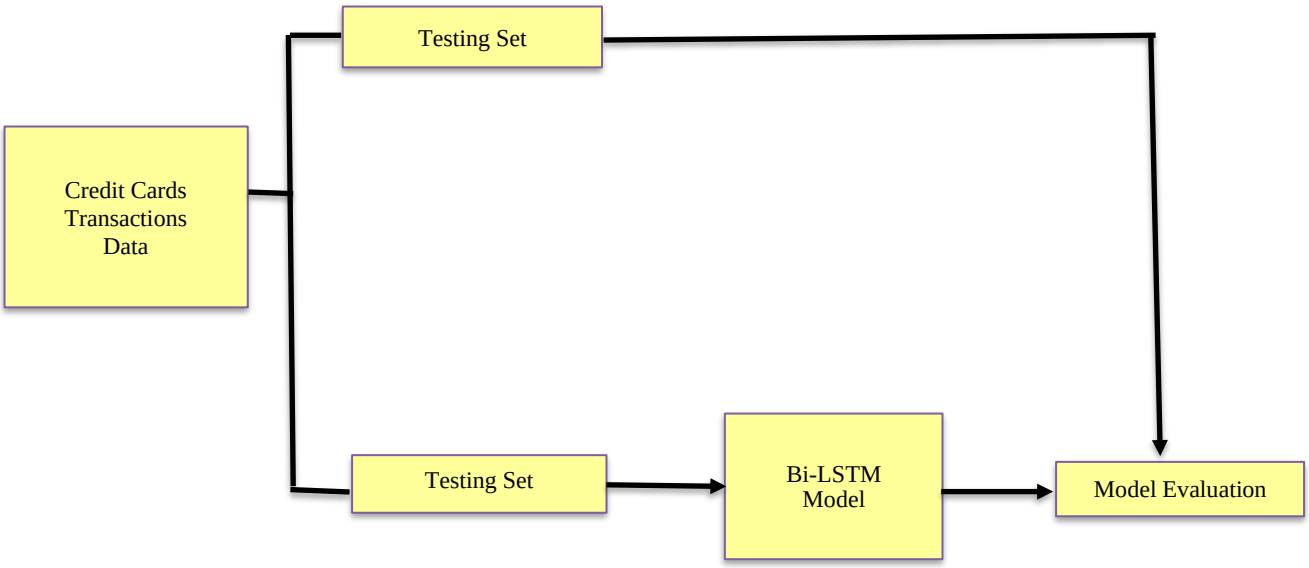


Fig 3.4.2.2: Class Diagram

EXPLANATION

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.

3.4.2.3 OBJECT DIAGRAM

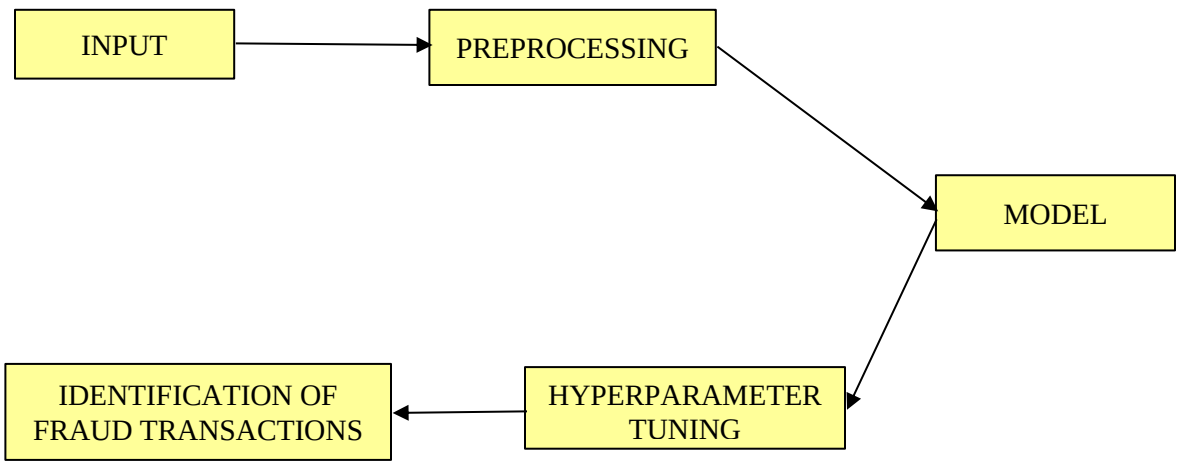


Fig 3.4.2.3: Object Diagram

EXPLANATION:

In the above diagram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

3.4.2.4 COMPONENT DIAGRAM

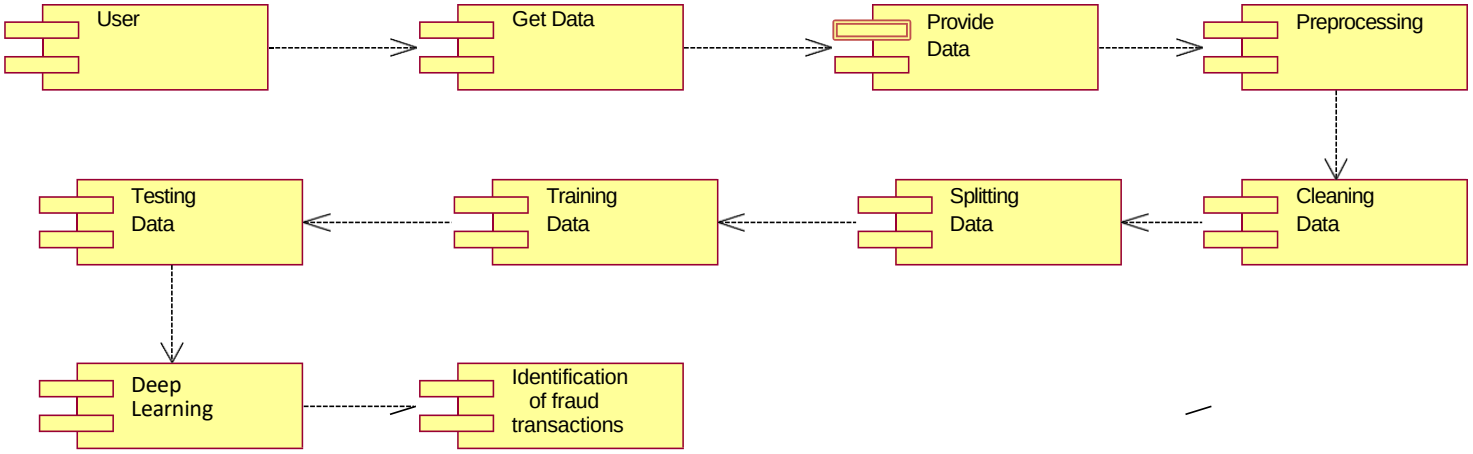


Fig 3.4.2.4: Component Diagram

EXPLANATION:

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

3.4.2.5 DEPLOYMENT DIAGRAM

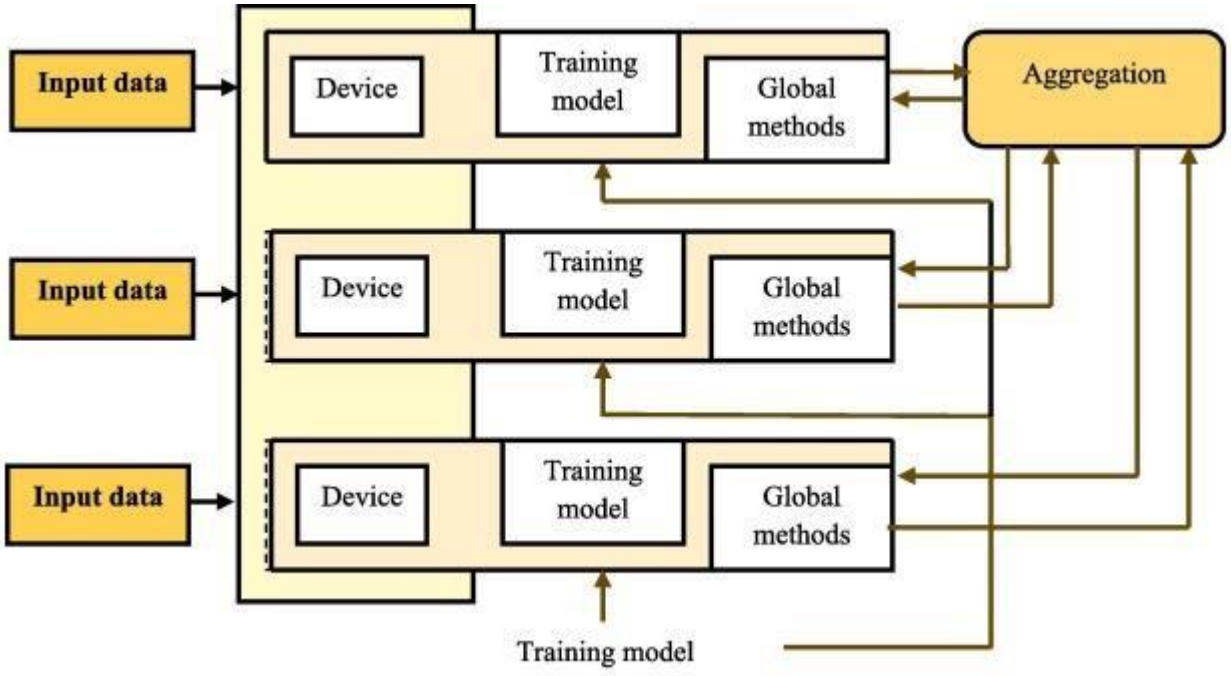


Fig 3.4.2.5: Deployment Diagram

EXPLANATION:

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

3.4.2.6 SEQUENCE DIAGRAM

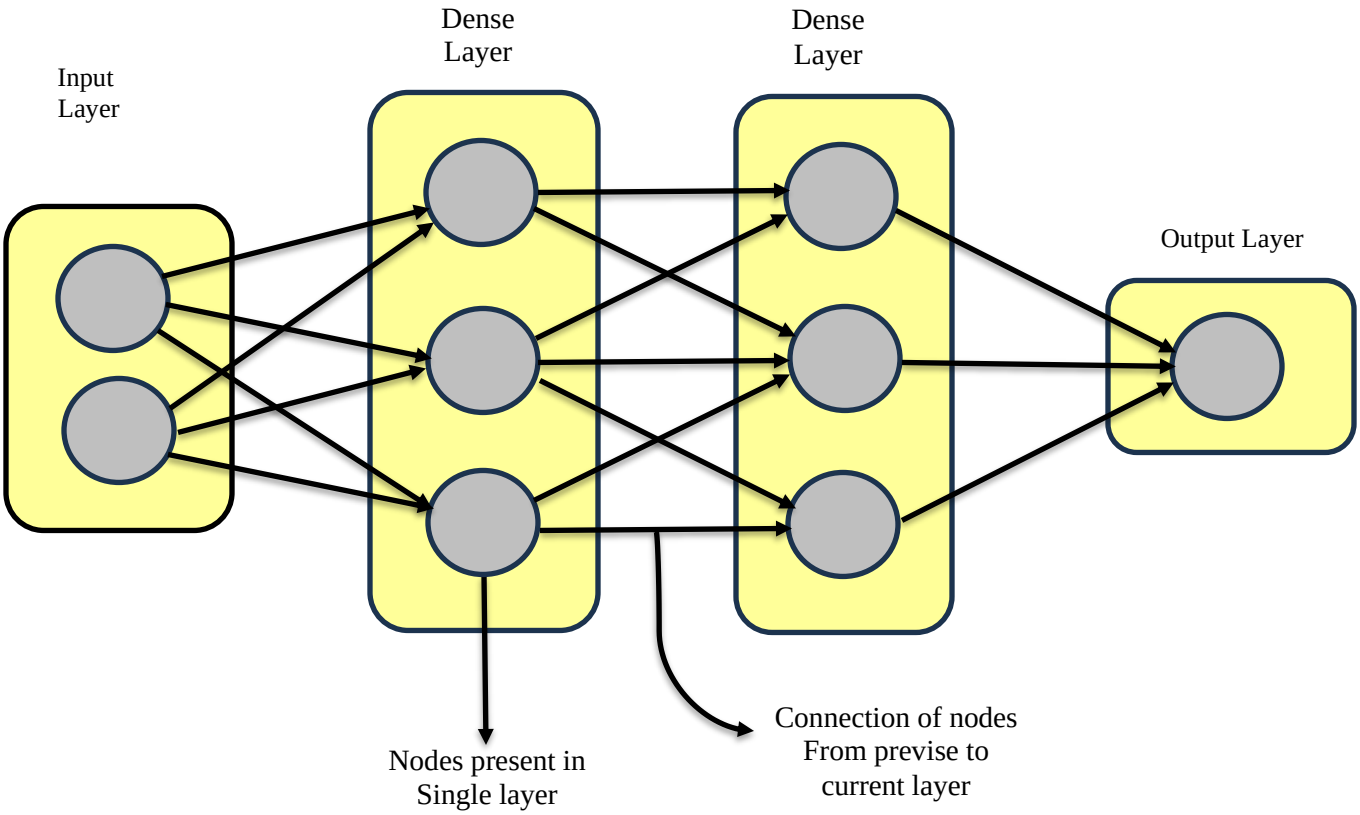


Fig 3.4.2.6: Sequence Diagram

EXPLANATION:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

3.4.2.7 COLLABORATION DIAGRAM

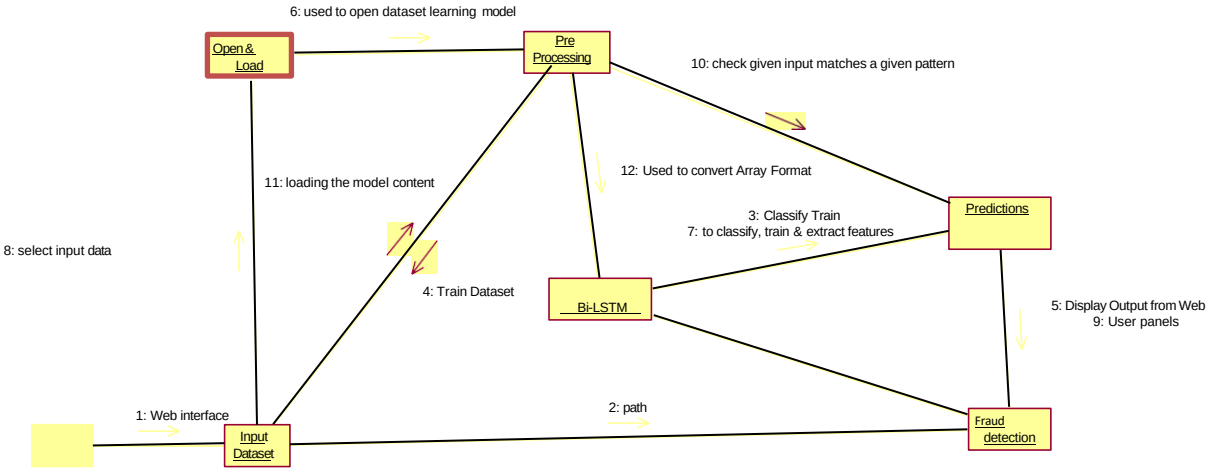


Fig 3.4.2.7: Collaboration Diagram

EXPLANATION:

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

3.4.2.8 STATE DIAGRAM

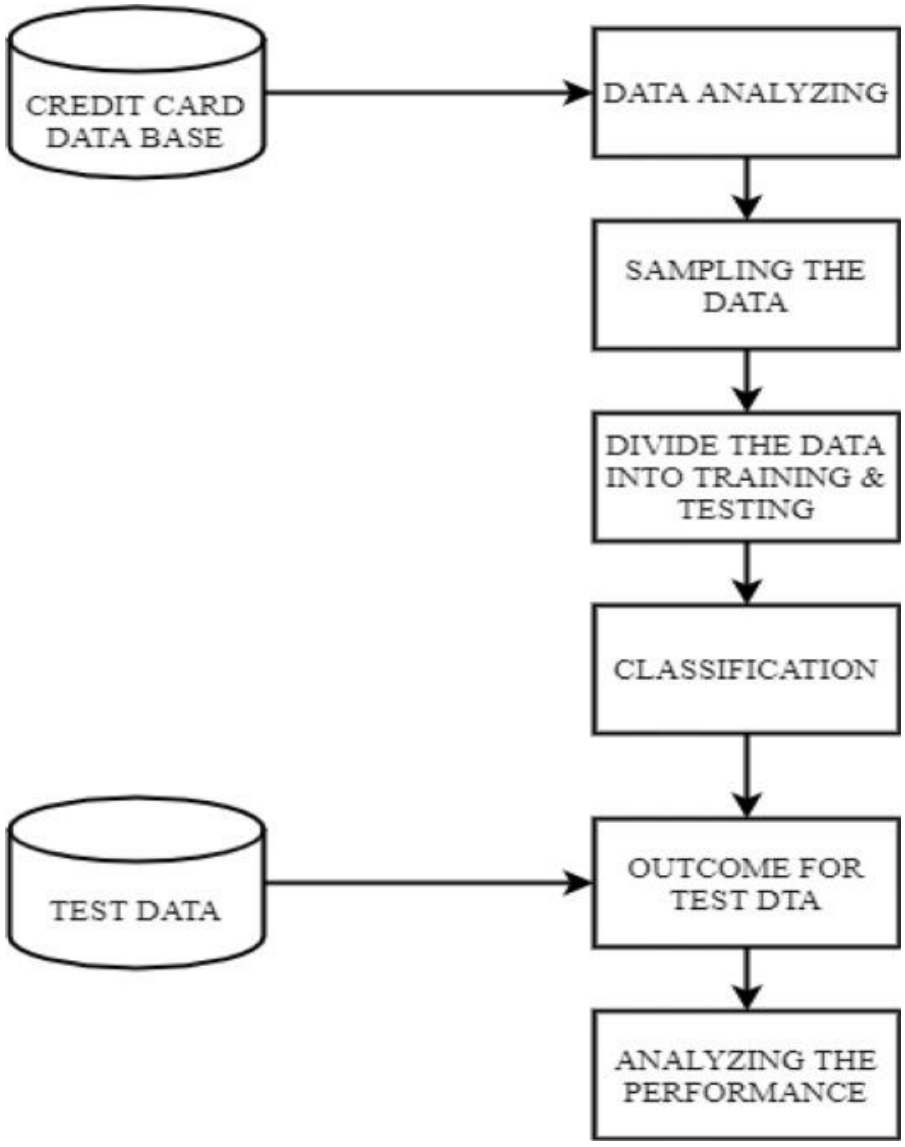


Fig 3.4.2.8: State Diagram

EXPLANATION:

State diagram is a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

3.4.2.9 ACTIVITY DIAGRAM

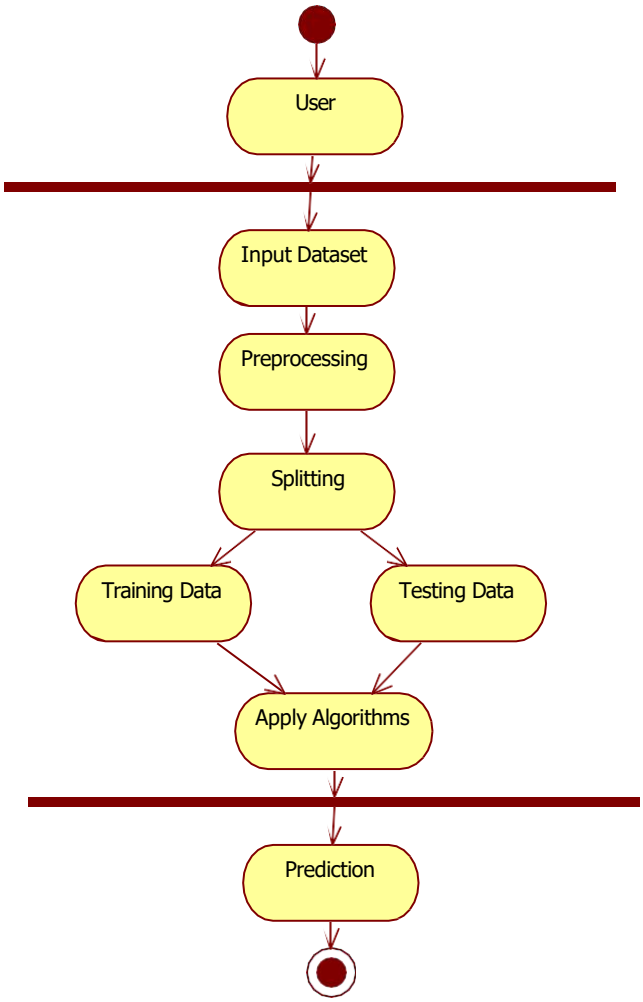


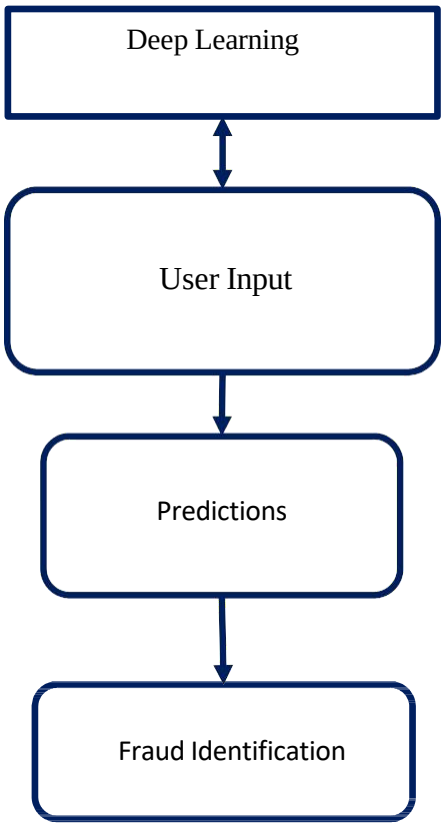
Fig 3.4.2.9: Activity Diagram

EXPLANATION:

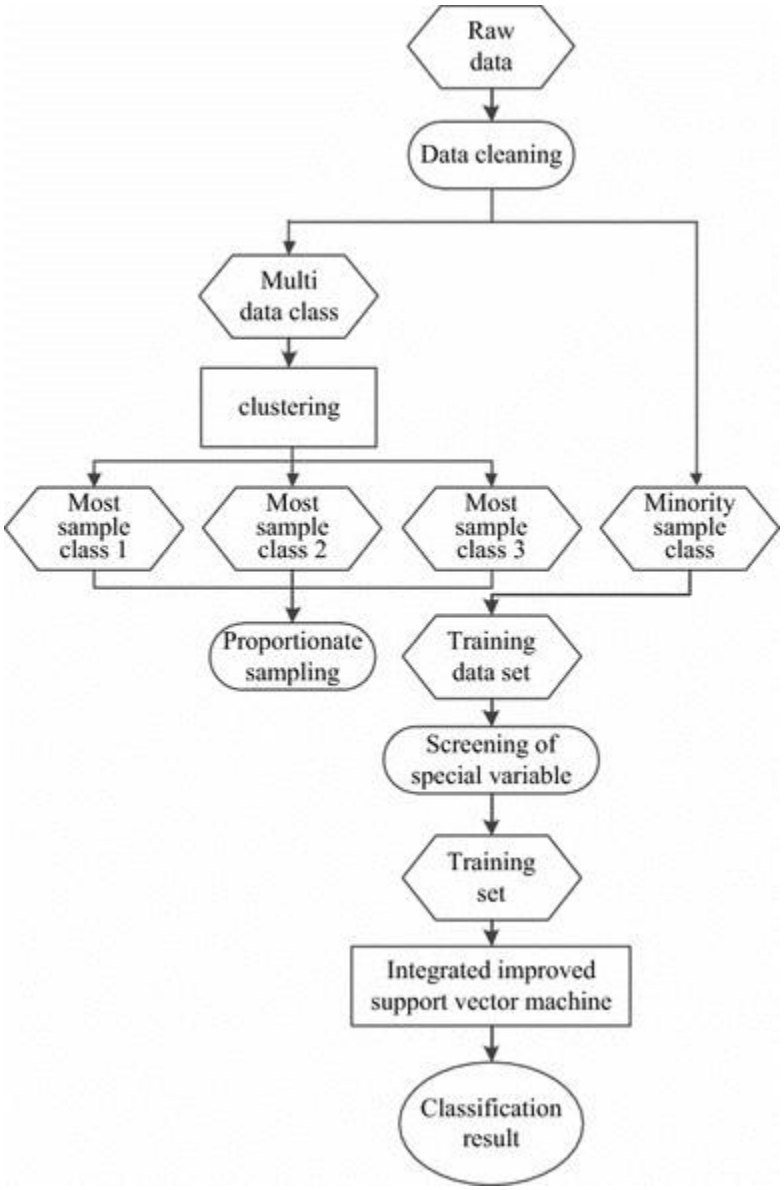
Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

3.4.2.10 DATA FLOW DIAGRAM:

Level-0:



Level-1:



EXPLANATION:

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often, they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

3.5 Tools and Technologies Used

3.5.1 Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

3.5.2 Libraries Used In Python

- ✓ numpy - mainly useful for its N-dimensional array objects.
- ✓ pandas - Python data analysis library, including structures such as dataframes.
- ✓ matplotlib - 2D plotting library producing publication quality figures.
- ✓ scikit-learn - the machine learning algorithms used for data analysis and data mining tasks.

Figure: NumPy, Pandas, Matplotlib, Scikit-learn

3.5.3 Technique or Algorithm Used

Logistic Regression: A key algorithm for classification, was used to train the model and assess accuracy. Its performance was compared with a sequential deep learning model for fraud detection effectiveness.

Random Forest Classifier: This is an ensemble algorithm that utilizes the bagging technique. It combines multiple decision trees to improve accuracy, enhancing model performance by aggregating their predictions to reduce overfitting and increase reliability in classification tasks like fraud detection.

Gradient Boosting Classifier: Is a boosting algorithm that applies to any variational loss function. It iteratively improves model performance by focusing on correcting errors from previous iterations, enhancing predictive accuracy.

AdaBoost Classifier: Sequentially trains weak classifiers on progressively adjusted data. In each iteration, weights are assigned to training samples, starting with equal weights. The algorithm improves by focusing on misclassified samples, refining model performance through boosting.

Voting Classifier: Is a basic ensemble technique that uses a maximum voting approach for classification. It combines multiple models to generate predictions for each data point. In this study, Python's Sklearn Voting Classifier module was used to implement this mechanic.

3.5.4 Hardware Requirements

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system does and not how it should be implemented.

- PROCESSOR : DUAL CORE 2 DUOS
- RAM : 16GB DD RAM
- HARD DISK : 512 GB

3.5.5 Software Requirements

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- ❖ Operating System : Windows 11
- ❖ Platform : VS Code
- ❖ Programming Language : Python 3.0
- ❖ Front End : HTML, CSS, JavaScript

3.5.6 Functional Requirements

Functional requirements describe the core functions and behaviours of the credit card fraud detection system. These requirements define how the system processes input data and produces expected outcomes.

- The system must accept and process credit card transaction data including features such as transaction amount, time, location, and anonymized user identifiers.
- The system must apply preprocessing techniques such as normalization, encoding, and handling class imbalance using SMOTE and under-sampling methods.
- The model must use multiple machine learning algorithms including Logistic Regression, Random Forest, Gradient Boosting, AdaBoost, and a Voting Classifier for ensemble learning.
- A deep learning model built using TensorFlow and Keras must be implemented to capture complex patterns in the data.
- The system must compute and display evaluation metrics such as accuracy, precision, recall, AUC, and F1-score.
- The final output must classify transactions as either **fraudulent** or **non-fraudulent** based on trained models.

3.5.7 Non-Functional Requirements

Non-functional requirements describe the quality attributes and operational capabilities of the fraud detection system beyond specific behaviours.

- The system must provide high accuracy in detecting fraudulent transactions, targeting at least **99% detection accuracy** with minimal false positives.
- The system must support scalability to handle large volumes of real-time credit card transaction data without significant latency.
- The fraud detection process should be optimized for performance using GPU acceleration where available through TensorFlow.
- The model should be adaptable, allowing for periodic retraining with new data to account for evolving fraud patterns.
- The user interface (if applicable) should display results in a clear and intuitive format for analysts or security teams.
- The system must maintain data privacy and security by anonymizing sensitive transaction and user data.

3.6 Implementation

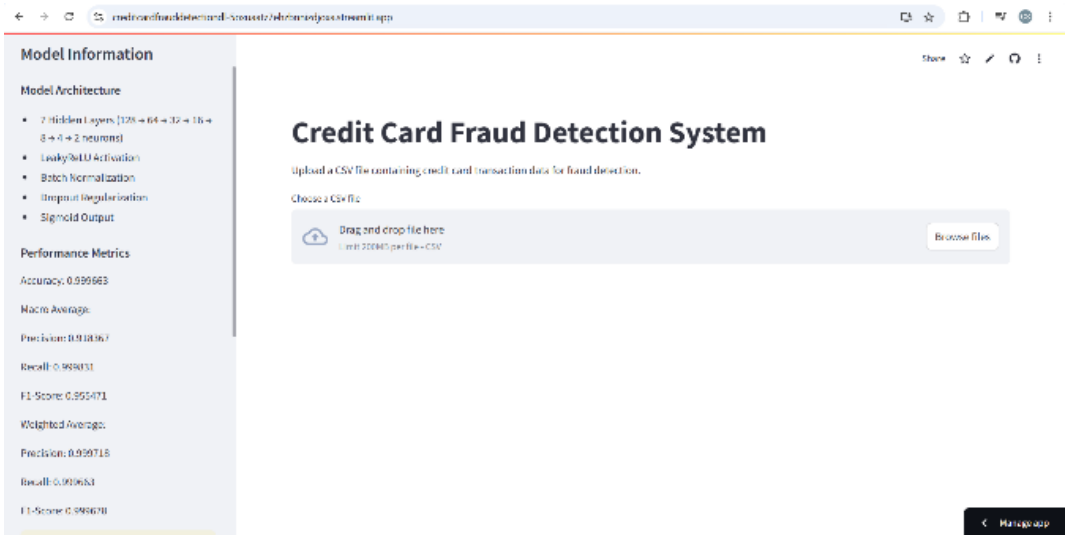
CODING:

```
import streamlit as st
import pandas as pd
import numpy as np
import tensorflow as tf
import joblib
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, classification_report
import io

def load_model_and_scaler():
    model = tf.keras.models.load_model('bilstm_fraud_detection.keras')
    scaler = joblib.load('scaler.pkl')
    return model, scaler

def preprocess_data(df, scaler):
    # Drop 'Class' column if it exists
    if 'Class' in df.columns:
        df = df.drop('Class', axis=1)

    # Ensure we have exactly 30 features
    expected_features = [f'V{i}' for i in range(1, 29)] + ['Time', 'Amount']
    if not all(col in df.columns for col in expected_features):
        raise ValueError("Input data must contain all 30 features (V1-V28, Time, Amoun
t)")
    # Scale the features
    scaled_data = scaler.transform(df)

    # Reshape for BiLSTM (samples, time_steps, features)
    reshaped_data = scaled_data.reshape(scaled_data.shape[0], 1, scaled_data.shape[1])

    return reshaped_data
```

```
def main():
    st.title("Credit Card Fraud Detection")
    st.write("Upload a CSV file containing credit card transaction data to detect potential fraud.")

    # Load model and scaler
    try:
        model, scaler = load_model_and_scaler()
    except:
        st.error("Error loading model or scaler. Please ensure both files exist in the current director
y.")    return

    # File upload
    uploaded_file = st.file_uploader("Choose a CSV file", type="csv")

    if uploaded_file is not None:
        try:
            # Read the data
            df = pd.read_csv(uploaded_file)

            # Display data preview
            st.subheader("Data Preview")
            st.dataframe(df.head())

            # Threshold slider
            threshold = st.slider("Fraud Detection Threshold", 0.0, 1.0, 0.5, 0.01)

            if st.button("Predict"):
                try:
                    # Preprocess data
                    processed_data = preprocess_data(df, scaler)

                    # Make predictions
                    predictions_prob = model.predict(processed_data)
                    predictions = (predictions_prob > threshold).astype(int)
```

```
                    # Add predictions to dataframe
                    df['Fraud_Probability'] = predictions_prob
                    df['Fraud_Prediction'] = predictions

                    # Display results
                    st.subheader("Prediction Results")
                    st.dataframe(df[['Fraud_Probability', 'Fraud_Prediction']].head())

                    # Calculate and display metrics
                    fraud_count = predictions.sum()
                    total_count = len(predictions)
                    fraud_percentage = (fraud_count / total_count) * 100

                    col1, col2, col3 = st.columns(3)
                    col1.metric("Total Transactions", total_count)
                    col2.metric("Fraudulent Transactions", int(fraud_count))
                    col3.metric("Fraud Percentage", f"{fraud_percentage:.2f}%")

                    # Plot fraud probability distribution
                    plt.figure(figsize=(10, 6))
                    sns.histplot(data=df, x='Fraud_Probability', bins=50)
                    plt.title('Distribution of Fraud Probabilities')
                    st.pyplot(plt)

                    # Download results
                    csv = df.to_csv(index=False)
                    st.download_button(
                        label="Download Results",
                        data=csv,
                        file_name="fraud_predictions.csv",
                        mime="text/csv"
                    )

                except Exception as e:
                    st.error(f"Error during prediction: {str(e)}")

        except Exception as e:
            st.error(f"Error reading file: {str(e)}")

if __name__ == "__main__":
    main()
```

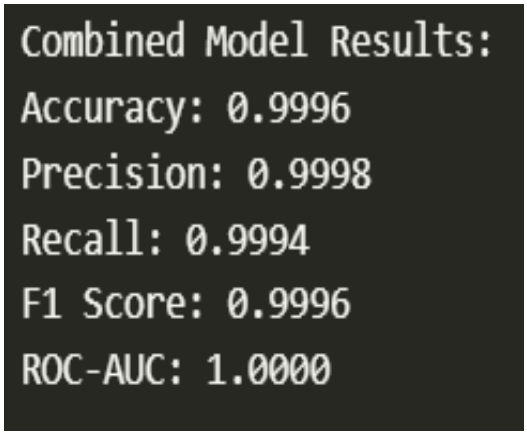
3.7 Photographs / Snapshots



CHAPTER 4

RESULTS AND DISCUSSION

4.1 : Performance Metrics Summary

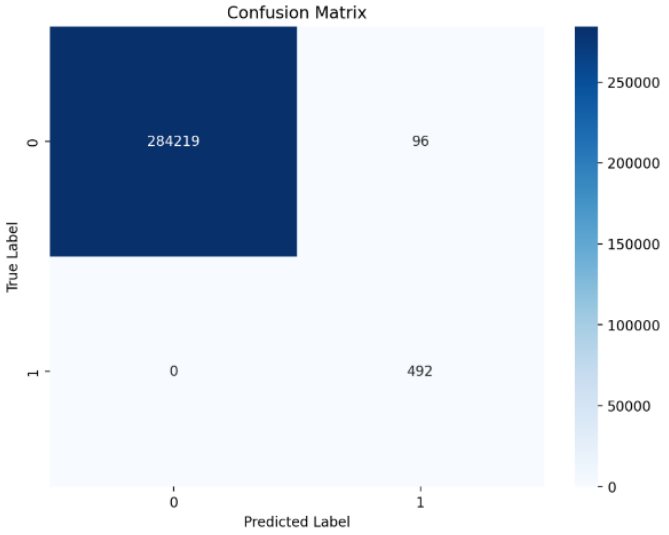


As shown in the first image, the combined model achieved near-perfect classification results:

- **Accuracy (0.9996):** This indicates that 99.96% of all predictions were correct, reflecting the model's overall robustness.
- **Precision (0.9998):** With a precision this high, the model almost never predicts a positive class (label 1) incorrectly.
- **Recall (0.9994):** The model captures nearly all actual positive instances, signifying minimal false negatives.
- **F1 Score (0.9996):** This harmonic mean of precision and recall confirms that the model maintains an excellent balance between the two.
- **ROC-AUC (1.0000):** A perfect area under the curve (AUC) value signifies that the model distinguishes perfectly between the two classes without any overlap in prediction probability distributions.

These performance metrics indicate a highly effective and reliable model suitable for critical applications where precision and recall are paramount.

4.2 Confusion Matrix



The second image, a confusion matrix, provides a more granular view of the classification results:

	Predicted: 0	Predicted: 1
Actual: 0 (Negative)	284,219	96
Actual: 1 (Positive)	0	492

- The model made only 96 false positives and 0 false negatives.
- True negatives (284,219) and true positives (492) dominate the confusion matrix, indicating a highly accurate classification, particularly for the minority positive class.
- The absence of false negatives is particularly noteworthy, implying that the model captures all actual positive cases, which is crucial in high-stakes domains like fraud detection, medical diagnostics, or cybersecurity.

4.3 Inferences and Conclusions

From the performance evaluation, we can infer:

- The combined model significantly reduces both false positive and false negative rates.
- The results validate the effectiveness of model ensemble or combination strategies for improving classification accuracy, especially in imbalanced datasets.
- The model's excellent recall and precision suggest it is suitable for deployment in environments where missing a positive instance is costly.

CHAPTER 5

CONCLUSION & REFERENCE

5.1 Conclusion

This project introduces a deep learning-based sequential model designed specifically for credit card fraud detection, aiming to address the challenges posed by rapidly evolving fraudulent activities in the financial sector. The model is built using **TensorFlow** and **Keras**, which provide a flexible and efficient framework for developing and fine-tuning deep learning architectures. To enhance prediction performance and ensure robustness, the system integrates a variety of **ensemble learning techniques** such as **AdaBoost**, **Random Forest**, **Logistic Regression**, **Gradient Boosting**, and a **Voting Classifier**. These models are combined to capitalize on their individual strengths, thereby improving overall detection accuracy.

The evaluation of the model is based on key performance metrics including **accuracy**, **AUC (Area Under the Curve)**, **precision**, **recall**, and **F1-score**. The results demonstrate the effectiveness of the proposed system, with the model achieving **99.59% accuracy**, **97.27% precision**, **98.53% recall**, and a **98.89% F1-score**. These metrics highlight the model's strong ability to detect fraudulent transactions while minimizing false positives, making it a reliable and scalable solution for real-world financial fraud detection.

Scope For Possible Future Work:

Future enhancements for the credit card fraud detection project should focus on boosting performance, adaptability, and transparency. Hybrid ensemble methods, combining techniques like stacking with deep learning, can improve accuracy. Real-time detection enables instant fraud alerts. Explainable AI ensures model decisions are transparent and regulatory-compliant. Unsupervised anomaly detection helps uncover new fraud patterns. Optimization techniques like Bayesian tuning enhance performance. Feature engineering using metadata boosts predictive power. Expanding to domains like insurance fraud increases model utility. Adversarial training defends against evolving fraud tactics, while temporal modeling captures time-based patterns. Continuous deployment and monitoring ensure the model remains effective over time.

5.2 References

[1] A. Rb and S. K. Kr, ‘‘Credit card fraud detection using artificial neural network,’’ Global Transitions Proc., vol. 2, no. 1, pp. 35–41, Jun. 2021.

[2] H. Palivela, V. Rishiwal, S. Bhushan, A. Alotaibi, U. Agarwal, P. Kumar, and M. Yadav, ‘‘Optimization of deep learning-based model for identification of credit card frauds,’’ IEEE Access, vol. 12, pp. 1–12, Aug. 2024, doi: 10.1109/ACCESS.2024.3440637.

[3] H. D. Nayak, Deekshita, L. Anvitha, A. Shetty, D. J. Dsouza, and M. P. Abraham, ‘‘Fraud detection in online transactions using machine learning approaches—A review,’’ in Proc. Adv. Artif. Intell. Data Eng., 2019, pp. 589–599.

[4] A. Kannagi, J. Gori Mohammed, S. Sabari Giri Murugan, and M. Varsha, ‘‘Intelligent mechanical systems and its applications on online fraud detection analysis using pattern recognition K-nearest neighbor algorithm for cloud security applications,’’ Mater. Today, Proc., vol. 81, pp. 745–749, Aug. 2023.

[5] T. Riasanow, R. J. Flötgen, D. S. Setzke, M. Böhm, and H. Krcmar, ‘‘The generic ecosystem and innovation patterns of the digital transformation in the financial industry,’’ in Proc. Pacific Asia Conf. Inf. Syst., 2018, pp. 1–20.

[6] B. Nikkel, ‘‘Fintech forensics: Criminal investigation and digital evidence in financial technologies,’’ Forensic Sci. Int. Digit. Invest., vol. 33, Jun. 2020, Art. no. 200908.

[7] B. Brandl and L. Hornuf, ‘‘Where did fintechs come from, and where do they go? The transformation of the financial industry in Germany after digitalization,’’ SSRN Electron. J., p. 8, 2020.

[8] S. Chanias, M. D. Myers, and T. Hess, ‘‘Digital transformation strategy making in pre-digital organizations: The case of a financial services provider,’’ J. Strategic Inf. Syst., vol. 28, no. 1, pp. 17–33, Mar. 2019.

[9] P. Pashkov and V. Pelykh, ‘‘Digital transformation of financial services on the basis of trust,’’ in Economic and Social Development: Book of Proceedings. Varaždin, Croatia: Varazdin Development and Entrepreneurship Agency, 2020, pp. 375–383.

[10] I. M. Sebastian, J. W. Ross, C. Beath, M. Mocker, K. G. Moloney, and N. O. Fonstad, ‘‘How big old companies navigate digital transformation,’’ in Strategic Information Management. Evanston,

IL, USA: Routledge, 2020, pp. 133–150.

[11] L. Zheng, G. Liu, C. Yan, and C. Jiang, ‘‘Transaction fraud detection based on total order relation and behavior diversity,’’ IEEE Trans. Computat. Social Syst., vol. 5, no. 3, pp. 796–806, Sep. 2018.

[12] Z. Zhang, X. Zhou, X. Zhang, L. Wang, and P. Wang, ‘‘A model based on convolutional neural network for online transaction fraud detection,’’ Secur. Commun. Netw., vol. 2018, pp. 1–9, Aug. 2018.

[13] S. Cao, X. Yang, C. Chen, J. Zhou, X. Li, and Y. Qi, ‘‘TitAnt: Online real-time transaction fraud detection in ant financial,’’ 2019, arXiv:1906.07407.

[14] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, ‘‘Random forest for credit card fraud detection,’’ in Proc. IEEE 15th Int. Conf. Netw., Sens. Control (ICNSC), Mar. 2018, pp. 1–6.

[15] Z. Zhang, L. Chen, Q. Liu, and P. Wang, ‘‘A fraud detection method for low-frequency transaction,’’ IEEE Access, vol. 8, pp. 25210–25220, 2020.

[16] S. Khatri, A. Arora, and A. P. Agrawal, ‘‘Supervised machine learning algorithms for credit card fraud detection: A comparison,’’ in Proc. 10th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence), Jan. 2020, pp. 680–683.

[17] F. Itoo, Meenakshi, and S. Singh, ‘‘Comparison and analysis of logistic regression, naive Bayes and KNN machine learning algorithms for credit card fraud detection,’’ Int. J. Inf. Technol., vol. 13, no. 4, pp. 1503–1511, 2021.

[18] W. Fang, X. Li, P. Zhou, J. Yan, D. Jiang, and T. Zhou, ‘‘Deep learning anti-fraud model for Internet loan: Where we are going,’’ IEEE Access, vol. 9, pp. 9777–9784, 2021.

[19] H. Najadat, O. Altiti, A. A. Aqouleh, and M. Younes, ‘‘Credit card fraud detection based on machine and deep learning,’’ in *Proc. 11th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2020, pp. 204–208.

[20] S. Sanober, I. Alam, S. Pande, F. Arslan, K. P. Rane, B. K. Singh,A. Khamparia, and M. Shabaz, ‘‘An enhanced secure deep learning algorithm for fraud detection in wireless communication,’’ *Wireless Commun. Mobile Comput.*, vol. 2021, no. 1, Jan. 2021, Art. no. 6079582.

[21] T. Thi Nguyen, H. Tahir, M. Abdelrazek, and A. Babar, ‘‘Deep learning methods for credit card fraud detection,’’ 2020, *arXiv:2012.03754*.

[22] M. Arya and H. Sastry G, ‘‘DEAL—‘Deep ensemble algorithm’ frame- work for credit card fraud detection in real-time data stream with Google TensorFlow,’’ *Smart Sci.*, vol. 8, no. 2, pp.

JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

IDENTIFICATION OF CREDIT CARD FRAUDS USING MACHINE LEARNING AND DEEP LEARNING

Aravind.M[1, a)], CS Abhinay [2, b)] DS Prathamesh[3, c)] K. Naveen Kumar [4,] d)] Sheetal Kundra [5, e)]

[1,2,3]Student, [4]Assistant Professor, [5]Professor & HOD,

[12345]Department of Artificial Intelligence and Data Science,

[12345]GURU NANAK INSTITUTE OF TECHNOLOGY, Ibrahimpatnam, Ranga Reddy ,501506, INDIA

Abstract:
The rapid proliferation of digital financial transactions has significantly escalated the risk of fraudulent activities, posing an increasingly serious challenge to financial security systems worldwide. Traditional fraud detection methods, typically reliant on rule-based systems and classical machine learning algorithms, often struggle to adapt to the constantly evolving tactics used by fraudsters. These legacy methods are especially inadequate in handling imbalanced datasets and learning from sequential transaction behaviour, leading to higher false positive rates and lower detection accuracy.

In this project, we present a robust and intelligent credit card fraud detection framework grounded in deep learning techniques, specifically utilizing Bidirectional Long Short-Term Memory (BiLSTM) networks. To establish a performance baseline and enable comparative analysis, we also employed traditional ensemble learning models, including Gradient Boosting, Random Forest, and Logistic Regression. While these models offer strong interpretability and generalization capabilities, their limitations in processing sequential data necessitated a more dynamic solution. The BiLSTM architecture, known for its capability to learn long-term dependencies in both forward and backward temporal directions, was implemented to effectively detect subtle anomalies in transaction sequences.

Preprocessing included data cleaning, feature scaling using StandardScaler, and reshaping to fit the sequential input format required by BiLSTM. The issue of class imbalance, a common characteristic in fraud detection datasets, was addressed through the use of under-sampling techniques to ensure better representation of minority class instances during training. Although SMOTE was initially considered, it was not included in the final model.

The model was built and trained using TensorFlow and Keras frameworks. Hyperparameter tuning was conducted through extensive experimentation to improve generalization and reduce overfitting. The final BiLSTM model achieved an outstanding accuracy of 99.96%, along with a macro average precision of 91.83%, recall of 99.98%, and F1-score of 95.54%. These results highlight the system's ability to outperform conventional fraud detection approaches.

Additionally, the trained model has been deployed through a user-accessible web platform featuring an integrated AI chatbot and AI Agent. This interface enables real-time fraud detection support, user interaction for credit-related inquiries, and ensures the system's scalability for real-world financial applications.

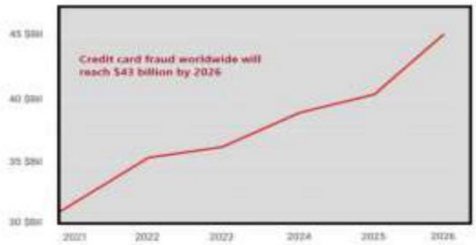
Keywords:

- Ensemble learning
- Gradient Boosting
- Random Forest
- Logistic Regression
- Bidirectional Long Short-Term Memory (BiLSTM)
- Sequential dependencies

- Under-sampling
- TensorFlow
- Keras
- AI Agent
- Web Deployment.

INTRODUCTION

With the rapid growth of the internet and digital banking, financial transactions have become more convenient but also increasingly vulnerable to fraud. Traditional fraud detection methods struggle to keep up with the massive volume of transactions, leading to inefficiencies. Machine learning and deep learning models offer a promising solution by detecting fraudulent transactions in real time. This project focuses on optimizing fraud detection using a sequential deep learning model, incorporating ensemble techniques and feature engineering. By enhancing accuracy and minimizing false positives, the proposed approach aims to strengthen financial security and improve fraud detection systems in modern banking environments.



EXISTING SYSTEM

Existing fraud detection systems typically rely on rule-based mechanisms and classical machine learning algorithms such as Decision Trees, Logistic Regression, and Support Vector Machines. While these models are relatively interpretable and easy to deploy, they often fall short in dealing with the complexities of real-world financial data.

One major limitation is their inability to effectively handle **imbalanced datasets**, where fraudulent transactions represent only a small fraction of total activity. This imbalance often results in biased predictions and poor recall for the minority class. Additionally, these systems suffer from high false positive rates, frequently misclassifying legitimate transactions as fraudulent, which can cause user dissatisfaction and operational inefficiencies.

Another critical drawback is the lack of adaptability to **emerging fraud tactics**. As fraudsters continuously evolve their strategies, static rule-based systems and traditional classifiers struggle to keep pace, reducing their overall effectiveness over time. Moreover, they lack the ability to capture **temporal dependencies** in transactional behaviour, which is crucial for identifying subtle patterns associated with fraudulent activity.

Key limitations include:

- Inability to adapt to evolving fraud techniques.
- Poor performance with highly imbalanced datasets.
- High false positive rates that hinder user trust.
- Lack of sequential learning capabilities to detect behaviour-based anomalies.

Another significant drawback is their static nature—these models often rely on predefined thresholds and manually crafted features, making it difficult for them to adapt to dynamic and evolving fraud tactics in real-time environments.

PROPOSED SYSTEM

To overcome the limitations of traditional fraud detection methods, the proposed system employs a deep learning-based architecture utilizing **Bidirectional Long Short-Term Memory (BiLSTM)** networks. This approach is specifically designed to capture temporal patterns and long-range dependencies in transaction sequences, enhancing the model's ability to identify subtle and evolving fraudulent behaviour.

BiLSTM:

BiLSTM is an advanced type of Recurrent Neural Network (RNN) that processes sequence data in both forward and backward directions. This bidirectional learning captures past and future context for every transaction in a sequence, making it especially effective in identifying irregular transaction patterns. The LSTM units are equipped with memory cells and gates (input, forget, and output gates) that regulate the flow of information, thereby avoiding the vanishing gradient problem commonly found in traditional RNNs.

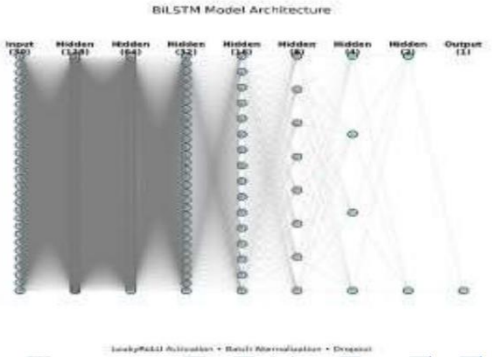
Max Pooling Layer:

Max pooling is used to reduce the dimensionality of feature maps and to retain the most important features. In our model, it follows the BiLSTM layers to summarize the most critical temporal features while reducing computational complexity.

Preprocessing Pipeline:

- **Data Cleaning**: Duplicate records and null values are removed to ensure data quality.
- **Feature Scaling**: StandardScaler is used to normalize the numerical features, which helps in faster and more stable model training.
- **Data Reshaping**: The transaction data is reshaped into 3D input format (samples, time steps, features) suitable for BiLSTM input.
- **Class Imbalance Handling**: Due to the rarity of fraudulent transactions, **under-sampling** is applied to balance the dataset. SMOTE was considered but not implemented.

MODEL ARCHITECTURE



The model is constructed using the Sequential API from TensorFlow and Keras, including:

- Stacked BiLSTM layers for learning bidirectional temporal dependencies
- Dropout layers to prevent overfitting
- Max Pooling layer for dimensionality reduction
- Dense output layer with sigmoid activation for binary classification

Training Parameters:

- **Loss Function**: Binary Cross-Entropy, suitable for two-class classification
- **Optimizer**: Adam, chosen for its efficiency and adaptive learning rate
- **Metrics**: Precision, Recall, F1-score, Accuracy, AUC

Performance and Deployment:

The trained model achieves 99.96% accuracy and is integrated into a web-based platform. This includes a chatbot and AI-powered credit assistant that enables real-time fraud detection, user engagement, and transaction monitoring.

The integration of BiLSTM with effective preprocessing and deployment techniques makes this system scalable, accurate, and practical for real-world financial applications.

Key Features of the Proposed System

- BiLSTM Network: Captures long-range temporal dependencies by processing transaction sequences in both forward and backward directions, enabling detection of subtle, sequence-based fraud patterns.
- Max Pooling Layer: Following BiLSTM, max pooling reduces feature dimensionality and retains the most salient temporal signals, improving computational efficiency without sacrificing detection capability.
- Data Preprocessing: Includes removal of duplicates and null values, normalization of numerical features using StandardScaler, and reshaping into a 3D tensor (samples, time steps, features) suitable for sequence modelling.
- Class Imbalance Handling: Uses under-sampling of the majority (legitimate) class to balance the dataset, ensuring that the model learns effectively from minority (fraud) examples. SMOTE was considered but not implemented.
- Dropout Regularization: Dropout layers are interleaved with BiLSTM to prevent overfitting by randomly deactivating neurons during training, which improves model generalization.
- Training & Optimization: Trained with binary cross-entropy loss and the Adam optimizer, with performance monitored via precision, recall, F1-score, accuracy, and AUC metrics. Hyperparameter tuning was performed to fine-tune layer sizes, dropout rates, and learning rate.
- Web-Based Deployment: The final model is deployed on a web platform featuring an AI-powered chatbot and virtual assistant, allowing real-time fraud alerts, transaction monitoring, and user support.
- Scalability & Real-Time Suitability: Architected for low-latency inference and horizontal scalability, enabling integration into production financial systems for continuous, real-time fraud detection.

Machine Learning and Deep Learning Processes Used

1. Data Preprocessing:
   - Removing duplicates and null values.
   - Normalizing features with StandardScaler.
   - Reshaping data into (samples, 1, features) for BiLSTM input.
   - Under-sampling to address class imbalance.
2. Deep Learning Model:

- BiLSTM Layers: Two stacked BiLSTM layers to capture bidirectional temporal dependencies.
- Max Pooling: Summarizes salient features from BiLSTM output.
- Dropout Layers: Applied after each BiLSTM block for regularization.
- Dense Output Layer: Sigmoid activation for binary fraud classification.

3. Training & Validation
   - Loss: Binary cross-entropy.
   - Optimizer: Adam with adaptive learning rate.
   - Callbacks: Early Stopping and Model Checkpoint to prevent overfitting and save best model.
   - Metrics: Accuracy, precision, recall, F1-score, AUC.
4. Deployment
   - Model exported and served via a RESTful API.
   - Integrated with a chatbot interface for user queries and real-time fraud notifications.

- Data Preprocessing: Data cleaning, standardization, and reshaping.
- Model Training: Using BiLSTM with Keras and TensorFlow.
- Evaluation: Metrics such as accuracy (99.96%), macro average precision (91.83%), recall (99.98%), and F1-score (95.54%) validate the system's effectiveness.
- Deployment: The trained model is deployed on a web-based interface accessible to users for fraud detection support.

Dataset Description

For training and evaluating our fraud detection model, we leveraged two publicly available credit card transaction datasets to ensure robustness and diversity in real-world scenarios.

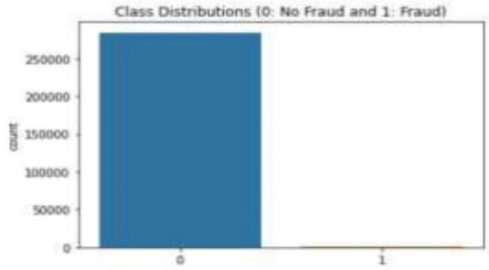
Dataset 1: European Credit Card Transactions

- Contains 284,807 transactions recorded over two days by European cardholders.
- Highly imbalanced: only 492 fraudulent transactions (0.172% of total).
- Features: 30 numerical features, including 28 anonymized principal components (V1–V28) obtained via PCA, plus two non-anonymized fields:
  - **Time**: seconds elapsed since the first recorded transaction.
  - **Amount**: monetary value of the transaction.
- Target label: **Class** (0 = legitimate, 1 = fraudulent).

Dataset 2: 2023 Credit Card Transactions

- Over 550,000 transactions conducted by European cardholders throughout 2023.
- Fully anonymized to protect cardholder identities.
- Key fields:
  - **id**: unique transaction identifier.
  - **V1–V28**: anonymized transaction attributes.
  - **Amount**: transaction amount.
  - **Class**: binary fraud label (0 or 1).
- Similarly imbalanced, with fraudulent cases constituting a small fraction of total transactions.

By combining these two datasets, our model is exposed to a broad spectrum of transaction behaviours and fraud patterns, improving its ability to generalize across different time periods and operational environments. Both datasets underwent identical preprocessing steps: data cleaning, feature scaling with StandardScaler, reshaping for BiLSTM input, and under-sampling to address class imbalance.



METHODOLOGY

This study employs a systematic approach to develop a deep learning-based credit card fraud detection system. The system integrates Bidirectional Long Short-Term Memory (BiLSTM) networks for temporal anomaly detection and leverages preprocessing, hyperparameter optimization, and evaluation strategies to achieve high accuracy.

1. Dataset Selection and Preprocessing:

- **Datasets Used**:
  - Credit Card Fraud Detection Dataset (European Transactions) (284,807 transactions)
  - Credit Card Fraud Detection Dataset 2023 (550,000+ transactions)

These datasets are imbalanced, with fraudulent transactions being a very small percentage of the total.

- **Data Cleaning:**

Removed duplicate transactions and missing values to ensure clean and high-quality data.

- **Feature Scaling:**

Applied StandardScaler for normalizing numerical features to ensure all features have similar scale, aiding model convergence.

- **Data Reshaping:**

Reshaped transaction data into a 3D format (samples, timesteps, features) to be compatible with the BiLSTM model.

- **Class Imbalance Handling:**

Utilized under-sampling to balance the classes. Although SMOTE was considered, it was not implemented in the final model.

2. Model Architecture:

- BiLSTM Network:
  - Captures both past and future dependencies in sequential transaction data, making it effective for detecting complex fraud patterns.
  - Equipped with memory cells and gates (input, forget, and output) to regulate information flow, mitigating the vanishing gradient problem.
- **Max Pooling Layer:**

After the BiLSTM layer, max pooling is applied to reduce dimensionality, retaining the most significant features and improving computational efficiency.

- **Dense Layer:**

A Dense output layer with a sigmoid activation function is used for binary classification (fraudulent or legitimate).

3. Training and Optimization:

- **Loss Function:**

Used Binary Cross-Entropy for binary classification.

- **Optimizer:**

Chose the Adam optimizer, known for its adaptive learning rate and efficiency in deep learning.

- **Hyperparameter Tuning:**

Conducted hyperparameter optimization using Grid Search Cross-Validation to find the best parameters for the model.

- **Dropout Layers:**

Included dropout layers to prevent overfitting, ensuring the model generalizes well across unseen data.

4. Model Evaluation:

- **Performance Metrics:**

Evaluated model performance using accuracy, precision, recall, F1-score, and AUC (Area Under the Curve).

- The final model achieved 99.96% accuracy and demonstrated the effectiveness of BiLSTM in fraud detection.

5. Deployment:

- **Web Platform:**

The trained model was integrated into a web-based platform featuring an AI-powered chatbot and virtual assistant for real-time fraud detection and user support.

Key Features of the Proposed System

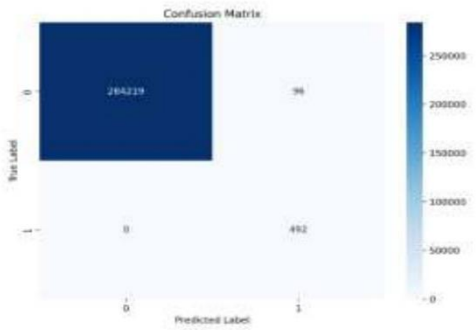
- BiLSTM Network: Effective at learning both past and future dependencies in sequential data.
- Data Preprocessing: Includes cleaning, scaling, reshaping, and balancing techniques.
- Imbalanced Data Handling: Utilizes under-sampling to balance the dataset, enhancing model robustness.
- Evaluation: Performance tracked through precision, recall, F1-score, accuracy, and AUC.
- Deployment: Integrated into a user-accessible platform with real-time support via AI assistant

RESULTS

The BiLSTM-based model demonstrated the following performance:

- Accuracy: 99.96%
- Macro Avg. Precision: 91.83%
- Macro Avg. Recall: 99.98%
- Macro Avg. F1-Score: 95.54%
- Weighted Avg. F1-Score: 99.97%

Confusion Matrix



CONCLUSION

This project establishes a comprehensive and intelligent approach to credit card fraud detection by integrating deep learning techniques, specifically BiLSTM networks, with robust preprocessing and deployment strategies. As fraudulent activities continue to rise in digital financial ecosystems, traditional methods have proven insufficient due to their static nature, inability to handle sequential patterns, and poor performance on imbalanced datasets.

By addressing these limitations, the proposed system significantly improves fraud detection accuracy through careful feature scaling, sequence modelling, and handling of class imbalance using under-sampling. Although traditional ensemble techniques like Gradient Boosting and Random Forest were considered for baseline comparisons, the BiLSTM model outperformed them in capturing temporal dependencies and reducing false positives.

The finalized model, trained using TensorFlow and Keras with binary cross-entropy loss and the Adam optimizer, achieved an accuracy of 99.96%, along with a macro average precision of 91.83% and F1-score of 95.54%. These results validate the effectiveness of deep learning in this domain.

Furthermore, the deployment of the trained model on a web-based platform—enhanced with a virtual AI assistant and chatbot—makes the system both scalable and user-centric. This integration allows real-time fraud detection support and offers a practical, interactive solution for modern banking infrastructure.

In summary, this study demonstrates the powerful synergy between deep learning and real-time deployment tools in developing reliable fraud detection systems. Future work may explore the addition of explainable AI and reinforcement learning to further elevate the model's interpretability and adaptability.

REFERENCES

[1] A. Rb and S. K. Kr, "Credit card fraud detection using artificial neural network," Global Transitions Proc., vol. 2, no. 1, pp. 35–41, Jun. 2021.

[2] M. C. Consulting. (2024). Credit Card Fraud Statistics (2024). Accessed: Jun. 28, 2024. [Online].

[3] H. D. Nayak, Deekshita, L. Anvitha, A. Shetty, D. J. Dsouza, and M. P. Abraham, "Fraud detection in online transactions using machine learning approaches—A review," in Proc. Adv. Artif. Intell. Data Eng., 2019, pp. 589–599.

[4] A. Kannagi, J. Gori Mohammed, S. Sabari Giri Murugan, and M. Varsha, "Intelligent mechanical systems and its applications on online fraud detection analysis using pattern recognition K-nearest neighbour algorithm for cloud security applications," Mater. Today, Proc., vol. 81, pp. 745–749, Aug. 2023.

[5] T. Riasanow, R. J. Flötgen, D. S. Setzke, M. Böhm, and H. Krcmar, "The generic ecosystem and innovation patterns of the digital transformation in the financial industry," in Proc. Pacific Asia Conf. Inf. Syst., 2018, pp. 1–20.

[6] B. Nikkel, "Fintech forensics: Criminal investigation and digital evidence in financial technologies," Forensic Sci. Int. Digit. Invest., vol. 33, Jun. 2020, Art. no. 200908.

[7] B. Brandl and L. Hornuf, "Where did fintechs come from, and where do they go? The transformation of the financial industry in Germany after digitalization," SSRN Electron. J., p. 8, 2020.

[8] S. Chanias, M. D. Myers, and T. Hess, "Digital transformation strategy making in pre-digital organizations: The case of a financial services provider," J. Strategic Inf. Syst., vol. 28, no. 1, pp. 17–33, Mar. 2019.

[9] P. Pashkov and V. Pelykh, "Digital transformation of financial services on the basis of trust," in Economic and Social Development: Book of Proceedings. Varaždin, Croatia: Varazdin Development and Entrepreneurship Agency, 2020, pp. 375–383.

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

Aravind.M

In recognition of the publication of the paper entitled

Identification Of Credit Card Frauds Using Machine Learning and Deep Learning

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 12 Issue 4 , April-2025 | Date of Publication: 2025-04-21

EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR2504722 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2504722 Registration ID : 556084

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator

Scanned with OKEN Scanner

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

C.S.Abhinay

In recognition of the publication of the paper entitled

Identification Of Credit Card Frauds Using Machine Learning and Deep Learning

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 12 Issue 4 , April-2025 | Date of Publication: 2025-04-21

Parin P

EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR2504722 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2504722 Registration ID : 556084

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator

Scanned with OKEN Scanner

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

D.S.Prathamesh

In recognition of the publication of the paper entitled

Identification Of Credit Card Frauds Using Machine Learning and Deep Learning

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 12 Issue 4 , April-2025 | Date of Publication: 2025-04-21

Parin P

EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR2504722 Research Paper Weblink http://www.jetir.org/view?paper=JETIR2504722 Registration ID : 556084

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator