A Project Report on

COMPARATIVE STUDY OF MACHINE LEARNING ALORITHMS FOR FRAUD DETECTION IN BLOCKCHAIN

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

In

Computer Science and Engineering

Submitted by

Kumbala Pavan Reddy
(20H51A0515)

Mamidi Varun
(20H51A0516)

Thalari Nihith Novah
(20H51A0553)

Under the esteemed guidance of

Ms. K. Ragini
(Assistant Professor of CSE)

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A^+ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2020- 2024

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project report entitled **"Comparative Study of Machine Learning Algorithms for Fraud Detection in Blockchain"** being submitted by Kumbala Pavan Reddy (20H51A0515), Mamidi Varun (20H51A0516), Thalari Nihith Novah (20H51A0553) in partial fulfillment for the award of **Bachelor of Technology in Computer Science And Engineering** is a record of bonafide work carried out under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Ms. K. Ragini
Assistant Professor
Dept. Of CSE

Dr. Siva Skandha Sanagala
Associate Professor and HOD
Dept. of CSE

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express our heartfelt gratitude to all the people who helped in making this project a grand success.

We are grateful to **Ms. K. Ragini, Assistant Professor**, Department of Computer Science and Engineering for her valuable technical suggestions and guidance during the execution of this project work.

We would like to thank**, Dr. Siva Skandha Sanagala,** Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete our project work successfully.

We are very grateful to **Dr. Ghanta Devadasu**, Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Computer Science and Engineering for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary& Correspondent, CMR Group of Institutions, and **Shri Ch. Abhinav Reddy**, CEO, CMR Group of Institutions for their continuous care and support.

Finally, we extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly or indirectly in completion of this project work.

K. Pavan Reddy 20H51A0515
M. Varun 20H51A0516
T. Nihith Novah 20H51A0553

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	LIST OF FIGURES	iii
	LIST OF TABLE	iv
	ABSTRACT	v
1	**INTRODUCTION**	**1**
	1.1 Problem Statement	2
	1.2 Research Objective	3
	1.3 Project Scope and Limitations	4
2	**BACKGROUND WORK**	**5**
	2.1. A Machine Learning and Blockchain Based Efficient Fraud Detection Mechanism	6
	2.1.1.Introduction	6
	2.1.2.Merits,Demerits and Challenges	7
	2.1.3.Implementation of A Machine Learning and Blockchain Based Efficient Fraud Detection Mechanism	8
	2.2. Fraud Detection: A Review on Blockchain	15
	2.2.1.Introduction	15
	2.2.2.Merits,Demerits and Challenges	16
	2.2.3.Implementation of Fraud Detection: A Review on Blockchain	17
	2.3. Analysis of Fraud Detection in Blockchain system using Machine LearningAlgorithms	18
	2.3.1.Introduction	18
	2.3.2.Merits,Demerits and Challenges	19
	2.3.3.Implementation of Analysis of Fraud Detection in Blockchain system using Machine LearningAlgorithms	20

3	**PROPOSED SYSTEM**	**21**
	3.1. Objective of Proposed Model	22
	3.2. Algorithms Used for Proposed Model	23
	3.3. Designing	25
	3.3.1.UML Diagram	25
	3.3. Stepwise Implementation and Code	28
	3.4. Model Architecture	37
4	**RESULTS AND DISCUSSION**	**38**
	4.1. Output Screens	39
	4.2. Performance metrics	42
5	**CONCLUSION**	**43**
	5.1 Conclusion and Future Enhancement	44
	REFERENCES	**45**
	GITHUB LINK	**47**
	PUBLICATION PAPER	
	PUBLICATION CERTIFICATES	

List of Figures

FIGURE NO.	TITLE	PAGE NO.
2.1.3.1	Imbalanced data .	8
2.1.3.2	Balanced data	8
2.1.3.3	Logloss of XGboost	9
2.1.3.4	Correlation with class Fraudulent or not	10
2.1.3.5	Classification error of Xgboost	11
2.1.3.6	Precision of RF	11
2.1.3.7	Accuracy of Xgboost	11
2.1.3.8	Confusion matrix with random forest	12
2.1.3.9	Accuracy of random forest	13
2.1.3.10	Transactions published and stored on Blockchain	13
2.3.3.1	Preprocessing steps	20
3.3.1.1	Use Case Diagram	25
3.3.1.2	Class Diagram	26
3.3.1.3	Sequence Diagram	27
3.4.1	Model Architecture	37
4.1.1	Output screen	39
4.1.2	Dataset before preprocessing	39
4.1.3	Splitting of dataset	40
4.1.4	Performance of first four algorithms	40
4.1.5	Performance of other four algorithms	41
4.1.6	Graphical representation of performance	41
4.1.7	Results of test data	42

List of Table

FIGURE NO.	TITLE	PAGE NO.
4.2.1	Tabular format representation of performance of algorithms	42

ABSTRACT

The economy and trust in a blockchain network are significantly impacted by fraudulent transactions. Consensus methods like proof of work or proof of stake can confirm a transaction's validity, but they cannot confirm the identity of the persons that participated in the transaction or verified it. A blockchain network is still susceptible to fraud because of this. Use of machine learning algorithms is one method for eradicating fraud. The existence of fraudulent exchanges in the economy discourages investors from investing in bitcoin and other blockchain-based businesses. False exchanges are regularly viewed with scepticism due to the gatherings in issue or the way they are put up. To prevent them from jeopardizing the trustworthiness of the neighborhood and the blockchain network, people endeavor to identify false exchanges wherever possible.

Numerous other Machine Learning approaches have been suggested to address this issue, but none of them has clearly emerged as the best one, even though some of the results show promise. This study looks at how well a few machine learning and a few deep learning models do at spotting bogus transactions in a blockchain network. Our goal is to pinpoint the clients and transactions that will probably resort to extortion. The machine learning techniques train the dataset based on the fraudulent and integrated transaction patterns and predict the new incoming transactions. The blockchain technology is integrated with machine learning algorithms to detect fraudulent transactions in the Bitcoin network.

CHAPTER 1
INTRODUCTION

CHAPTER 1
INTRODUCTION

1.1.Problem Statement

The issue of identifying fraudulent transactions has long been researched. The economy suffers from fraudulent transactions, which also make individuals less likely to buy bitcoins or have faith in other blockchain based products. Fraudulent transactions are frequently suspect, either because of the parties involved or because of the way they are structured. To keep fraudulent transactions from jeopardizing the community and integrity of the blockchain network, members of a blockchain network strive to identify them as quickly as feasible. Such comparison research will assist in selecting the optimum algorithm based on the trade-off between accuracy and computing speed. Blockchain transactions are constant because once they are recorded, they cannot be changed or reversed. Before a "block" of transactions is added to the blockchain, network clients must agree on the validity of each transaction.

The Fraudulent transactions are harmful to the economy and discourage people from investing in bitcoins or even trusting other blockchain-based solutions. Fraudulent transactions are usually suspicious either in terms of participants involved in the transaction or the nature of the transaction. Members of a blockchain network want to detect Fraudulent transactions as soon as possible to prevent them from harming the blockchain network’s community. Many Machine Learning techniques have been proposed to deal with this problem, some results appear to be quite promising, but there is no obvious superior method.

1.2.Research Objective

The objective of the project is to create sophisticated machine learning models for analyzing and predicting the data set utilizing typical machine learning methods, statistics, and calculus to forecast the frequency and volume of fraudulent and legitimate transactions. We also provide an extensive comparative study of various supervised machine learning techniques like decision trees, Naive Bayes, logistic regression, multilayer perceptron, and so on for the above task.

To compare the performance of various supervised machine learning models like SVM, Decision Tree, Naive Bayes, Logistic Regression, and few deep learning models in detecting fraudulent transactions in a blockchain network. Such comparative study will help decide the best algorithm based on accuracy and computational speed trade-off. Our goal is to find which algorithm gives the best accuracy and also to find the whether the transaction is fraudulent transaction or legitimate.

1.3.Project Scope and Limitations

The scope of the project is to develop advanced machine learning models that utilize typical machine learning methods, statistics, and calculus to analyze and predict the frequency and volume of both fraudulent and legitimate transactions within a blockchain network. The primary scope is to create predictive models capable of identifying and distinguishing between fraudulent and legitimate transactions. The project aims to achieve this through the application of various supervised machine learning techniques such as Support Vector Machines (SVM), Decision Trees, Naive Bayes, Logistic Regression, and deep learning models.

The comparative study of these models will provide valuable insights into their performance in detecting fraudulent transactions. By assessing accuracy and computational speed trade-offs, the project seeks to determine which algorithm is most suitable for this specific task. Additionally, the project intends to identify transactions with the transactions that are involvement in fraudulent activities, which can be valuable for fraud detection and prevention in a blockchain network.

However, there are certain limitations to consider in this project. Firstly, the quality and quantity of the available data will significantly impact the model's performance, and obtaining a comprehensive and reliable dataset can be challenging. Moreover, the success of the project depends on the assumption that fraudulent activities leave distinct patterns in the data, which may not always be the case. Additionally, while the comparative study aims to find the most effective algorithm, it may not consider all possible factors relevant to practical implementation, such as the scalability and interpretability of the chosen models. Furthermore, the project may face ethical considerations regarding user privacy and data handling, which should be carefully addressed. Lastly, the project's results might not be directly applicable to all blockchain networks, as the characteristics of the data and the nature of fraud may vary between different systems.

CHAPTER 2
BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1. A Machine Learning and Blockchain Based Efficient Fraud Detection Mechanism

2.1.1. Introduction

Every industry, including banking, education, health care, and others, has modernized as a result of technological growth [1]. Moreover, with the advent of communication technology, online transactions and means of payment are also being modernized. Through this modernization, traditional currencies are being converted into digital currencies, and all financial transactions are being conducted digitally[2]. However, these transactions are not fully secured and are vulnerable to various digital attacks, such as fraud issues, anomalies, and privacy breaches. Additionally, as the volume of transactions rises, there is an increase in fraud associated with financial transactions. As a result, billions of dollars are lost globally every year. Any suspicious activity on a network that behaves abnormally is called an anomaly. In cybersecurity and digital financial exchange, anomaly detection is used to detect fraud and network invasion. The goal of anomaly detection is to protect the network from illegal and fraudulent activities. In the financial sector, anomaly detection applications have investigated suspicious activity and identified hackers and fraudulent users.

However, all anomaly detection methods in traditional financial systems are designed for centralized systems[2]. Therefore, with the development of digital currencies, such as Bitcoin, anomaly detection methods using the blockchain are improving. Despite these advances, there are still many fraud occurrences. Many artificial intelligences (AI) and machine learning techniques have been proposed to detect anomalies and fraud in digital transactions; however, there is no suitable solution for centralized systems.

2.1.2. Merits, Demerits and Challenges

Blockchain is the latest and most secure technology that covers various research areas related to security. Blockchain development is based on digital currencies and is used to secure digital financial transactions. It protects financial systems from fraudulent attacks. Therefore, a blockchain-based machine learning algorithm is proposed to secure digital transactions. The proposed model predicts whether the incoming transaction in the blockchain is fraudulent or not. The proposed machine learning algorithms are trained and tested on a bitcoin-based dataset based on bitcoin transactions and predict the behavior of the incoming transactions. The given dataset is based on 30,047 entities, with smaller numbers of fraudulent entities. Due to the small amount of fraudulent data in the dataset, good results cannot be obtained because of the data imbalance problem. Therefore, we generate synthetic malicious data points through SMOTE to achieve better results. We use XGboost and random forest to classify the model and calculate the confusion matrix. This classification allows the model to distinguish between fraudulent and real data. The simulation results show that the proposed algorithm works adequately to find transaction fraud. Moreover, two attacker models are implemented to check the efficacy of the system against bugs and attacks. The proposed system is robust against double-spending and Sybil attacks

A major limitation of this proposal is that it can be affected by the adversarial attack it also address such threats.

2.1.3. Implementation

This section first presents the simulation results of the model, then we present the results after inducing modern cyber attacks to the system, i.e., Sybil attack, and double-spending attack[4]. The selected dataset is highly skewed, as shown in Figures 2.1.3.1 and 2.1.3.2. The classification models are biased toward the majority class due to the imbalance of the data.

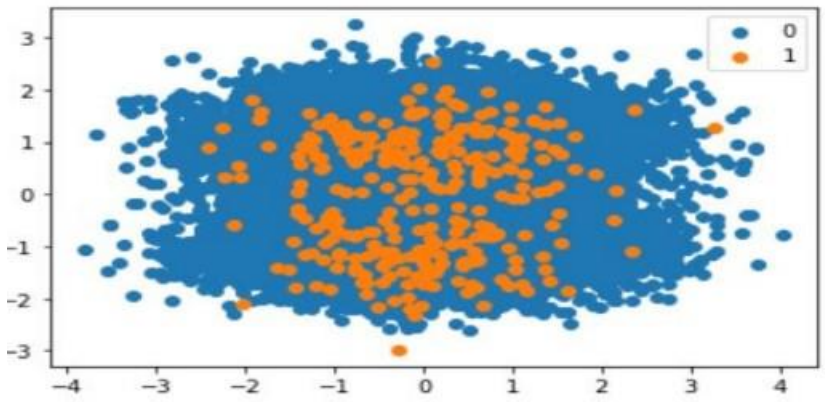


Figure 2.1.3.1: Imbalanced data

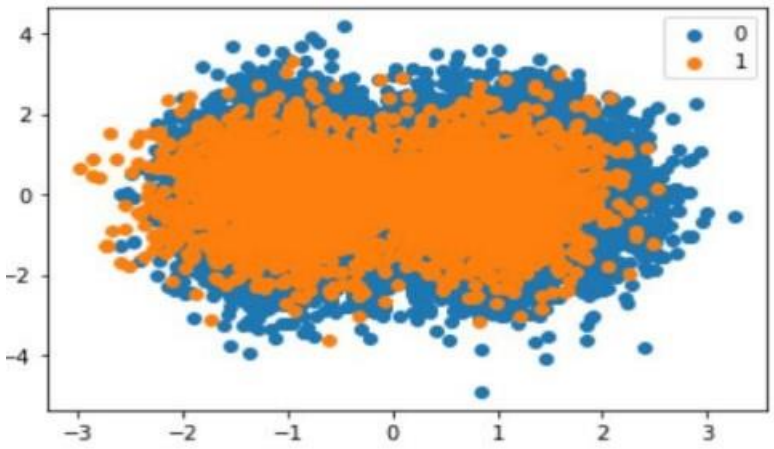


Figure 2.1.3.2: Balanced data

Figure 2.1.3.1 shows the presence of malicious and honest transactions in the dataset. It can be seen from the figure that the number of honest transactions is higher than the number of malicious transactions. This imbalanced nature of the data leads to a bias in the classification. Synthetic data are used to solve this problem. The malicious entities are oversampled using SMOTE. The synthesized transactions are added to the dataset to limit the bias of the model during classification. The results obtained after using SMOTE are shown in Figure 2.1.3.2. The observed log loss of XGBoost during training is shown in Figure 2.1.3.3. The log loss is observed for both the training data and the test data. From the figure, it can be seen that at a count of 10 iterations, a drastic drop is observed for both the training and test data.

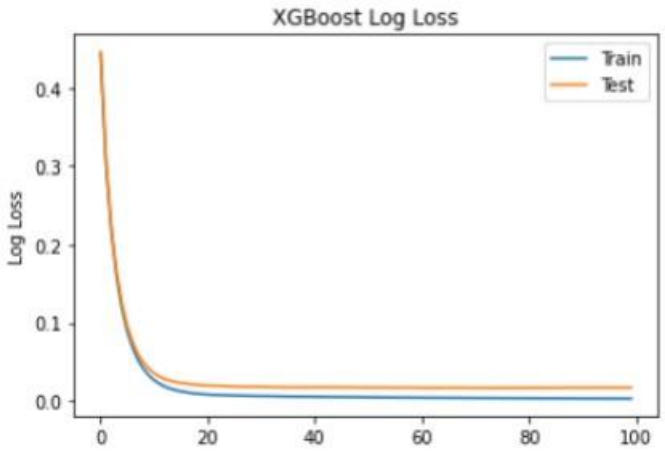


Figure 2.1.3.3: Logloss of XGboost

Moreover, the smoothness of the curves indicates that the model efficiently captures the nonlinear patterns of the data. For the test data, the log loss is higher than for the training data. However, the difference is not too large. The smaller difference between the training and test curves indicates that the model is well trained on unseen data. The trained model can be applied to real-world scenarios for anomaly detection in blockchain networks.

Figure 2.1.3.4 shows the correlation between the fraudulent and non-fraudulent class. Meanwhile, thevalue almost equal to 0, in the case of mean ni nb tc, shows the minimum correlation between fraudulent and non-fraudulent.

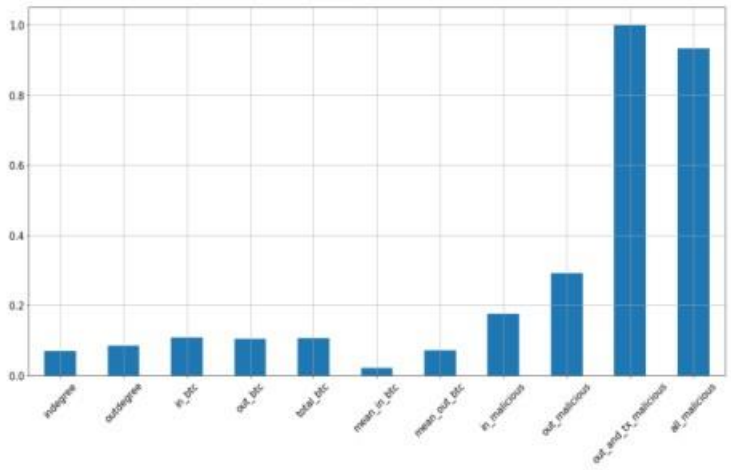


Figure 2.1.3.4: Correlation with class Fraudulent or not

Figure 2.1.3.4 shows the error that occurs when classifying with XGBoost. It shows the error for both training and test data. It can be observed that the classification error decreases as the number of iterations increases. The error is high for training data, and the figure shows a gradual decrease, while it is lower for test data and decreases rapidly. The precision–recall curve of the XGboost model is visualized in Figure 2.1.3.5. This curve predicts the harmonic mean of both precision and recall. It is seen that a very slight decrease is observed, starting from 1. As soon as the recall value reaches more than 0.9, there is a sudden drop in the precision value. Figure 2.1.3.6 shows the accuracy when XGBoost is used. It shows that the highest peak of 0 to 1 indicates that the model achieves optimal accuracy in classifying blockchain transactions as legitimate or malicious. After reaching the maximum value of 0.9, the accuracy remains constant throughout the training.

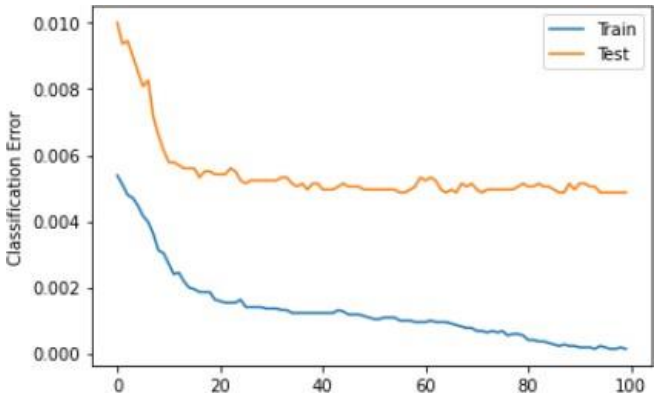


Figure 2.1.3.5: Classification error of XGboost

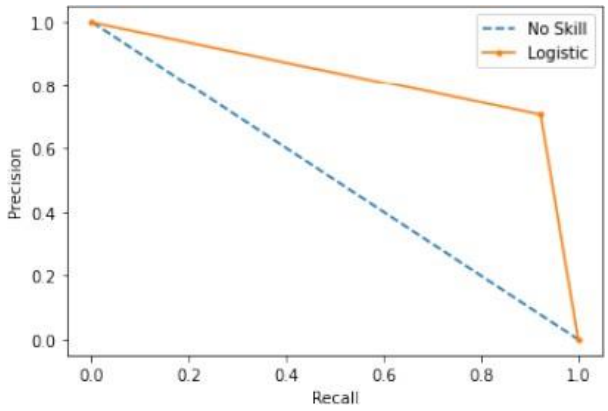


Figure 2.1.3.6: Precision of RF

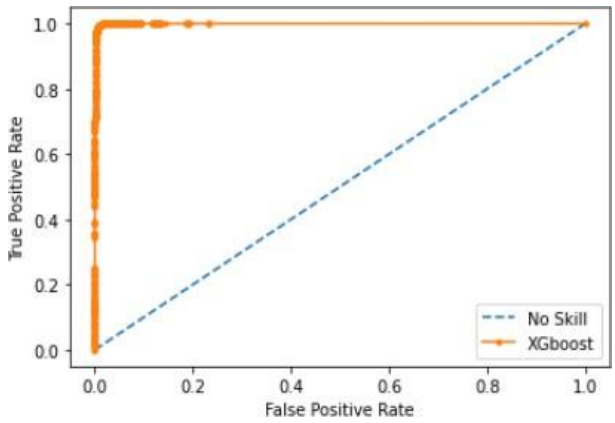


Figure 2.1.3.7: Accuracy of XGboost

Figure 2.1.3.7 shows the confusion matrix obtained using RF. In this matrix, random forest selects 9014 random samples, correctly identifying 9009 predictions. This means that the proposed model efficiently discriminates between malicious and legitimate transactions. The matrix shows that the highest values are obtained in the case of true negatives, namely 99%. In the other three cases, the number of values is lower. This shows that the proposed model is efficient in detecting true negative transactions. Moreover, the phenomenon of majority voting in the random forest increases the performance of the model during classification. Figure 2.1.3.8 shows the AUC of a random forest. The AUC describes how well the model distinguishes between the positive and negative classes. It can be seen that the value of the AUC increases dramatically at the beginning to almost 0.85. Thereafter, a gradual increase is observed until the maximum value of 0.92 AUC is reached. The random forest model achieves an AUC of 0.92, which means that it performs well in capturing legitimate and malicious transactions.

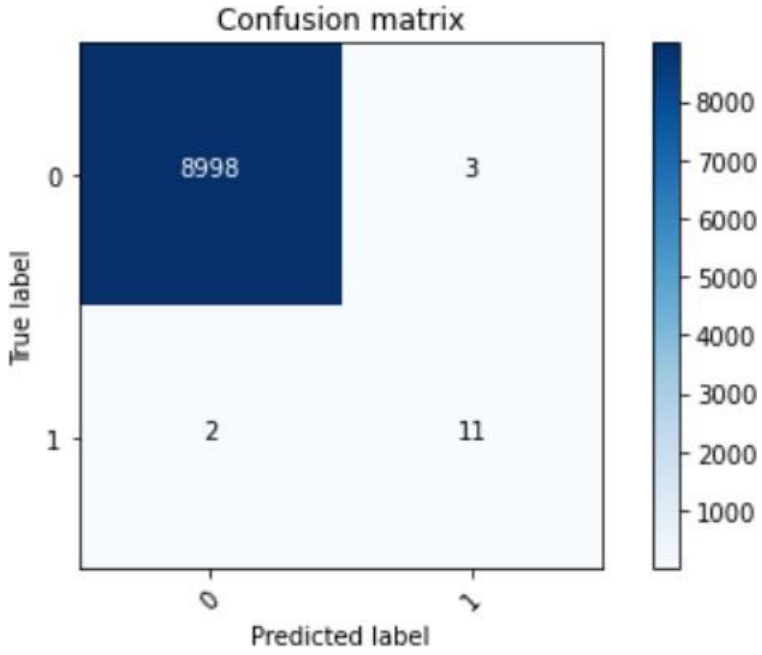


Figure 2.1.3.8: Confusion matrix with random forest

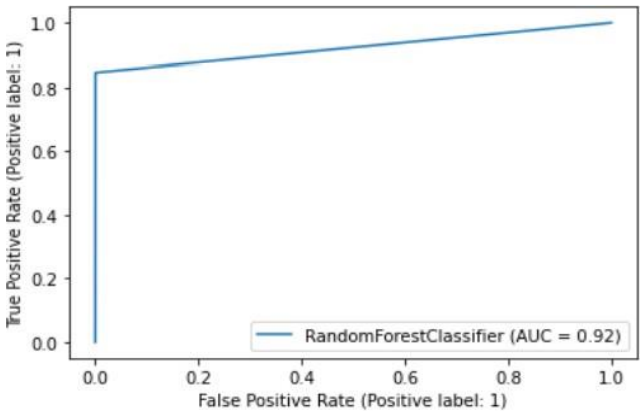


Figure 2.1.3.9: Accuracy of Random Forest

Figure 2.1.3.9 shows the transaction and execution costs incurred in executing the functions involved in the blockchain smart contract. The costs are expressed in terms of gas, a basic unit of gas consumption in the blockchain network. From the figure 2.1.3.10, it can be seen that the transaction costs of all functions remain the same, while the execution costs of the publish transaction function are the highest, as mining costs are also included. Overall, the transaction costs are higher than the execution costs for all functions. The reason for this is that the former includes the processing costs of entire transactions, while the latter includes only the execution costs of some operations in a given function.

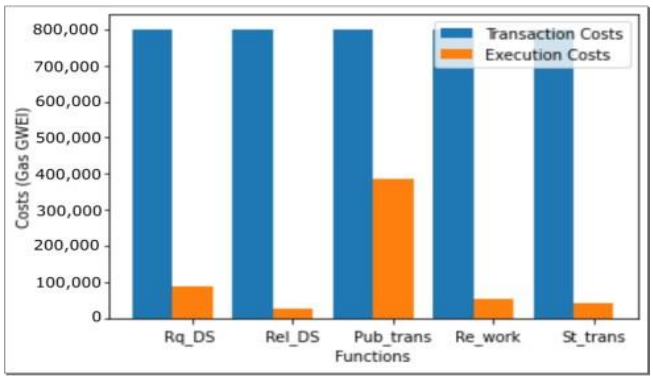


Figure 2.1.3.10: Transactions published and stored on Blockchain

In the blockchain, a transaction is only confirmed after the agreement/verification of all nodes. This verification takes a specific period, which creates a chance for cyber attacks. Double spending is one of these attacks that exploit the transaction verification time. Every transaction on the blockchain takes time for verification, and attackers use this time to their advantage. During the transaction verification delay, the attacker uses the same coin at two places as the verification of both transactions takes place simultaneously. In this way, digital currency is duplicated and falsified easily. The authors worked on the two double-spending attacker models. They enhance the two existing attacker models of Satoshi Nakamoto and Rosenfield for double spending. The first proposed model is called the “generalized model”, inwhich authors added a time parameter. This parameter is used to calculate the time advantage of an attacker. The second proposed model is known as the time-based model. This model countsthe time when an attacker and honest node mined their last blocks.

2.2 Fraud Detection: A Review on Blockchain

2.2.1 Introduction

In recent years, the world of blockchain technology has captured widespread attention. This dynamic field has witnessed the introduction of multiple measures aimed at detecting and mitigating fraudulent transactions and unusual activities that deviate from established behavioral patterns[5]. To bolster the accuracy of fraud detection, a range of outlier analysis techniques have been employed, including decision trees, support vector machines, evolutionary algorithms, Bayesian belief networks, and the formidable neural networks. Despite considerable progress, only a limited number of models have proven capable of reliably identifying all forms of fraudulent behavior.

Even after unmasking deception and accurately tracing the origins of deceitful inputs, the quest for truth and well-informed decision-making remains an elusive endeavor. As our interactions increasingly transition to the digital realm, upholding fundamental principles of security, precision, reliability, and transparency among market participants becomes paramount. This research scrutinizes the global landscape of operational fraud detection systems, spotlighting their demonstrated effectiveness in combating deceptive practices. We delve into ten diverse domains where blockchain technology has wielded transformative influence and propose practical solutions to the everyday challenges confronting both individuals and organizations.

2.2.2 Merits, Demerits and Challenges

Blockchain technology is described in this paper as a way to decrease the layers of corruption in government procedures. This study focuses on employing a novel form of encryption method to overcome security and privacy concerns in blockchain. The proposed model ensures that everyone linked to a soon-to-be implemented blockchain network may see all government procedures. This enables ordinary citizens to investigate the operation of any government scheme, track its progress, and track financial transfers.

- The following are some of the drawbacks of this model:
- Stagnation of funds caused by middle-level authorities.
- Money is misappropriated in the middle levels, with everyone blaming each other.
- Schemes are executed slowly.
- Identifying the true needy/beneficiary is a challenge.
- Inappropriate financial allocation.

They also review state-of-the-art technologies for detecting online fraud and intrusions, identify certain fraud and malicious activities that blockchain technology can effectively prevent, and make recommendations for strategically fighting various attacks to which blockchain technology may be vulnerable[6]. Existing machine learning and data-mining algorithms could find new uses in identifying fraud and intrusions in blockchain-based transactions. Guided machine learning methods like deep-learning neural networks, support vector machines, and Bayesian belief networks may help detect outlier behaviors by profiling, monitoring, and detecting behavioral trends based on people's transaction histories. Despite the advancement in technology, still, the problems regarding Video Fraudulence are faced and there is no concrete solution for this problem.

2.2.3 Implementation

Blockchain applications have attracted a lot of attention. They are more valuable than money and can be used to replace fiat money and traditional banking. The ability to trade wealth on a blockchain, on the other hand, is at the heart of the system and must be reliable. Blockchains have built-in features that assure the system's stability and durability. Malicious actors can still use well-known tactics to steal money, such as virus software or falsified emails. We also undertake a sensitivity analysis to show how the models offered rely on specific attributes and how the lack of some of them impacts overall system performance. Blockchain can be used to fight and prevent fraud in a business network[7]. One of the fundamental characteristics that determines blockchain's worth is its ability to share data rapidly and securely without relying on a single institution to assume responsibility for data security. One of the most significant advantages of blockchain technology is increased security. The increased security provided by blockchain is due to the way the technology works: With end-to-end encryption, blockchain generates an unalterable record of transactions, preventing fraud and unauthorized activity.

Furthermore, blockchain data is kept across a network of computers, making it nearly impossible to attack (unlike conventional computer systems that store data together in servers). Furthermore, by anonymizing data and requiring permissions to limit access, blockchain can solve privacy concerns better than traditional computer systems. Because blockchain transactions cannot be removed or modified, they are immutable. Before a "block" of transactions can be added to the blockchain, network participants must agree that the transaction is valid via a consensus process.

2.3 Analysis of Fraud Detection in Blockchain system using Machine LearningAlgorithms

2.3.1 Introduction

Blockchain uses end-to-end encryption to produce a changeless record of exchanges, eliminating fraud and other illegal activity[9]. Information is stored on the blockchain using a network of PCs, making it virtually impossible to hack (in contrast to ordinary PC frameworks that stores information together in servers). Additionally, blockchain can more easily address security concerns than conventional PC frameworks by encrypting information and requiring consents to limit access. Blockchain transactions are constant because once they are recorded, they cannot be changed or reversed. Before a "block" of transactions is added to the blockchain, network clients must agree on the validity of each transaction.

Fraudsters employ a variety of techniques to hide their illegal activities, including the production of fictitious records, the alteration of physical or electronic records, and the manipulation of data in an association's bookkeeping frameworks. Using a shared electronic record can help reduce extortion because itincreases the openness and clarity of communications between members of a company organization and within a production network. False trades are easier to spot since groups can track the evolution of resources and experiences.

2.3.2 Merits, Demerits and Challenges

The research proactively addresses the challenge of identifying fraudulent transactions in blockchain networks, acknowledging the potential harm they can inflict on the economy and user confidence. This forward-looking approach is essential for maintaining trust in the blockchain ecosystem. The study explores both traditional Machine Learning models and deep learning models, indicating a comprehensive examination of AI methods to detect fraudulent transactions. This breadth of analysis can help identify the most effective techniques, balancing precision and processing efficiency. By investigating AI algorithms, the research aims to identify clients and transactions with a higher likelihood of engaging in fraudulent activities. This has the potential to significantly reduce the impact of fraudulent exchanges and bolster the securityof blockchain networks.

The abstract does not provide specific findings or outcomes of the research, making it challenging to assess the effectiveness of the AI models discussed. Readersare left wondering about the practical impact of the study.The abstract uses technical language and concepts that may be challenging for a non-technical audience to understand. This could limit the accessibility of the research to a broader readership.

The blockchain landscape is continually evolving, and fraudsters adapt to new methods and technologies[10]. Detecting fraudulent transactions requires staying ahead of these evolving tactics, which poses a substantial challenge. Effective AI models for fraud detection require high-quality, labeled data. Gathering such data in the blockchain context can be a challenge due to the decentralized and often anonymous nature of transactions. Balancing fraud detection with user privacy is a significant challenge. The use of AI to identify fraudulent behavior may inadvertently infringe on user privacy rights, raising ethical and legal concerns.

2.3.3 Implementation

Pre-handling stage: We preprocess using network node embedding and the node2vec method. The combined ratings dataset is then read to produce a data frame. The node2vec method's outputs are then normalized, and the normalized values are then saved in a file. When a transaction is discovered to be fraudulent, we assign it a score of 1, and when it is not, we assign it a score of 0. Then the mean and SD of the node features are calculated, and the outcomes are saved to a CSV file. Next, train and test sets of the gathered data were created.

Building and preparing different models: Test (0.2) and train (0.8) data were included in our analysis. Then, in our train and test sets, we evaluate the ratio of honest to dishonest transactions.We use machine learning and deep learning techniques to predict the likelihood that a transaction will be successful.

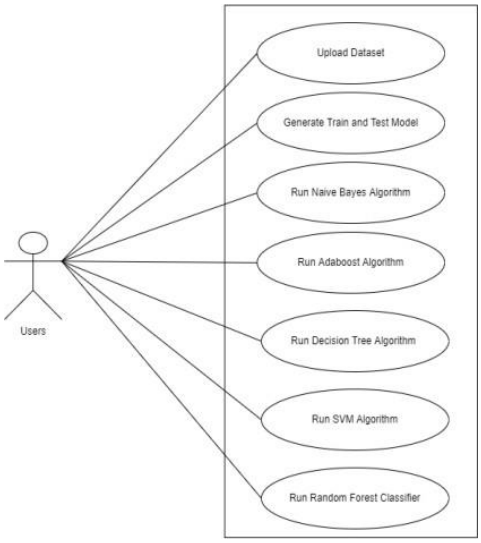


Figure 2.3.3.1 Preprocessing steps

Performance assessment of the relative multitude of models: We evaluate each of our classification models. In order to estimate a parameter, sampling in machine learning involves selecting a sample of data from the dataset with replacement. So, we start by choosing the bootstrap sample size. The model's efficacy is next evaluated using the mean of all accuracy values obtained in this way, after which the sample size is determined.

CHAPTER 3
PROPOSED SYSTEM

CHAPTER 3

PROPOSED SYSTEM

3.1. Objective of Proposed Model

The research objective is to undertake a detailed comparative analysis of multiple supervised machine learning models, such as Support Vector Machine (SVM), Decision Tree, Naive Bayes, and Logistic Regression, along with various deep learning architectures, to ascertain their effectiveness in identifying fraudulent transactions within a blockchain network. The study aims to delve into the nuanced performance metrics of each algorithm, including accuracy and computational efficiency, to better understand their respective strengths and limitations. By rigorously evaluating these models, the research seeks to provide valuable insights into the trade-offs between accuracy and computational speed, thereby enabling stakeholders to make informed decisions regarding the selection of the most suitable algorithm for detecting fraudulent activities in blockchain systems.

This comparative study holds significant implications for enhancing the security and integrity of blockchain networks, offering a roadmap for the implementation of robust fraud detection mechanisms tailored to the unique challenges posed by decentralized ledger technologies.

3.2. Algorithms Used for Proposed Model

For implementing the model we have compared the eight different algorithms in the propsed model.

Logistic Regression: Logistic Regression is a statistical method used for binary classification tasks. It calculates the probability of a sample belonging to a certain class based on its features and applies a logistic function to make predictions, making it effective for linearly separable data.

Multilayer Perceptron: Multilayer Perceptron is a type of artificial neural network that comprises multiple layers of nodes (neurons) with nonlinear activation functions. It excels in capturing complex patterns in data by introducing nonlinearity through hidden layers, making it suitable for tasks where data cannot be linearly separated.

Naive Bayes: Naive Bayes is a probabilistic classifier based on Bayes' theorem with the assumption of independence between features. It calculates the probability of a sample belonging to each class and selects the class with the highest probability, making it efficient and effective for text classification and other similar tasks.

Adaboost: Adaboost is an ensemble learning method that combines multiple weak classifiers to create a strong classifier. It iteratively trains models on the dataset, adjusting the weights of misclassified samples to focus on difficult-to-classify instances, ultimately improving overall accuracy.

Decision Tree: Decision Tree is a tree-like structure where each internal node represents a feature, each branch represents a decision rule, and each leaf node represents a class label. It partitions the data into subsets based on the value of attributes and is particularly useful for interpretable classification tasks.

Support Vector Machine (SVM): SVM is a supervised learning algorithm that finds the optimal hyperplane to separate data points of different classes in a high-dimensional space. It works by transforming the input data into a higher-dimensional space using kernel functions, allowing it to find a maximal margin hyperplane for classification.

Random Forest Classifier: Random Forest Classifier is an ensemble learning method that builds multiple decision trees during training and combines their predictions through averaging or voting. It improves upon the decision tree algorithm by reducing overfitting and increasing accuracy through the aggregation of multiple models.

Neural Network: Neural Network is a computational model inspired by the structure and function of biological neural networks. It consists of interconnected layers of neurons, each performing weighted summation and applying an activation function. The use of multiple layers allows neural networks to learn complex patterns and relationships in data, making them highly versatile for various machine learning tasks.

3.3. Designing

3.3.1 UML DIAGRAM

A. USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. 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.

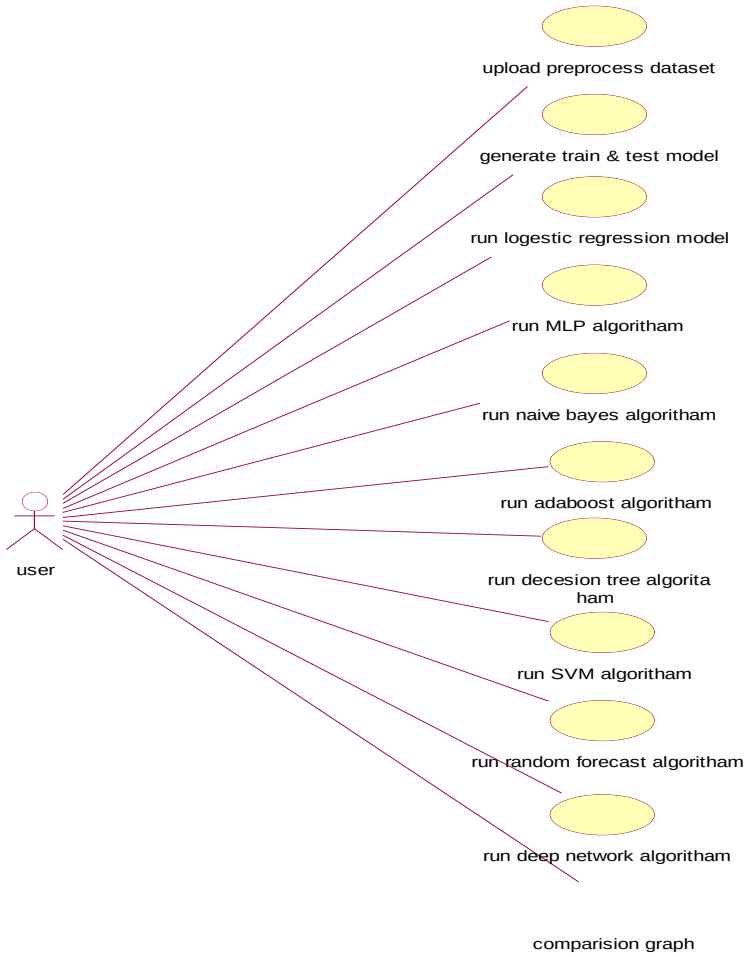


Figure 3.3.1.1: Use Case Diagram

B. Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

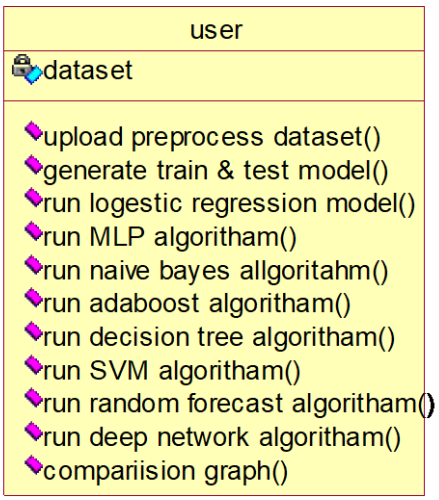


Figure 3.3.1.2: Class Diagram

C. Sequence Diagram

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. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

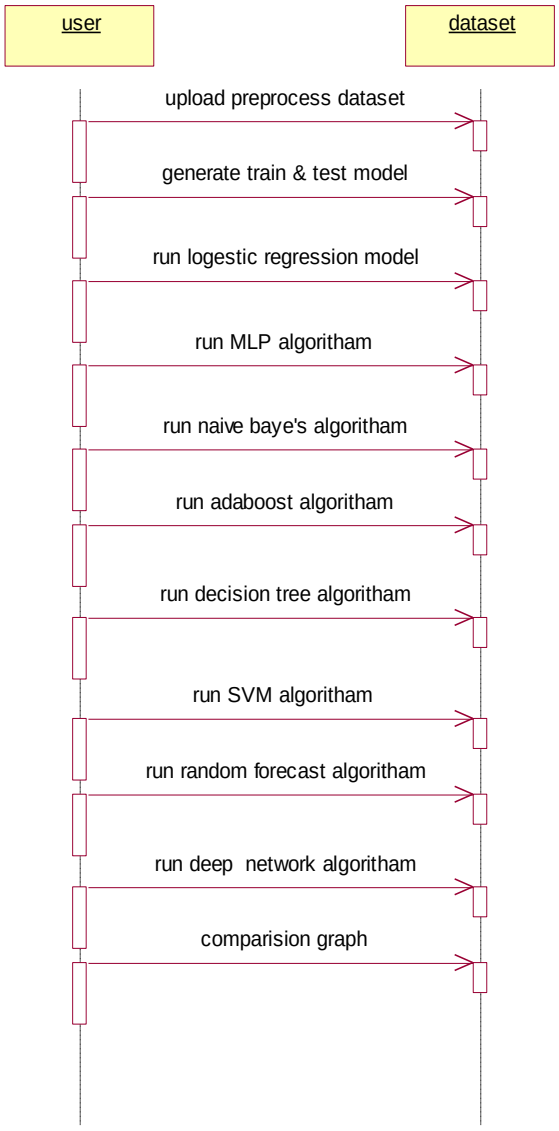


Figure 3.3.1.3: Sequence Diagram

3.3. Stepwise Implementation and Code

The project is structured into several modules to facilitate the implementation process. The first module, "Upload & Preprocess Dataset," involves providing users with a button to upload and read the dataset, followed by a preprocessing step to handle missing values. This ensures that the dataset is clean and ready for analysis.

The next module, "Generate Train & Test Model," allows users to generate the training and testing datasets. Upon clicking the button, users are presented with an overview of the dataset, including the total number of records and columns. The dataset is then split into training and testing subsets. Users can subsequently run various machine learning algorithms by clicking on individual buttons.

After running the algorithms, users are provided with the performance metrics or accuracy of each algorithm. Additionally, the remaining algorithms' accuracy is displayed, allowing users to compare the performance of different algorithms. This comparative analysis helps users identify the most effective algorithm for their specific use case.

Furthermore, the project includes a "Comparison Graph" module, which generates a graphical representation of the performance comparison between different algorithms. This graph provides a visual aid for users to interpret and understand the relative strengths of each algorithm more intuitively.

Overall, the project's modular design streamlines the dataset preprocessing, model training, and performance evaluation processes, enabling users to efficiently analyze and compare various machine learning algorithms for their specific application.

CODE:

Main.py

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
from tkinter.filedialog import askopenfilename
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import normalize
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import os
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn import svm
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
import seaborn as sns
import webbrowser
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import AdaBoostClassifier
from keras.utils.np_utils import to_categorical
from keras.layers import  MaxPooling2D
from keras.layers import Dense, Dropout, Activation, Flatten
```

```
from keras.layers import Convolution2D
from keras.models import Sequential
from keras.models import model_from_json
import pickle

global filename
global X,Y
global dataset
global main
global text
accuracy = []
precision = []
recall = []
fscore = []
global X_train, X_test, y_train, y_test, predict_cls
global classifier

main = tkinter.Tk()
main.title("Comparative Study of Machine Learning Algorithms for Fraud Detection in Blockchain") #designing main screen
main.geometry("1300x1200")

#fucntion to upload dataset
def uploadDataset():
    global filename
    global dataset
    text.delete('1.0', END)
    filename = filedialog.askopenfilename(initialdir="Dataset")
    text.insert(END,filename+" loaded\n\n")
    dataset = pd.read_csv(filename)
    text.insert(END,"Dataset before preprocessing\n\n")
```

```
    text.insert(END,str(dataset.head()))
    text.update_idletasks()
    label = dataset.groupby('FLAG').size()
    label.plot(kind="bar")
    plt.title("Blockchain Fraud Detection Graph 0 means Normal & 1 means Fraud")
    plt.show()
#function to perform dataset preprocessing
def trainTest():
    global X,Y
    global dataset
    global X_train, X_test, y_train, y_test
    text.delete('1.0', END)
    #replace missing values with 0
    dataset.fillna(0, inplace = True)
    Y = dataset['FLAG'].ravel()
    dataset = dataset.values
    X = dataset[:,4:dataset.shape[1]-2]
    X = normalize(X)
    indices = np.arange(X.shape[0])
    np.random.shuffle(indices)
    X = X[indices]
    Y = Y[indices]
    X = X[0:5000]
    Y = Y[0:5000]
    print(Y)
    print(X)
    text.insert(END,"Dataset after features normalization\n\n")
    text.insert(END,str(X)+"\n\n")
    text.insert(END,"Total records found in dataset : "+str(X.shape[0])+"\n")
    text.insert(END,"Total features found in dataset: "+str(X.shape[1])+"\n\n")
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
```

```
text.insert(END,"Dataset Train and Test Split\n\n")
text.insert(END,"80% dataset records used to train ML algorithms : "+str(X_train.shape[0])+"\n")
    text.insert(END,"20% dataset records used to train ML algorithms : "+str(X_test.shape[0])+"\n")
def calculateMetrics(algorithm, predict, y_test):
    a = accuracy_score(y_test,predict)*100
    p = precision_score(y_test, predict,average='macro') * 100
    r = recall_score(y_test, predict,average='macro') * 100
    f = f1_score(y_test, predict,average='macro') * 100
    accuracy.append(a)
    precision.append(p)
    recall.append(r)
    fscore.append(f)
    text.insert(END,algorithm+" Accuracy  :  "+str(a)+"\n")
    text.insert(END,algorithm+" Precision : "+str(p)+"\n")
    text.insert(END,algorithm+" Recall    : "+str(r)+"\n")
    text.insert(END,algorithm+" FScore    : "+str(f)+"\n\n")

def runLogisticRegression():
    global X,Y, X_train, X_test, y_train, y_test
    global accuracy, precision,recall, fscore
    accuracy.clear()
    precision.clear()
    recall.clear()
    fscore.clear()
    text.delete('1.0', END)
    lr = LogisticRegression()
    lr.fit(X, Y)
    predict = lr.predict(X_test)
    calculateMetrics("Logistic Regression", predict, y_test)
```

```
def runMLP():
    mlp = MLPClassifier()
    mlp.fit(X_train, y_train)
    predict = mlp.predict(X_test)
    calculateMetrics("MLP", predict, y_test)
def runNaiveBayes():
    cls = GaussianNB()
    cls.fit(X_train, y_train)
    predict = cls.predict(X_test)
    calculateMetrics("Naive Bayes", predict, y_test)
def runAdaBoost():
    cls = AdaBoostClassifier()
    cls.fit(X_train, y_train)
    predict = cls.predict(X_test)
    calculateMetrics("AdaBoost", predict, y_test)
def runDT():
    global predict_cls
    cls = DecisionTreeClassifier()
    cls.fit(X_train, y_train)
    predict = cls.predict(X_test)
    calculateMetrics("Decision Tree", predict, y_test)
def runSVM():
    cls = svm.SVC()
    cls.fit(X_train, y_train)
    predict = cls.predict(X_test)
    calculateMetrics("SVM", predict, y_test)
def runRF():
    global predict_cls
    rf = RandomForestClassifier()
    rf.fit(X_train, y_train)
    predict = rf.predict(X_test)
    predict_cls = rf
    calculateMetrics("Random Forest", predict, y_test)
```

```
def predict():
    global predict_cls
    text.delete('1.0', END)
    filename = filedialog.askopenfilename(initialdir="Dataset")
    dataset = pd.read_csv(filename)
    dataset.fillna(0, inplace = True)
    dataset = dataset.values
    X = dataset[:,4:dataset.shape[1]-2]
    X1 = normalize(X)
    prediction = predict_cls.predict(X1)
    print(prediction)
    for i in range(len(prediction)):
        if prediction[i] == 0:
            text.insert(END,"Test DATA : "+str(X[i])+" ===> PREDICTED AS NORMAL\n\n")
        else:
            text.insert(END,"Test DATA : "+str(X[i])+" ===> PREDICTED AS FRAUD\n\n")


def runDeepNetwork():
    global X, Y
    X = np.reshape(X, (X.shape[0], X.shape[1], 1, 1))
    Y = to_categorical(Y)
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
    if os.path.exists('model/model.json'):
        with open('model/model.json', "r") as json_file:
            loaded_model_json = json_file.read()
            classifier = model_from_json(loaded_model_json)
        json_file.close()
        classifier.load_weights("model/model_weights.h5")
        classifier._make_predict_function()
    else:
        classifier = Sequential()
```

```
        classifier.add(Convolution2D(32, 1, 1, input_shape = (X_train.shape[1], X_train.shape[2], X_train.shape[3]), activation = 'relu'))
        classifier.add(MaxPooling2D(pool_size = (1, 1)))
        classifier.add(Convolution2D(32, 1, 1, activation = 'relu'))
        classifier.add(MaxPooling2D(pool_size = (1, 1)))
classifier.add(Flatten())
        classifier.add(Dense(output_dim = 256, activation = 'relu'))
        classifier.add(Dense(output_dim = Y.shape[1], activation = 'softmax'))
        print(classifier.summary())
        classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
        hist = classifier.fit(X, Y, batch_size=16, epochs=10, shuffle=True, verbose=2)
        classifier.save_weights('model/model_weights.h5')
        model_json = classifier.to_json()
        with open("model/model.json", "w") as json_file:
            json_file.write(model_json)
        json_file.close()
    predict = classifier.predict(X_test)
    predict = np.argmax(predict, axis=1)
    y_test = np.argmax(y_test, axis=1)
    calculateMetrics("Deep Neural Network", predict, y_test)
def graph():
    output = "<html><body><table align=center border=1><tr><th>Algorithm Name</th><th>Accuracy</th><th>Precision</th><th>Recall</th>"
    output+="<th>FSCORE</th></tr>"
    output+="<tr><td>Logistic Regression Algorithm</td><td>"+str(accuracy[0])+"</td><td>"+str(precision[0])+"</td><td>"+str(recall[0])+"</td><td>"+str(fscore[0])+"</td></tr>"
    output+="<tr><td>MLP
```

```
Algorithm</td><td>"+str(accuracy[1])+"</td><td>"+str(precision[1])+"</td><td>"+str(recall
[1])+"</td><td>"+str(fscore[1])+"</td></tr>"
   output+="<tr><td>Naive Bayes
Algorithm</td><td>"+str(accuracy[2])+"</td><td>"+str(precision[2])+"</td><td>"+str(recall
[2])+"</td><td>"+str(fscore[2])+"</td></tr>"
   output+="<tr><td>AdaBoost
Algorithm</td><td>"+str(accuracy[3])+"</td><td>"+str(precision[3])+"</td><td>"+str(recall
[3])+"</td><td>"+str(fscore[3])+"</td></tr>"
   output+="<tr><td>Decision Tree
Algorithm</td><td>"+str(accuracy[4])+"</td><td>"+str(precision[4])+"</td><td>"+str(recall
[4])+"</td><td>"+str(fscore[4])+"</td></tr>"
   output+="<tr><td>SVM
Algorithm</td><td>"+str(accuracy[5])+"</td><td>"+str(precision[5])+"</td><td>"+str(recall
[5])+"</td><td>"+str(fscore[5])+"</td></tr>"
   output+="<tr><td>Random Forest
Algorithm</td><td>"+str(accuracy[6])+"</td><td>"+str(precision[6])+"</td><td>"+str(recall
[6])+"</td><td>"+str(fscore[6])+"</td></tr>"
   output+="<tr><td>Deep Neural Network
Algorithm</td><td>"+str(accuracy[7])+"</td><td>"+str(precision[7])+"</td><td>"+str(recall
[7])+"</td><td>"+str(fscore[7])+"</td></tr>"
   output+="</table></body></html>"

f = open("table.html", "w")
   f.write(output)
   f.close()
   webbrowser.open("table.html",new=2)

main.config(bg='LightSkyBlue')
main.mainloop()
```

3.4 Model Architecture

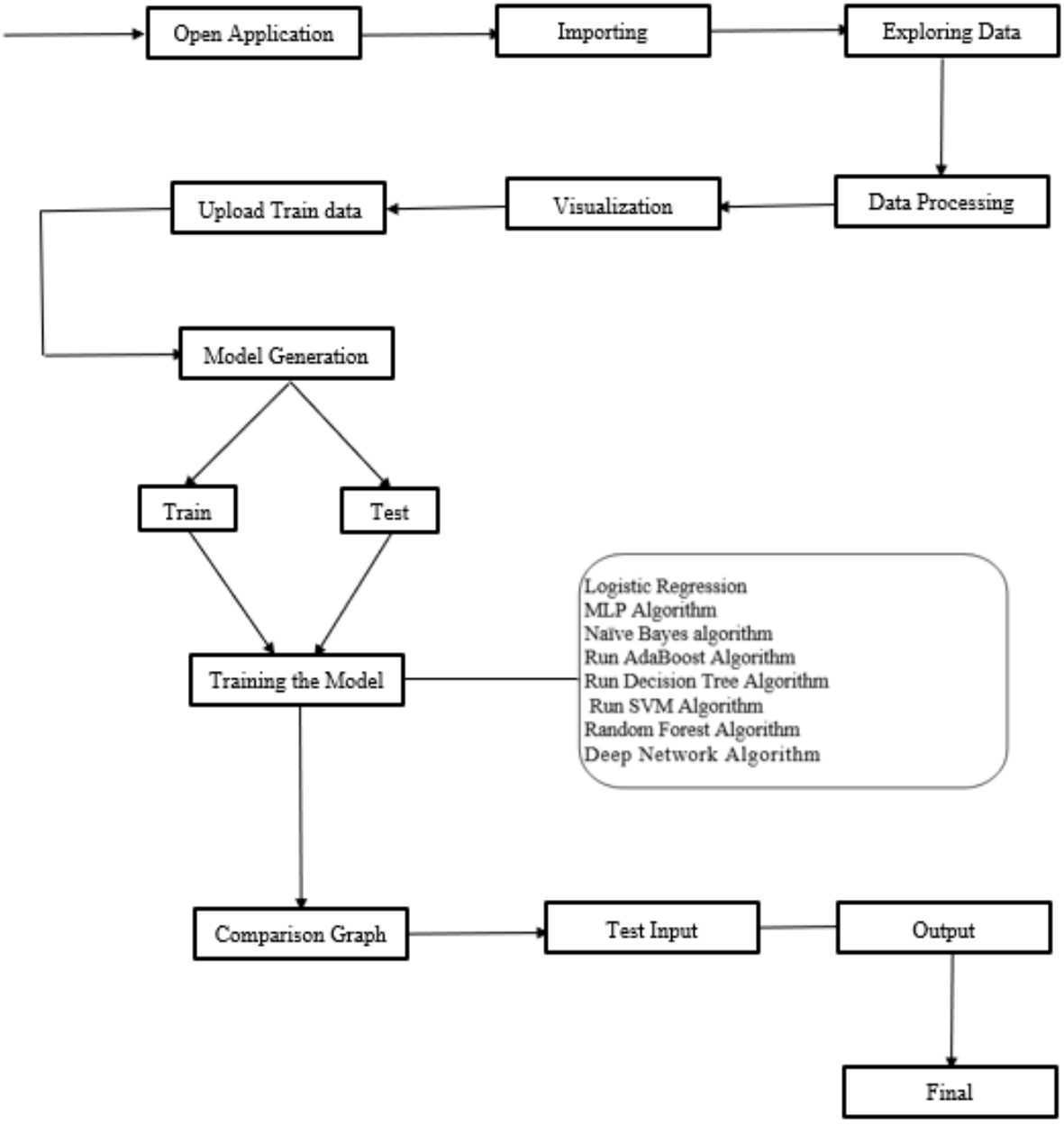


Figure 3.4.1: Model Architecture

In Figure 3.4.1, we begin by opening the application and importing the data. Next, we upload the dataset and proceed to process it by filling in any missing values. Subsequently, we split the data into training and testing sets. Each algorithm is then trained on the training data, and graphs are generated to assess their performance. This allows us to determine which algorithm performs the best. Finally, we utilize the test data to identify fraudulent transactions.

CHAPTER 4
RESULTS AND DISCUSSION

CHAPTER 4

RESULTS AND DISCUSSION

4.1. Output Screens

To run project double click on 'run.bat' file to get below screen

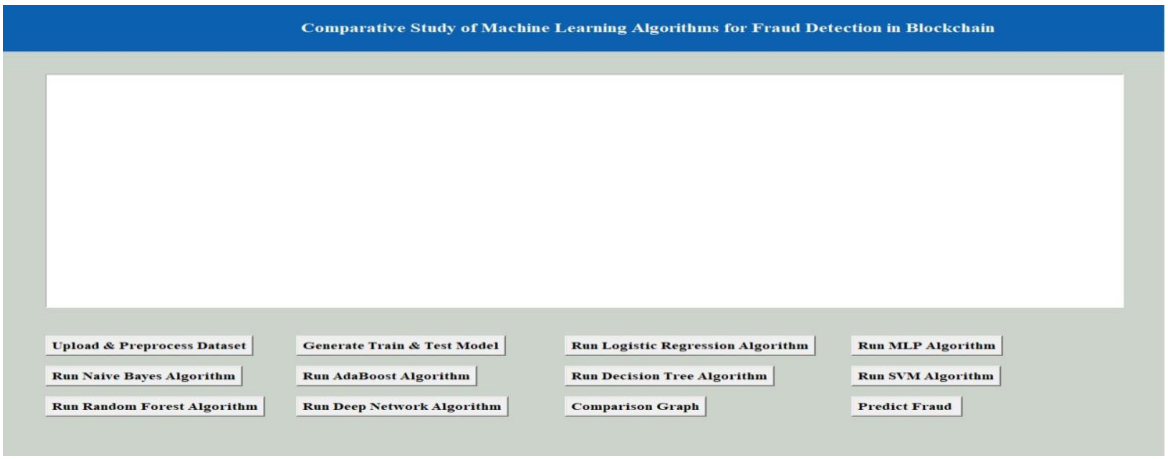


Figure 4.1.1: output screen

In above screen click on 'Upload & Preprocess Dataset' button to upload and read dataset and then remove missing values.

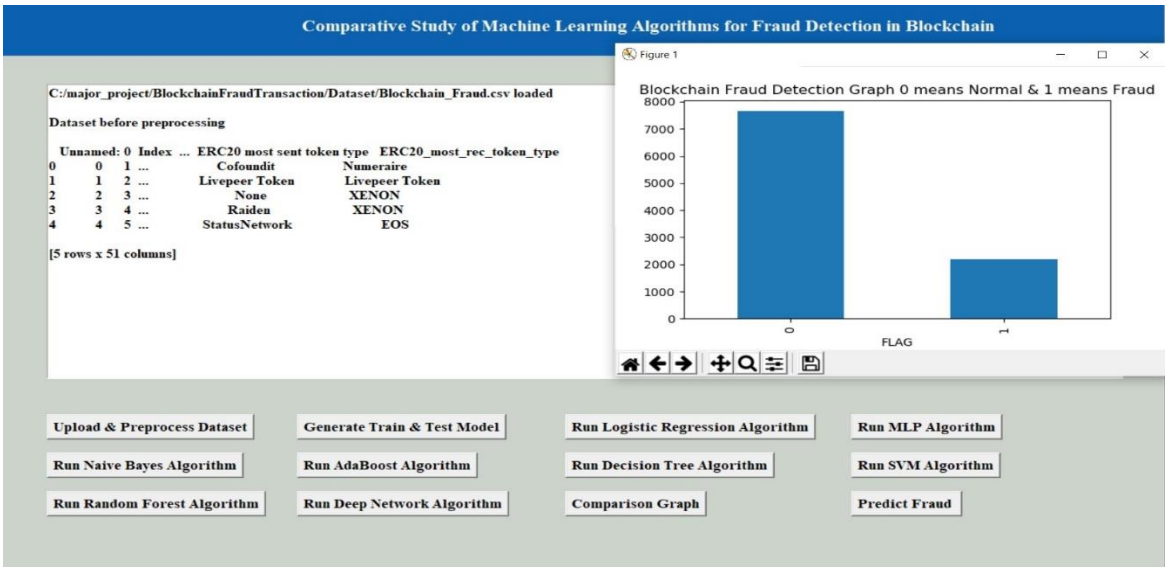


Figure 4.1.2: Dataset before preprocessing

In above screen dataset loaded and dataset contains some non-numeric data and ML algorithms will not take such data so we need to remove and graph x-axis contains type of transaction and y-axis contains number of records click on 'Generate Train & Test Model' button.

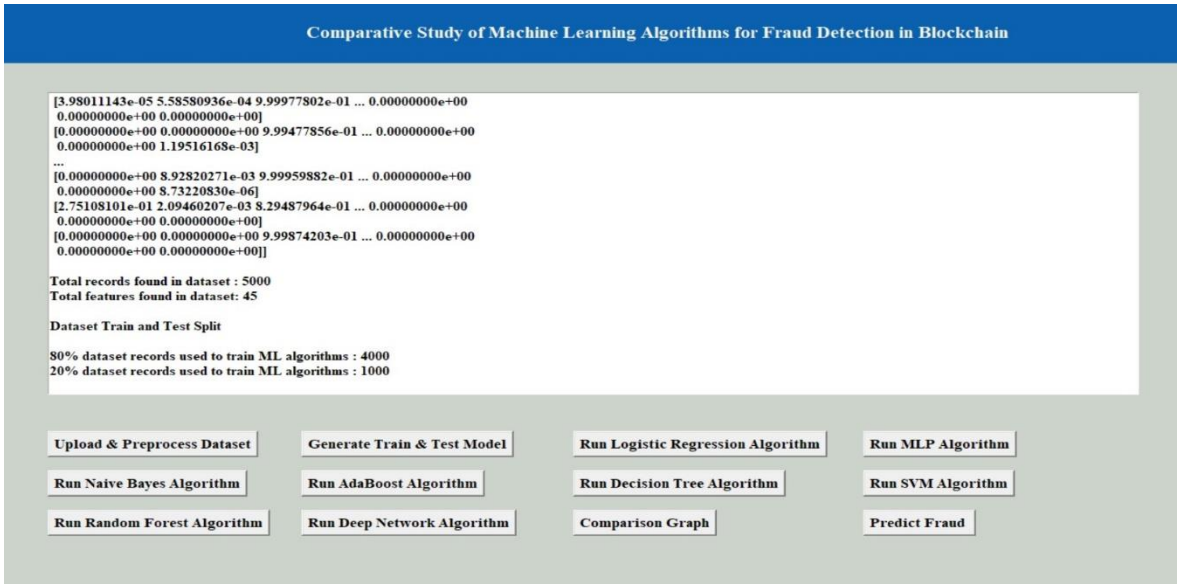


Figure 4.1.3: splitting of dataset

In above screen we can see all data converted to numeric format and we can see total records found in dataset with total columns and then split dataset into train and test and now train and test data is ready and now click on each button to run all algorithms and get below output.

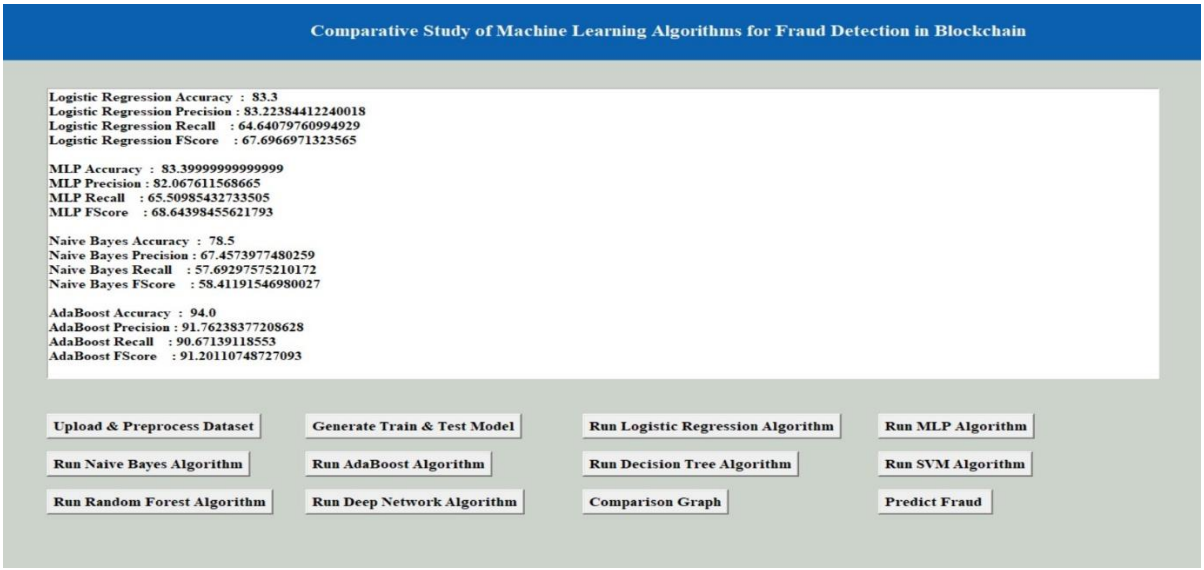


Figure 4.1.4: Performance of first four algorithms

It represents the performance of logistic regression, MLP, Naive Bayes and AdaBoost algorithms.

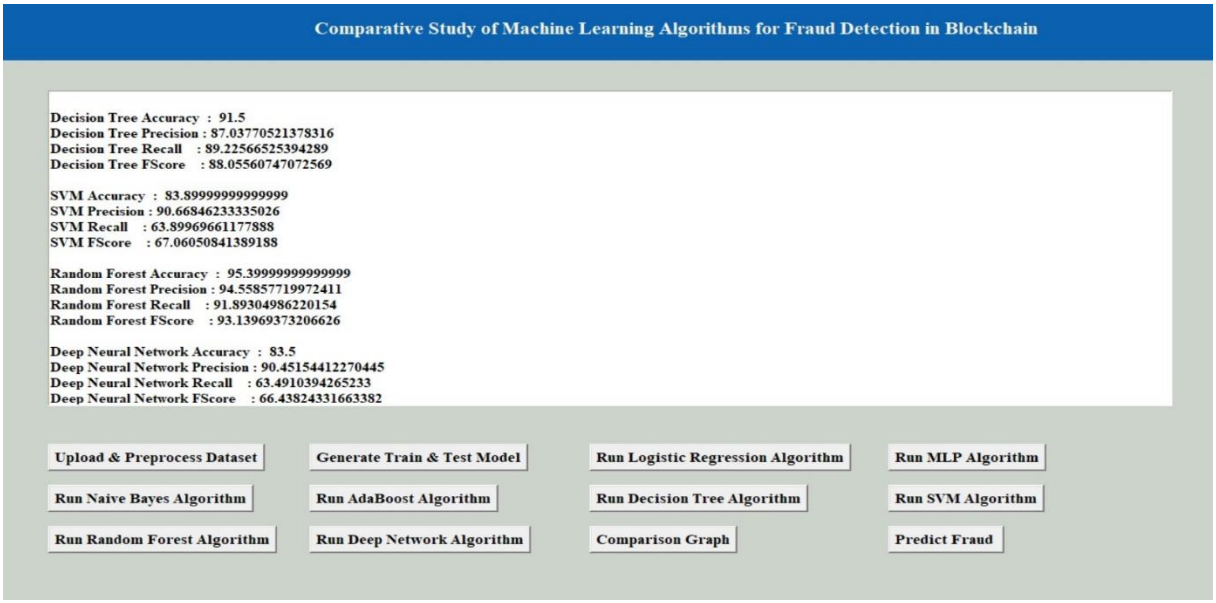


Figure 4.1.5: Performance of other four algorithms

Represents the performance of Decision Tree, SVM, Random Forest and DNN algorithms. Now click on 'Comparison Graph' button to get below output.

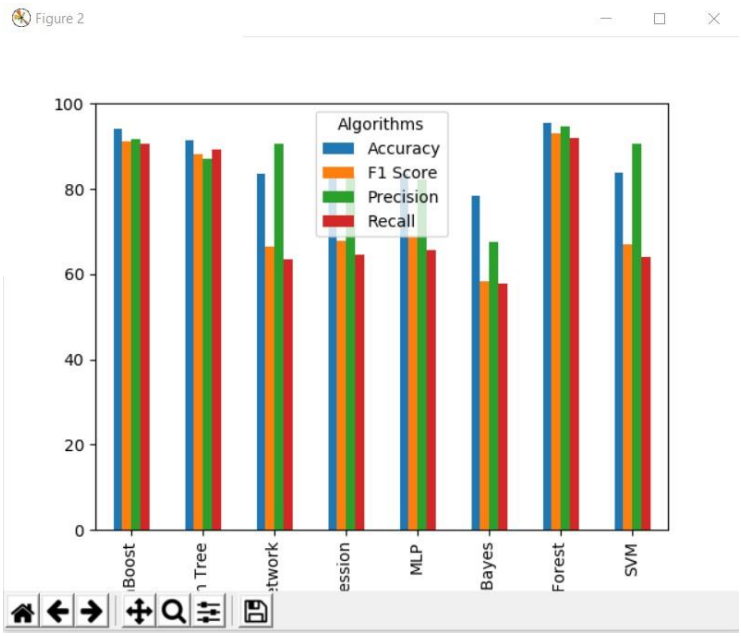


Figure 4.1.6: Graphical representation of performance

In above screen we can see the accuracy, precision, recall and FSCORE of each algorithm in graph and in all algorithms Random Forest giving better result.

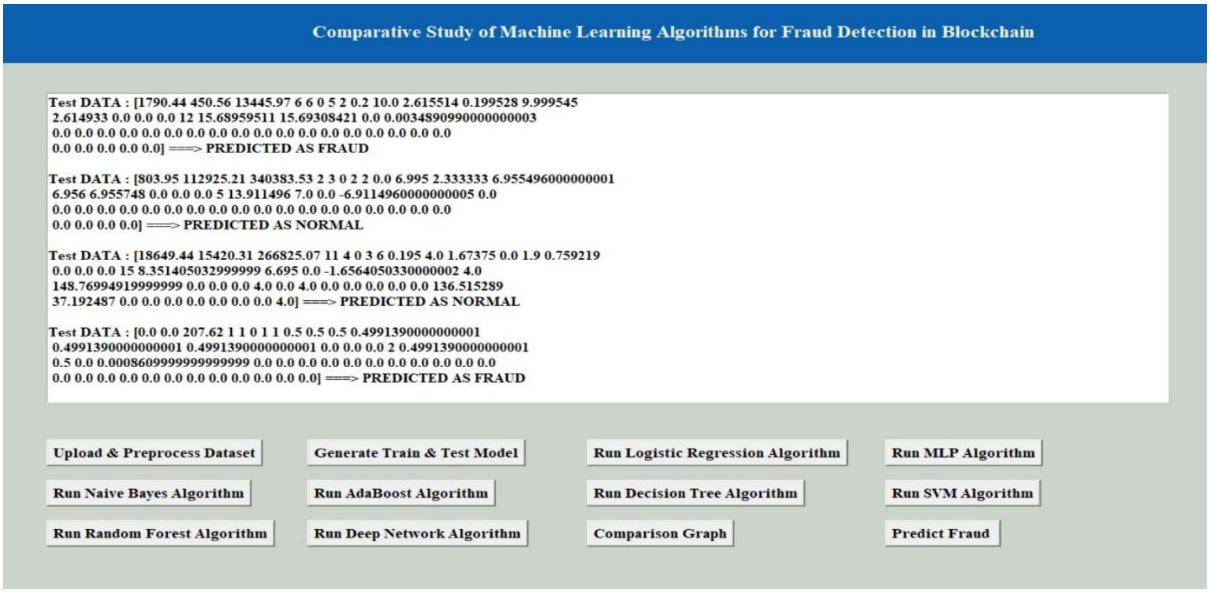


Figure 4.1.7: results of test data

Finally, when we click on predict Fraud button it asks for upload data, by analyzing the this test data it whether the transaction is fraud or not.

4.2 Performance Metrics:

Algorithm Name	Accuracy	Precision	Recall	FSCORE
Logistic Regression Algorithm	83.3	83.22384412240018	64.64079760994929	67.6966971323565
MLP Algorithm	83.39999999999999	82.067611568665	65.50985432733505	68.64398455621793
Naive Bayes Algorithm	78.5	67.4573977480259	57.69297575210172	58.41191546980027
AdaBoost Algorithm	94.0	91.76238377208628	90.67139118553	91.20110748727093
Decision Tree Algorithm	91.5	87.03770521378316	89.22566525394289	88.05560747072569
SVM Algorithm	83.89999999999999	90.66846233335026	63.89969661177888	67.06050841389188
Random Forest Algorithm	95.39999999999999	94.55857719972411	91.89304986220154	93.13969373206626
Deep Neural Network Algorithm	83.5	90.45154412270445	63.4910394265233	66.43824331663382

Figure 4.2.1: Tabular format representation of performance of algorithms

This Table interprets the data of the comparison of the accuracy, precision, recall and fscore of Logistic Regression, MLP, Naive Bayes, AdaBoost, Decision Tree, SVM, Random Forest and Deep Neural Network. We can have a clear view that interms of accuracy and f-score Random Forest has the high performance and in we can have a clear view that interms of accuracy and f-score of Random Forest is best out of all other algorithms.

CHAPTER 5
CONCLUSION

CHAPTER 5

CONCLUSION

5.1 Conclusion and Future Enhancement:

The study introduces a method for detecting fraudulent transactions in blockchain networks using machine learning. By analyzing various supervised learning algorithms—such as support vector machines, decision trees, logistic regression, and dense neural networks—the researchers conducted a comprehensive comparative analysis based on accuracy. The findings highlight the potential of machine learning in enhancing fraud detection within blockchain systems. However, further enhancements and extensions are identified for future research.

Future research endeavors aim to expand the study by incorporating unsupervised algorithms, like clustering, to gain deeper insights into fraudulent activities. Additionally, there are plans to conduct exhaustive investigations into fraudulent behavior within private blockchain networks. Improving model performance, implementing real-time monitoring systems, and addressing privacy and security concerns are also essential areas for future research. These enhancements seek to advance the effectiveness and reliability of fraud detection mechanisms in blockchain environments, ensuring greater security and integrity.

REFERENCES

REFERENCES

[1] Cai, Y., Zhu, D. Fraud detections for online businesses: a perspective from blockchain technology. Financ Innov 2, 20 (2016). https://doi.org/10.1186/s40854-016-0039-4

[2] Hyvarinen, H., Risius, M. & Friis, G. A Blockchain-Based Approach ¨ Towards Overcoming Financial Fraud in Public Sector Services. Bus Inf Syst Eng 59, 441–456 (2017). https://doi.org/10.1007/s12599-017-0502- 4

[3] Xu, J.J. Are blockchains immune to all malicious attacks?. Finance Innov 2, 25 (2016). https://doi.org/10.1186/s40854-016-0046-5

[4] Zero-knowledge proof-of-identity: Sybil-resistant, anonymous authentication on permissionless blockchains and incentive compatible, strictly dominant …DC Sánchez - arXiv preprint arXiv:1905.09093, 2019 - arxiv.org

[5] Ensuring consensus on trust issues in capability-limited node networks with Blockchain technology S Hadjiefthymiades, M Chatzidakis, D Reisis - pergamos.lib.uoa.gr

[6] Comparative study on identity management methods using blockchain AG Nabi - University of Zurich, 2017 - files.ifi.uzh.ch

[7] The blockchain and the new architecture of trust K Werbach - 2018 - books.google.com

[8] Effectiveness of Machine and Deep Learning for Blockchain Technology in Fraud Detection and Prevention

Y Kumar, S Gupta - Applications of Artificial Intelligence, Big Data …, 2022 - taylorfrancis.com

[9] Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019 KG Al-Hashedi, P Magalingam - Computer Science Review, 2021 - Elsevier

[10] Analyzing Various Machine Learning Algorithms for Blockchain-Based Fraud Detection S Giribabu, V Sriharsha, PH BashA Research in, 2022 - acspublisher.com

Github Link:

https://github.com/varunmamidi/Batch-80_Major_Project

DOI:

https://doi.org/10.22214/ijraset.2024.59045

iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** III **Month of publication:** March 2024

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

Examining Various ML Approaches in Blockchain for Fraud Detection

T. Nihith Novah[1], M. Varun[2], K. Pavan Reddy[3], K. Ragini[4]
[1, 2, 3]*UG Student, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana*
[4]*Professor, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana*

Abstract: ***Fraudulent transactions significantly impact blockchain network trust and the economy. Traditional consensus methods (e.g., proof of work or proof of stake) can't confirm the identity of participants, leaving the network susceptible to fraud. Machine learning algorithms offer a potential solution to detect fraudulent transactions and participants. Fraudulent exchanges in the blockchain economy deter investors and raise skepticism. This study explores the effectiveness of controlled AI and deep learning models in identifying fraudulent transactions and users, integrating machine learning with blockchain technology.***
Keywords: Blockchain, Digital Currency, Transactions, Decentralized Network, Fraudulent, Proof of work, Peer-to-Peer Transactions.

I. INTRODUCTION

The persistent issue of identifying fraudulent transactions within blockchain networks has garnered substantial attention over time. Such transactions not only pose a threat to the economy but also undermine trust in cryptocurrencies like bitcoins and other blockchain-based solutions. Fraudulent activities typically raise suspicion either due to the parties involved or the transaction characteristics. Members of blockchain networks are keen on swiftly identifying fraudulent transactions to safeguard the community and preserve the network's integrity. Numerous Machine Learning (ML) techniques have been proposed to tackle this challenge, yielding promising results. However, there lacks a definitive superior method among them. This paper endeavors to compare the efficacy of various supervised ML models, including Support Vector Machines (SVM), Decision Trees, Naive Bayes, Logistic Regression, and several deep learning models, in detecting fraudulent transactions within blockchain networks. Such a comparative analysis aims to discern the optimal algorithm by balancing accuracy and computational speed. The overarching objective is to pinpoint users and transactions with the highest likelihood of involvement in fraudulent activities.

II. EXISTING SYSTEM

1) Synthetic data generation using SMOTE oversamples malicious entities, reducing classification bias. Results, depicted for balanced dataset. Observing log loss during XGBoost training. The small gap between training and test data log loss suggests the model's capability for real-world anomaly detection in blockchain networks.
2) Blockchain technology is a powerful tool for preventing fraud in business networks. It creates an unalterable transaction record, ensures data security, and addresses privacy concerns by anonymizing data and enforcing permission-based access. The consensus process adds an extra layer of validation before transactions are added to the blockchain.
3) Preprocessing and Data Handling Model Building and Evaluation Various machine learning and deep learning techniques are employed to predict transaction success. Models are evaluated using bootstrapping to estimate parameters and determine efficacy based on mean accuracy Values.

III. LITERATURE SURVEY

1) In her research conducted in 2016, Xu delved into the vulnerabilities of blockchain technology to malicious attacks, emphasizing the distinction between identifiable fraudulent activities and those that persist as challenges. She underscored the limitations of blockchain in detecting sophisticated attacks like identity theft and system hacking, which exploit its reliance on predefined rules. Xu's findings underscore the need for the integration of machine learning solutions to enhance the security posture of blockchain systems in mitigating such threats.
2) In their 2019 study, Shi et al. employed transaction aggregation techniques to analyze customer behavior preceding transactions in the Bitcoin market. They aimed to detect fake transactions by examining customer behavior, developing a model capable of identifying anomalies in unknown datasets, including those provided by banks with privacy concerns. The model treated all participant attributes equally without prioritization, effectively distinguishing between legal and fake transactions within improper datasets.

3) Ostapowicz and Zbikowski (2019) employed Supervised Machine Learning methods to detect fraudulent accounts within blockchain systems. Their approach aimed to combat threats such as malware and fake emails, commonly used by malicious actors to pilfer funds. Using a dataset containing over 300,000 accounts, they applied Random Forests, Support Vector Machines, and XGBoost classifiers to identify and flag fraudulent activities.

4) Apruzzese et al. (2020) addressed the vulnerability of intrusion detection systems in cybersecurity, particularly when utilizing datasets with highly sensitive training data. They proposed hardening random forest cyber detectors averse to adversarial attacks to improve cyber-attack detection. Despite the use of random forest algorithms for enhanced detection, there remains scope for further improvement in detecting cyber-attacks.

IV. METHODOLOGY

A. Classification of Attacks

Blockchain technology is often heralded for its robust security in financial transactions. However, despite its advancements, it remains susceptible to contemporary cyber-attacks. Even though the integration of multiple security measures, certain proficient cybercriminals try to orchestrating potent attacks against blockchain networks. Attacks such Sybil attacks, double-spending attacks, Denial of service attacks and remain terrifying challenges to blockchain security. As a result, these attacks are disturbing the blockchain network adversely.

B. Dataset

The dataset comprises various fields including indexes, transaction timestamps, unique addresses, minimum and maximum contract values, transaction types, etc., which are utilized to analyze transaction histories and determine whether transactions are legitimate or fraudulent. It encompasses nearly 9746 records. Using these dataset records, we assess the accuracy, precision, F-score, and recall of different algorithms to determine which algorithm performs the best.

C. Data Analysis

Exploratory Data Analysis (EDA) is a fundamental step in understanding and comprehensively analyzing a dataset. It involves several key tasks aimed at gaining insights into the data's structure and characteristics. Initially, assigning meaningful column names is crucial as it provides important identifiers to each attribute, facilitating easier interpretation and analysis. Subsequently, validating for invalid values ensures that there are no erroneous entries within the dataset, which could otherwise distort analysis outcomes or impede display performance. Moreover, data visualization techniques such as creating plots and charts are employed to visually represent the distribution of data and explore relationships between different features. These visualizations aid in identifying patterns, trends, and anomalies within the dataset, thereby informing subsequent steps in the data analysis process. In our project, it's crucial to factor to know the transaction is fraud or legitimate and verify the transaction history.

D. Algorithms

In our extent, we utilize a differing extend of calculations to address different angles of our issue. Here the data is divided into two sets train (0.8) and test (0.2). The ratio of fraudulent to legitimate transactions is then verified in our train and test sets. By using different ML and Deep Learning models to detect whether the transaction is fraudulent or not. They are Logistic Regression, MLP, Naive Bayes, AdaBoost, Decision Tree, SVM, Random Forest, Deep Network.

E. Implementation Block Diagram

Machine learning algorithms have become increasingly popular in various domains due to their ability to extract insights from data and make predictions. However, implementing these algorithms effectively requires careful consideration of several factors, including data preprocessing, model training, algorithm selection, and performance evaluation. This research paper presents a framework that addresses these challenges by providing a structured approach to implementing machine learning algorithms on datasets. The framework consists of four main modules: Upload & Preprocess Dataset, Generate Train & Test Model, Algorithm Execution and Performance Evaluation, and Comparison Graph. Each module plays a crucial role in different stages of the implementation process, ensuring a systematic and efficient approach to building machine learning models. The Upload & Preprocess Dataset module allows users to upload their datasets and perform preprocessing tasks. This module is essential for ensuring data quality and consistency before training the models. It includes functionality to handle missing values, a common issue in real-world datasets, which can significantly impact model performance if not addressed properly.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

The Generate Train & Test Model module facilitates dataset splitting into training and testing subsets. It provides users with essential information about the dataset, such as the total number of records and columns, enabling them to gain insights into the data's characteristics. This module sets the foundation for model training and evaluation by preparing the data for further analysis. The Algorithm Execution and Performance Evaluation module allow users to run various machine learning algorithms and assess their performance. Users can execute different algorithms and compare their accuracy scores to determine the most suitable model for their dataset. This module provides valuable insights into the strengths and weaknesses of each algorithm, enabling informed decision-making in algorithm selection. The Comparison Graph module generates visual representations of algorithm performance, facilitating easier analysis and interpretation of results. By visualizing the accuracy scores of different algorithms, users can identify trends and patterns that may not be apparent from numerical data alone. This module enhances the overall usability of the framework by providing intuitive visualizations for better decision-making.

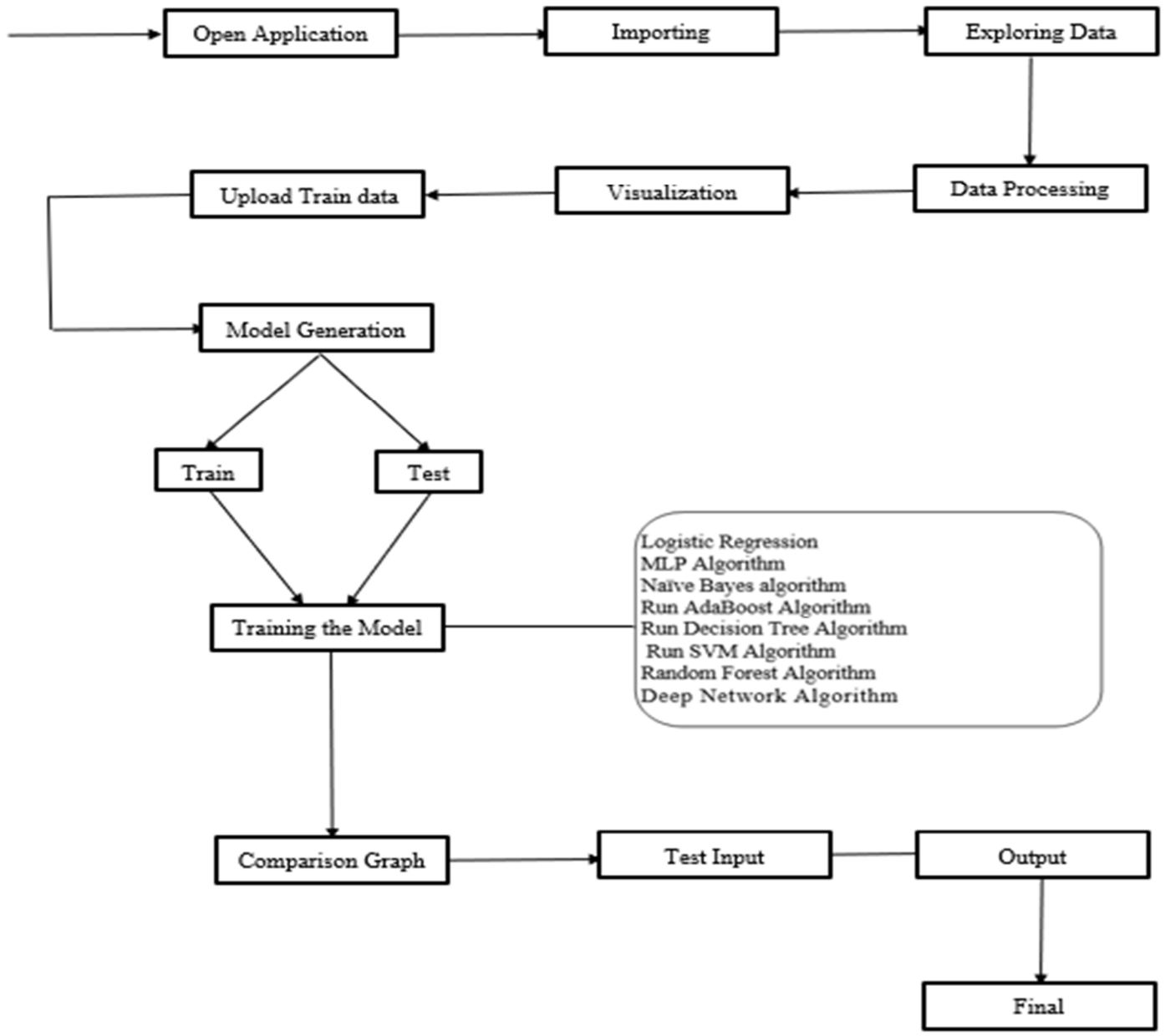
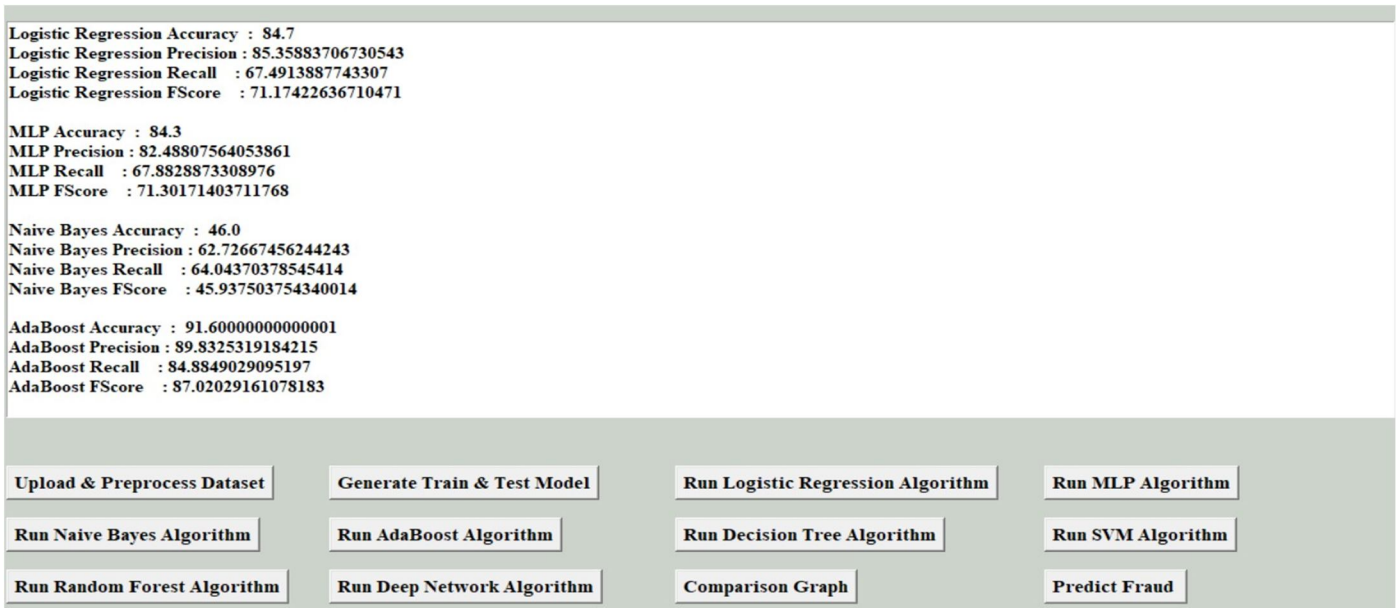


Fig. Architecture of the proposed model

V. PERFORMANCE

A. *Performance Validation Of Machine Learning Algorithms*



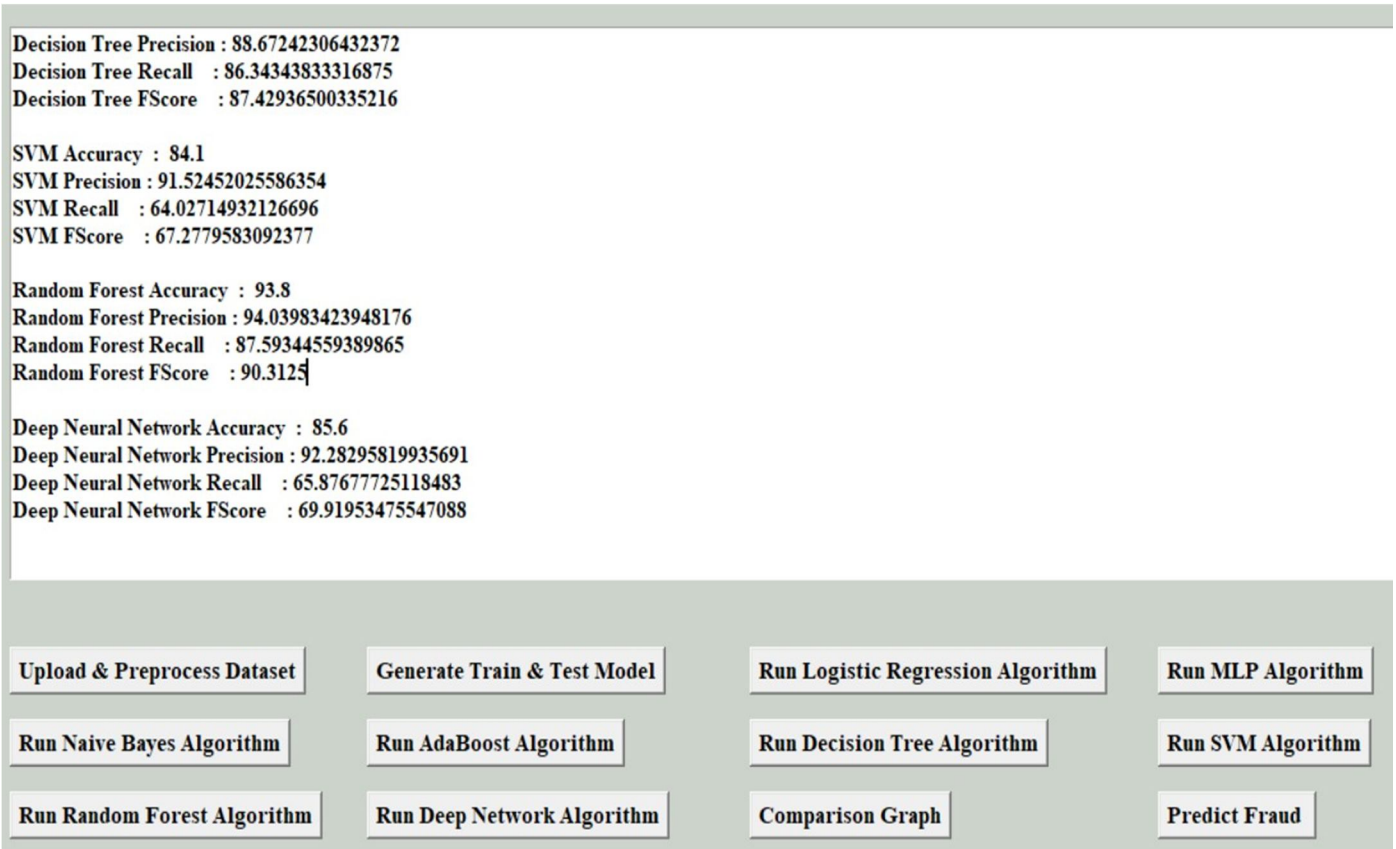


Fig 2: Performance Metrics 2

VI. RESULTS AND DISCUSSION

A. *Comparision Algorithm*

This Table interprets the data of the comparison of the accuracy, precision, recall and fscore of Logistic Regression, MLP, Naïve Bayes, AdaBoost, Decision Tree, SVM, Random Forest and Deep Neural Network.

Algorithm Name	Accuracy	Precision	Recall	FSCORE
Logistic Regression Algorithm	83.2	86.80351551638681	66.54978771520676	69.80828259447492
MLP Algorithm	83.5	83.80955209694665	68.56418285304238	71.86945380708582
Naive Bayes Algorithm	49.6	61.845714912761984	63.497344401498225	49.44144838212634
AdaBoost Algorithm	92.10000000000001	90.04081772655553	88.07590064745503	88.99781209655744
Decision Tree Algorithm	91.9	89.031051701956	88.92177808220667	88.97621720934981
SVM Algorithm	82.5	90.61158798283262	63.991769547325106	66.68437154350354
Random Forest Algorithm	94.19999999999999	94.06772025300488	89.882088164783	91.7365262635991
Deep Neural Network Algorithm	84.8	91.16875634864083	65.77291922002335	69.52295402778223

Table 1: Comparison Table

B. *Figures*

It represents the comparison bar graph of various Machine learning algorithms. X-axis represents the different algorithms like AdaBoost, Decision Tree, MLP, Random Forest, SVM, Naive Bayes, Logistic

We can have a clear view that interms of accuracy and f-score Random Forest has the high performance and in we can have a clear view that interms of accuracy and f-score of Random Forest is best out of all other algorithms.

Regression and Deep Neural Network Algorithm. Y-axis represents the scale of each algorithm. Here, we are calculating Accuracy, Precision, F1 score and recall for each algorithm to prove which is performing best.

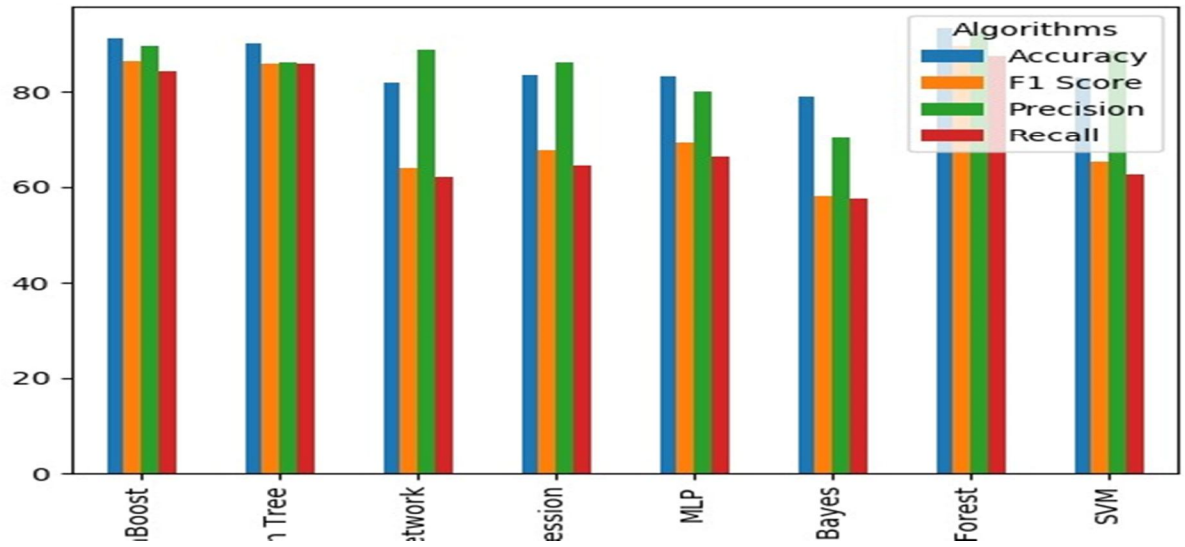


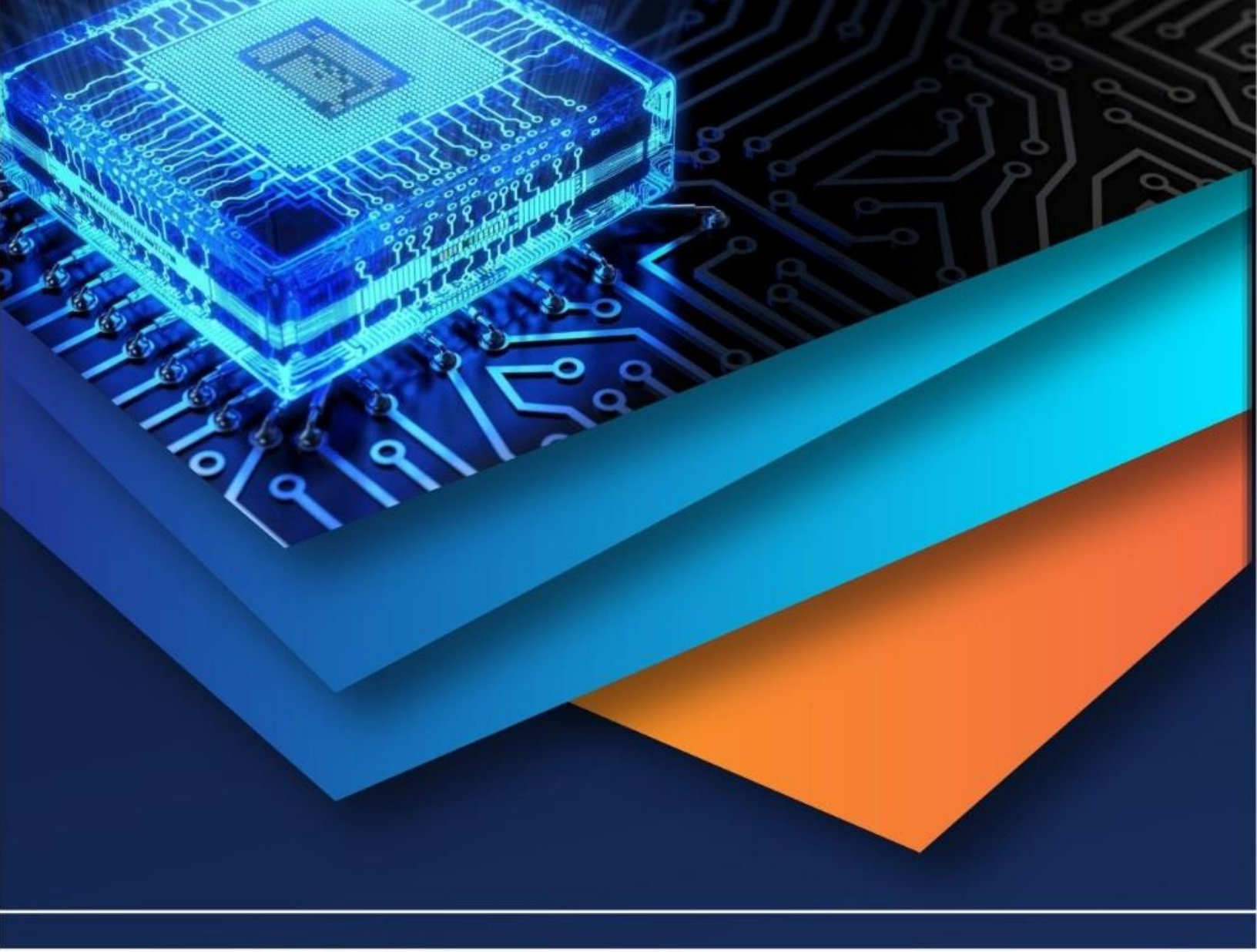
Fig. 3: Graphical representation of Comparison of models

VII. CONCLUSION

A study explored the use of machine learning to detect fraudulent transactions within a blockchain network. The researchers examined various supervised learning algorithms like support vector machines, decision trees, logistic regression, and dense neural networks, comparing their performance in terms of accuracy. They suggested extending the study to include unsupervised techniques such as clustering. Additionally, the researchers expressed intentions to conduct a detailed investigation into fraudulent activities within a private blockchain network in the future.

REFERENCES

[1] Zero-knowledge proof-of-identity: Sybil-resistant, anonymous authentication on permissionless blockchains and incentive compatible, strictly dominant …DC Sánchez - arXiv preprint arXiv:1905.09093, 2019 - arxiv.org
[2] Ensuring consensus on trust issues in capability-limited node networks with Blockchain technology S Hadjiefthymiades, M Chatzidakis, D Reisis - pergamos.lib.uoa.gr
[3] Comparative study on identity management methods using blockchain AG Nabi - University of Zurich, 2017 - files.ifi.uzh.ch
[4] The blockchain and the new architecture of trust K Werbach - 2018 - books.google.com
[5] Effectiveness of Machine and Deep Learning for Blockchain Technology in Fraud Detection and Prevention Y Kumar, S Gupta - Applications of Artificial Intelligence, Big Data …, 2022 - taylorfrancis.com
[6] Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019 KG Al-Hashedi, P Magalingam - Computer Science Review, 2021 – Elsevier
[7] Analyzing Various Machine Learning Algorithms for Blockchain-Based Fraud Detection S Giribabu, V Sriharsha, PH Basha… - … Research in …, 2022 - acspublisher.com
[8] https://learnprompting.org/docs/basics/instructions
[9] https://chat.openai.com/
[10] https://bard.google.com/chat



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET doi cross ref

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59045, entitled

Examining Various ML Approaches in Blockchain for Fraud Detection

by

T. Nihith Novah

after review is found suitable and has been published in Volume 12, Issue III, March 2024

in

International Journal for Research in Applied Science & Engineering Technology

(International Peer Reviewed and Refereed Journal)

Good luck for your future endeavors

Editor in Chief, iJRASET

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL

SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59045, entitled

Examining Various ML Approaches in Blockchain for Fraud Detection

by

M. Varun

after review is found suitable and has been published in
Volume 12, Issue III, March 2024
in

International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL

SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59045, entitled

Examining Various ML Approaches in Blockchain for Fraud Detection

by

K. Pavan Reddy

after review is found suitable and has been published in
Volume 12, Issue III, March 2024
in

International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET

ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL

SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59045, entitled

Examining Various ML Approaches in Blockchain for Fraud Detection

by

K. Ragini

after review is found suitable and has been published in
Volume 12, Issue III, March 2024
in

International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET