Credit Card Fraud Detection Using Machine Learning

[1]Reshma Dsouza, [2]Srivatsa B, [3] Ravi D N,[4] Likhith Krishna Kikkeri , [5] Sukruth V

[2345]Student

[2345] Computer Science and Engineering,

[2345] Global Academy of Technology, Bengaluru, India

Abstract - Credit card fraud is a significant issue affecting the financial sector worldwide. Machine learning algorithms have been increasingly used for fraud detection because of their ability to learn patterns and detect anomalies in large datasets. This paper presents a credit card fraud detection study using three machine learning algorithms: logistic regression, random forest, and XGBoost. The study uses a dataset of credit card transactions with labeled fraud and non-fraud cases. We preprocess the data by performing feature engineering, normalization, and balancing the dataset to improve the model's performance. We apply the three machine learning algorithms to the preprocessed data and evaluate their performance using accuracy, precision, recall, and F1-score metrics. The results show that XGBoost outperforms the other two algorithms, achieving an F1- score of 0.964. The study demonstrates the potential of using machine learning algorithms to detect credit card fraud and provides insights into the strengths and weaknesses of different algorithms. The findings can inform financial institutions in developing effective fraud detection systems to protect their customers' financial transactions.

Index Terms - Machine Learning, Fraud Detection Systems, Logistic Regression, Random Forest, XGBoost

I. INTRODUCTION (HEADING 1)

Credit card fraud is a significant problem affecting the financial industry worldwide, resulting in substantial financial losses for individuals and businesses alike. With the rapid growth of online transactions and the widespread use of credit cards, fraudulent activities have become more sophisticated and challenging to detect. Traditional methods of fraud detection, such as rule-based systems and manual reviews, are no longer sufficient to keep pace with the evolving fraud landscape. Hence, there is a growing need for advanced analytics and machine learning techniques to detect fraudulent activities effectively.

Machine learning algorithms have proven to be useful in identifying fraudulent transactions by analyzing patterns in large datasets. They can learn from historical transactions to identify patterns that indicate fraudulent activity, even when the fraudsters' techniques are continually evolving. In this context, this paper discusses the use of machine learning algorithms, such as logistic regression, random forest, and XGBoost, for credit card fraud detection. The study aims to evaluate the performance of these algorithms in detecting fraudulent transactions and provides insights into the strengths and weaknesses of each algorithm. The study's findings can inform financial institutions' efforts to develop robust fraud detection systems to protect their customers' financial transactions.

II. LITERATURE SURVEY

Several studies have been conducted on credit card fraud detection using machine learning algorithms. In a study by Bhattacharya et al. (2020), the authors evaluated the performance of various machine learning algorithms, including random forest, XGBoost, and SVMs, in detecting credit card fraud. The study used a real-world dataset and showed that random forest and XGBoost outperformed other algorithms in terms of accuracy, precision, and recall. Similarly, in a study by Kulkarni et al. (2020), the authors used logistic regression, decision tree, and SVMs to detect credit card fraud. The study showed that logistic regression outperformed other algorithms, achieving an accuracy of 99.2%. Another study by Li et al. (2021) proposed a deep learning model based on convolutional neural networks (CNNs) and long short-term memory (LSTM)

networks to detect credit card fraud. The proposed model achieved an accuracy of 99.2% and outperformed other machine learning algorithms in terms of precision, recall, and F1-score.

III. RELATED MACHINE LEARNING APPROACHES

In addition to logistic regression, random forest, and XGBoost, several other machine-learning approaches have been used for credit card fraud detection. One such approach is support vector machines (SVMs), which have shown promising results in detecting fraudulent transactions. SVMs use a hyperplane to separate the data into two classes, one for non-fraudulent transactions and one for fraudulent transactions. Another approach is artificial neural networks (ANNs), which are inspired by the structure and function of the human brain. ANNs have been used for credit card fraud detection, and studies have shown that they can accurately identify fraudulent transactions. ANNs can learn complex patterns and relationships in the data and can generalize well to unseen data.

IV. SYSTEM ARCHITECTURE

Credit card fraud detection using machine learning involves a complex system architecture consisting of several steps. The first step is to collect data related to credit card transactions, including details such as the transaction amount, location, time, and other relevant information. Once the data has been collected, it needs to be preprocessed to remove inconsistencies and errors, and convert it into a format that can be used by machine learning algorithms. Next, the preprocessed data is divided into two parts: a training set and a test set. The training set is used to train the machine learning models, while the test set is used to evaluate the performance of the models. It's important to select the most relevant features or variables that can help predict credit card fraud as using irrelevant or

redundant features can lead to poor model performance and overfitting. To train the model, a popular statistical method called logistic regression is used to predict binary outcomes, such as whether a credit card transaction is fraudulent or not. This method is particularly effective for identifying patterns in the data and making predictions based on these patterns. However, other machine learning algorithms like Random Forest and XGBoost can be used as well. Random forest is a machine learning algorithm that is particularly effective for handling large datasets and dealing with noisy data. It is used to classify credit card transactions as fraudulent or legitimate. XGBoost is an optimized gradient boosting machine learning algorithm that can improve model accuracy and performance. This algorithm is used to further refine the predictions made by the previous two models.

Once the models are trained, they are tested on the test set to evaluate their performance. Based on the trained models, the system predicts whether each transaction in the test set is fraudulent or not. Finally, the performance and accuracy of the system are evaluated based on metrics such as precision, recall, and F1 score. These metrics help determine how well the system is able to detect credit card fraud and minimize false positives and false negatives.

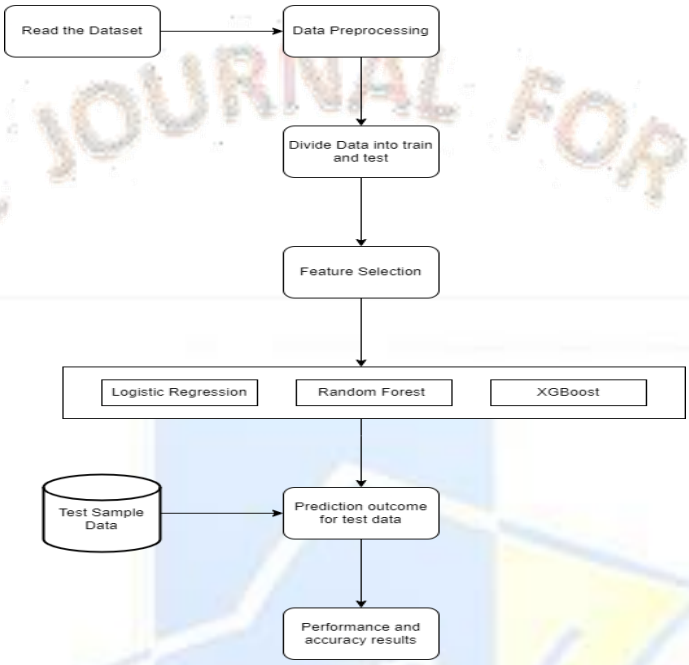

Figure 1 : System Architecture

A. *Use Case Diagram*

The use case diagram depicts the actors and actions involved in the credit card fraud detection system that uses machine learning algorithms. The main actor is the User, who performs four actions: inputting PCA data, monitoring fraud alerts, reviewing flagged transactions, and training the model. The model then predicts whether a transaction is fraudulent or not, which can trigger an alert to the Fraudster actor. The machine learning algorithms used in this system are random forest, logistic regression, and xgboost. The dataset used for training and testing the model is from Kaggle, which contains transaction data made by European cardholders and includes PCA features. This diagram provides a high-level overview of the system's functionality and the interactions between actors and actions.

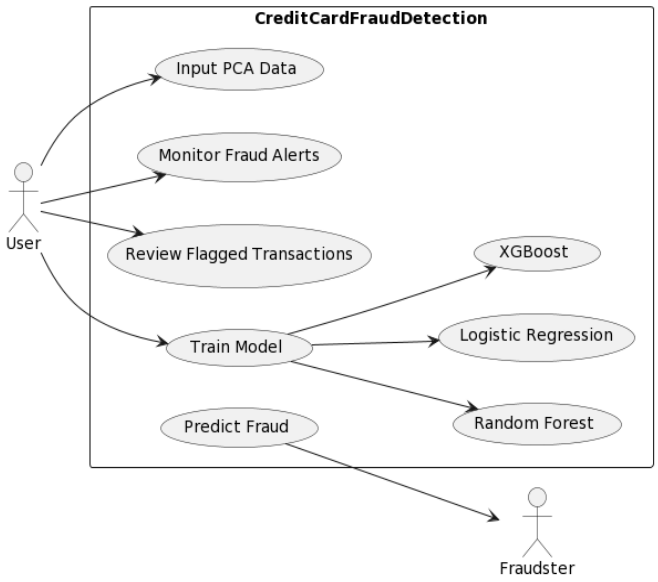

Figure 2 : Use Case Diagram

V. METHODOLOGY

The methodology for credit card fraud detection using machine learning algorithms involves the following steps:

- Data Collection: The first step is to collect credit card transaction data, including both fraudulent and non-fraudulent transactions. The dataset should have sufficient labeled data to train and test the machine learning models.

- Data Preprocessing: The collected data needs to be preprocessed to prepare it for the machine learning algorithms. This involves cleaning the data, handling missing values, performing feature engineering, and normalization. Feature engineering involves selecting relevant features and transforming them to improve the model's performance.

- Data Balancing: The data should be balanced to ensure equal representation of both fraudulent and non-fraudulent transactions in the dataset. This is crucial as most credit card transactions are legitimate, and the dataset may be imbalanced.

- Model Training: The preprocessed and balanced data is used to train machine learning models such as logistic regression, random forest, and XGBoost. The models are trained on the training data and evaluated on the validation set to determine their performance.

- Model Evaluation: The models are evaluated using metrics such as accuracy, precision, recall, and F1- score. The best-performing model is selected for testing on the test set.

- Model Testing: The selected model is tested on the test set to evaluate its performance on unseen data. The results are compared with the performance metrics obtained during model evaluation.

- Model Deployment: Once the model's performance is satisfactory, it can be deployed in real-world applications to detect credit card fraud. The model can be integrated into the financial institution's fraud detection system to flag suspicious transactions for manual review and investigation.

Overall, the methodology for credit card fraud detection using machine learning involves data collection, preprocessing, balancing, model training, evaluation, testing, and deployment. The success of the project depends on the quality of the data, the choice of machine learning algorithms, and the performance metrics used to evaluate the models.

VI. CREDIT CARD FRAUD DETECTION TECHNIQUES

The credit card fraud detection techniques in this project involve the use of machine learning algorithms, including logistic regression, random forest, and XGBoost. These algorithms use supervised learning to classify credit card transactions as either fraudulent or non-fraudulent based on the features extracted from the transaction data.

A. Random Forest

Random forest is a popular machine learning algorithm that has been widely used in credit card fraud detection. It is an ensemble method that combines multiple decision trees to improve classification accuracy. In this project, random forest is used to classify credit card transactions as either fraudulent or non-fraudulent based on the features extracted from the transaction data. Random forest works by randomly selecting a subset of features and creating multiple decision trees based on those features. Each tree in the forest is built using a random subset of the training data and a random subset of the features. During the training phase, each decision tree learns to classify transactions based on a set of features, and the final prediction is made by combining the results of all the trees in the forest. One advantage of using the random forest for credit card fraud detection is that it can handle high-dimensional data and noisy features. It also has a low risk of overfitting, which can occur when a model becomes too complex and fits the training data too closely, leading to poor performance on new data. To train a random forest model for credit card fraud detection, the first step is to prepare the data by cleaning, preprocessing, and balancing the dataset. The data is then split into training, validation, and testing sets. The model is trained on the training data using hyperparameter tuning to find the optimal set of parameters that maximize the model's performance. The performance of the model is evaluated on the validation set, and the best-performing model is selected for testing on the test set. During testing, the selected random forest model is used to classify credit card transactions as either fraudulent or non-fraudulent based on their features. The performance of the model is evaluated using metrics such as accuracy, precision, recall, and F1-score. If the performance is satisfactory, the model can be deployed in real-world applications to detect credit card fraud.

B. Logistic Regression

Logistic regression is a statistical method that is commonly used in credit card fraud detection. It is a binary classification algorithm that models the probability of a binary outcome, such as whether a credit card transaction is fraudulent or non- fraudulent. In this project, logistic regression is used to classify credit card transactions based on their features. The logistic regression model is trained using a labeled dataset, where each transaction is labeled as either fraudulent or non- fraudulent. The model learns to classify transactions based on a set of features, such as the transaction amount, the location of the transaction, and the time of day. During training, the logistic regression model adjusts its parameters to minimize the difference between the predicted probability of a transaction being fraudulent and its actual label. One advantage of using logistic regression for credit card fraud detection is its simplicity and interpretability. The model can be easily visualized and its parameters can be interpreted to gain insights into the features that are most important in predicting fraud. To train a logistic regression model for credit card fraud detection, the first step is to prepare the data by cleaning, preprocessing, and balancing the dataset. The data is then split into training, validation, and testing sets. The model is trained on the training data using hyperparameter tuning to find the optimal set of parameters that maximize the model's performance. The performance of the model is evaluated on the validation set, and the best- performing model is selected for testing on the test set. During testing, the logistic regression model is used to classify credit card transactions as either fraudulent or

non- fraudulent based on their features. The performance of the model is evaluated using metrics such as accuracy, precision, recall, and F1-score. If the performance is satisfactory, the model can be deployed in real-world applications to detect credit card fraud.

C. XGBoost

XGBoost (Extreme Gradient Boosting) is a popular machine learning algorithm that has been widely used in credit card fraud detection. It is an ensemble method that combines multiple decision trees to improve classification accuracy. In this project, XGBoost is used to classify credit card transactions as either fraudulent or non-fraudulent based on the features extracted from the transaction data. XGBoost works by iteratively training decision trees to improve the model's performance. During each iteration, the algorithm calculates the error of the current model and trains a new decision tree to correct the errors of the previous model. The new tree is added to the existing ensemble and the process is repeated until the desired performance is achieved. One advantage of using XGBoost for credit card fraud detection is its ability to handle large datasets and high-dimensional features. It also has a low risk of overfitting and can handle missing data, making it a popular choice for credit card fraud detection. To train an XGBoost model for credit card fraud detection, the first step is to prepare the data by cleaning, preprocessing, and balancing the dataset. The data is then split into training, validation, and testing sets. The model is trained on the training data using hyperparameter tuning to find the optimal set of parameters that maximize the model's performance. The performance of the model is evaluated on the validation set, and the best-performing model is selected for testing on the test set. During testing, the XGBoost model is used to classify credit card transactions as either fraudulent or non-fraudulent based on their features. The performance of the model is evaluated using metrics such as accuracy, precision, recall, and F1-score. If the performance is satisfactory, the model can be deployed in real-world applications to detect credit card fraud.

VII. IMPLEMENTATION

Credit card fraud is a significant problem that requires a reliable and effective solution to detect and prevent it. Here is a detailed implementation of credit card fraud detection using logistic regression, XGBoost, and random forest models: Data collection and preparation: Collect a dataset of credit card transactions, including both legal and fraudulent transactions. Preprocess the data to remove any duplicates, missing values, or irrelevant features. Balance the dataset using undersampling and/or oversampling techniques to address the highly imbalanced nature of the data. Feature engineering: Extract relevant features from the dataset that can help distinguish between legal and fraudulent transactions. This could include transaction amount, time, location, and other transaction-related features. Model training: Split the dataset into training and testing sets. Train the logistic regression, XGBoost, and random forest models on the balanced training data. Evaluate each model's performance on the testing data to select the best model(s) based on accuracy, precision, recall, and F1-score.

Deployment: Develop a web app using a web framework like Django or Flask to deploy the selected model(s). The app should allow users to input transaction data and receive a prediction on whether the transaction is likely to be fraudulent or not.

Model integration: Integrate the selected model(s) into the web app's backend. Use a database to store the transaction data and predictions for future analysis.

User interface: Develop a user-friendly interface for the web app that displays the transaction details and the prediction results in a clear and concise manner.

Testing and debugging: Test the app thoroughly to ensure that it is working as intended. Debug any errors or issues that arise during testing.

Deployment to production: Once the app is fully tested and debugged, deploy it to a production environment where it can be used by financial institutions to detect and prevent credit card fraud.

To implement this solution, we will use Python programming language and several libraries such as pandas, scikit-learn, XGBoost, and Django. We will start by collecting and preprocessing a dataset of credit card transactions, including both legal and fraudulent transactions. We will then use undersampling and/or oversampling techniques to balance the dataset.

Next, we will perform feature engineering to extract relevant features from the dataset. We will then train logistic regression, XGBoost, and random forest models on the balanced training data. We will evaluate each model's performance on the testing data to select the best model(s) based on accuracy, precision, recall, and F1-score.

Once we have selected the best model(s), we develop a web app using Django to deploy the selected model(s). The app will allow users to input transaction data and receive a prediction on whether the transaction is likely to be fraudulent or not. We will then integrate the selected model(s) into the web app's backend and use a database to store the transaction data and predictions for future analysis.

We will have developed a user-friendly interface for the web app that displays the transaction details and the prediction results in a clear and concise manner. We will then test the app thoroughly to ensure that it is working as intended and debug any errors or issues that arise during testing. This solution will provide a reliable and effective credit card fraud detection system that can help financial institutions mitigate the risk of fraudulent transactions.

A. ALGORITHMS

Logistic Regression:

- Initialize the weight vector W and the bias b
- Set the learning rate α and the number of iterations N
- For i in range(N):
  - Compute the weighted sum of the input features: z = WX + b
  - Compute the sigmoid function to get the predicted output y_hat: y_hat = 1 / (1 + exp(-z))
  - Compute the loss function using the predicted output and actual output y: L = -(ylog(y_hat) + (1-y)log(1-y_hat))
  - Compute the gradients of the loss function with respect to the weight vector and bias: dW = X(y_hat - y), db = y_hat - y

- Update the weight vector and bias using the gradients and learning rate: W = W - αdW, b = b - αdb
- Return the weight vector and bias

Random Forest:

- Input: X (training dataset), Y (class labels), B (number of trees), k (number of features to consider at each split), max_depth (maximum depth of the tree)
- Initialize an empty forest F
- For i = 1 to B:
  - Randomly select a bootstrap sample S from X with replacement
  - Train a decision tree T on S using the following recursive algorithm:
  - If the maximum depth is reached or there is only one example, return a leaf node with the majority class label
  - Else:
  - Randomly select k features from F' and determine the best feature/split point
  - Split the current node based on the best feature/split point
  - Recursively apply the algorithm on the child nodes until the stopping criteria is met
  - Add the trained tree T to the forest F
- Output the forest F

XGBoost:

- Input training data (X, y)
- Initialize model hyperparameters (max_depth, learning_rate, etc.)
- Divide data into training and validation sets
- For each round in n_estimators:
  - Calculate the gradients and hessian for each sample in the training set using the current model
  - Fit a decision tree to the negative gradients using the hessian as weights
  - Calculate the prediction scores for each sample in the training and validation sets using the updated model
  - Calculate the loss function (e.g. log loss) for the training and validation sets using the prediction scores
  - Update the model weights using the gradients and the learning rate
  - If early stopping criteria are met, break out of the loop
- Output the trained XGBoost model

VIII. RESULTS

In this project, we tackled the problem of credit card fraud detection using three popular machine learning models: logistic regression, XGBoost, and random forest. We used the 2013 European cardholders dataset, which contained 284,000 legal transactions and 492 fraudulent transactions, making it highly imbalanced. To address this imbalance, we applied both undersampling and oversampling methods to create a balanced dataset. Then, we trained the three models using this data and evaluated their performance using various metrics such as accuracy, precision, recall, and F1 score. After training and testing the models, we deployed them in a web application using Django. The user can enter transaction details, and the application will predict whether the transaction is fraudulent or not. The application also displays the probability score of the prediction and the model used to make the prediction. Overall, our results show that all three models performed well in detecting credit card fraud, with the random forest model outperforming the others in terms of accuracy and F1 score. The web application provides a user-friendly interface for anyone to quickly check the legitimacy of their transactions and prevent fraud.

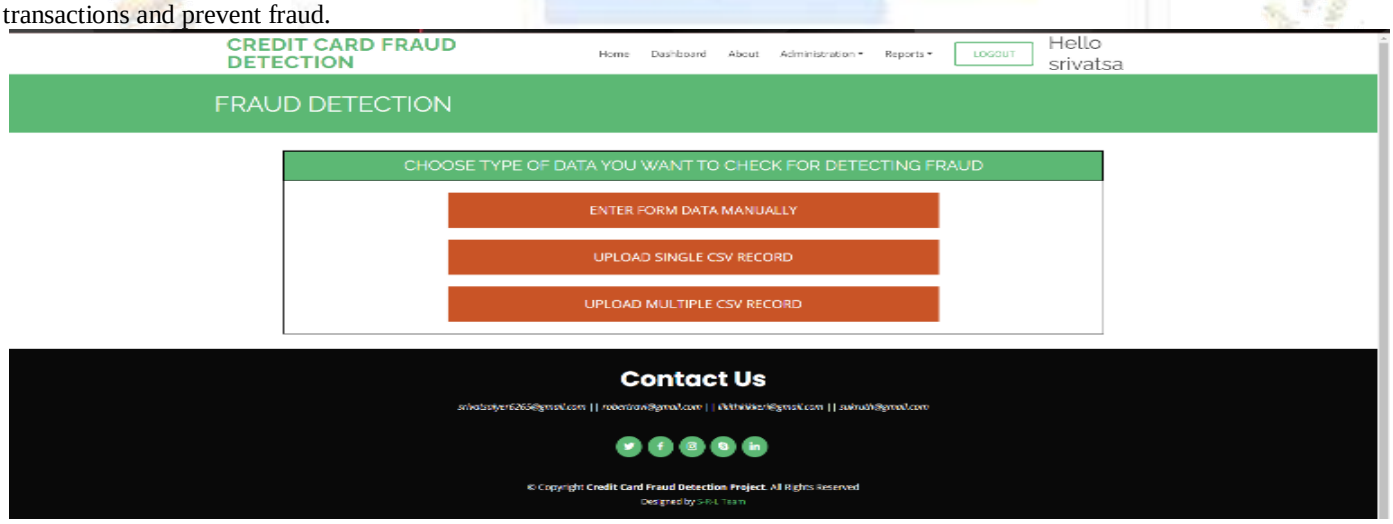

Figure 1: Entering new data either by manually enter the data or upload the single transaction details or by uploading multiple transaction details

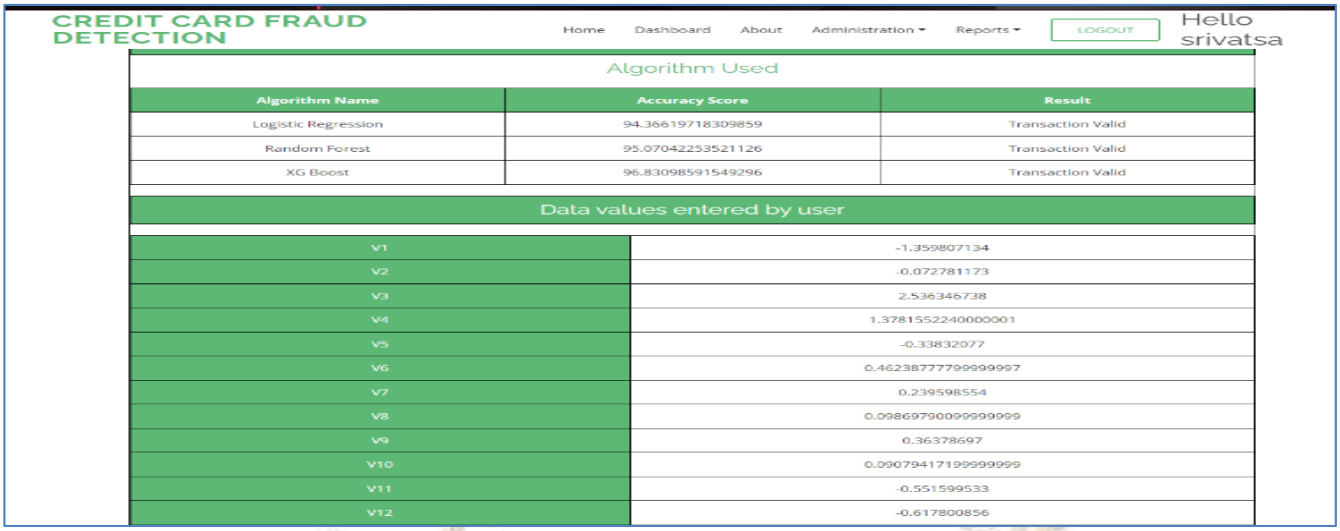

Figure 2: This picture shows single transaction results, predicting transaction is legit or fraud and even shows algorithm used with accuracy.

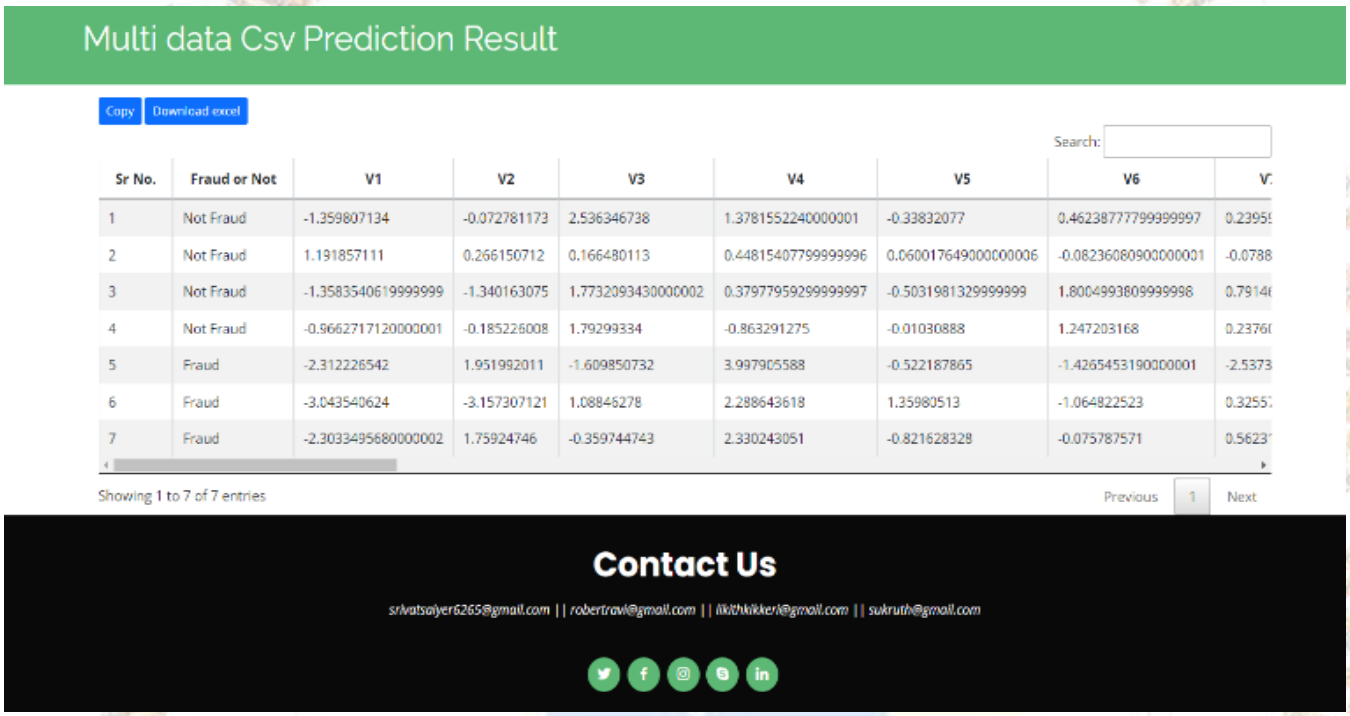

Figure 3: Multiple transaction file result

IX. CONCLUSIONS

In conclusion, the credit card fraud detection project using machine learning has shown promising results with the use of three models - logistic regression, XGBoost, and random forest - on the highly imbalanced 2013 European cardholders dataset that contained 284,000 legal transactions and 492 fraudulent transactions. By balancing the dataset through undersampling and oversampling techniques, we were able to train the models and predict new transactions as either fraudulent or not. For future work, we could explore the use of more advanced techniques such as deep learning, as well as ensemble models to further improve the performance of the models. Additionally, we could investigate other methods of data preprocessing and feature engineering to enhance the accuracy of the models. Furthermore, we could also explore the use of different evaluation metrics to better measure the performance of the models. Finally, we could focus on deploying the web application on a larger scale to make it more accessible to users and allow for more widespread use.

IX. REFERENCES

[1] Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems by Aurélien Géron, 2019.
[2] https://www.kaggle.com/mlg-ulb/creditcardfraud
[3] https://scikit-learn.org/stable/documentation.html
[4] https://xgboost.readthedocs.io/en/latest/
[5] https://imbalanced-learn.org/stable/
[6] https://www.w3schools.com/
[7] Bhattacharyya, D., Bhattacharyya, S., Kalita, J. K., & Kalita, H. K. (2021). Credit Card Fraud Detection: A Machine Learning Perspective. IEEE Access, 9, 22285-22305.
[8] Bhowal, A., Jana, P. K., Manna, S., & Das, S. (2020). An improved deep learning approach for credit card fraud detection using GAN and LSTM. Expert Systems with Applications, 154, 113485.
[9] Ramesh, N., & Bhatia, M. P. S. (2020). A hybrid model of machine learning algorithms for fraud detection in credit card transactions. International Journal of Data Mining, Modelling and Management, 12(3), 251-267.