International Scientific Journal of Engineering and Management
Volume: 03 Issue: 12 | Dec – 2024
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM02166

FRAUD PULSE

P. Rajapandian [1], R. Ramyasri [2], A. Deepa Shivani [3]

[1]Associate Professor, Department of Computer Applications, Sri Manakula Vinayagar Engineering College (Autonomous), Puducherry 605008, India,
rajapandian.mca@smvec.ac.in

[2]Post Graduate student, Department of Computer Applications, Sri Manakula Vinayagar Engineering College (Autonomous),
Puducherry 605008, India p.ramyasri2002@gmail.com

[3]Post Graduate student, Department of Computer Applications, Sri Manakula Vinayagar Engineering College (Autonomous),
Puducherry 605008, India deepashivaniaroumougam@gmail.com

Abstract
The proliferation of e-commerce and advanced communication technologies has made credit cards the dominant payment method globally, both online and offline. This surge in credit card usage, however, has concurrently fueled a significant rise in fraudulent activities, resulting in substantial financial losses for individuals and businesses alike. Fraudsters continuously adapt their tactics, posing a significant challenge for researchers developing effective fraud detection systems. The inherent class imbalance in credit card fraud datasets—where legitimate transactions vastly outnumber fraudulent ones— further complicates the development of accurate detection models. Given the crucial role credit cards play in modern economies, impacting households, businesses, and global enterprises, the need for robust fraud detection mechanisms is paramount. This research proposes a Gradient Boosting Classifier, a powerful machine learning technique, as a novel approach to address this critical issue. Our experimental results demonstrate the superior performance of the proposed method compared to other machine learning algorithms, achieving a training accuracy of 100% and a test accuracy of 91%. This high accuracy underscores the effectiveness of the Gradient Boosting Classifier in accurately identifying fraudulent credit card transactions, offering a significant contribution to mitigating the risks associated with this pervasive form of financial crime.

Keywords: *Credit Card Fraud Detection, Gradient Boosting Classifier, Machine Learning ,Imbalanced Dataset, Data Preprocessing, Accuracy, Flask Web Application, Support Vector Machine, Fraud Pulse, Feature Selection*

1. Introduction

The transformative impact of e-commerce and the pervasive adoption of digital payment technologies have elevated the credit card to its current status as the dominant mode of transaction globally. This ubiquitous presence, however, has inadvertently created an environment ripe for exploitation by organized crime syndicates and individual fraudsters alike. The sheer scale of daily credit card transactions, processing billions of interactions across diverse platforms and geographical locations, presents a monumental challenge for security systems. This complexity is further compounded by the constantly evolving tactics employed by fraudsters, who leverage technological advancements to create increasingly sophisticated schemes to circumvent security protocols. These schemes range from simple data breaches and phishing attacks targeting individual consumers, to highly organized operations involving the manipulation of entire payment processing systems. The dynamic nature of this adversarial relationship, where fraudsters continuously refine their techniques in response to improved security measures, necessitates a constant arms race in the development and deployment of advanced fraud detection systems.

Beyond the operational complexities, the inherent characteristics of credit card transaction datasets pose significant methodological challenges. The extreme class imbalance, where legitimate transactions vastly outnumber fraudulent ones by several orders of magnitude, presents a substantial hurdle for many machine learning algorithms. Traditional classification algorithms, optimized for balanced datasets, often struggle to effectively identify the minority class (fraudulent transactions) due to their tendency to prioritize overall accuracy, often at the expense of correctly classifying the rare, yet critically important, fraudulent instances. This leads to high rates of false negatives, where fraudulent transactions are misclassified as legitimate, resulting in significant financial losses and enabling further criminal activity. The consequences extend beyond individual victims,

ISSN: 2583-6129
DOI: 10.55041/ISJEM02166

impacting the financial stability of businesses, the integrity of payment processing networks, and the overall confidence in the security of digital transactions. The societal cost of inaction is substantial, encompassing not only direct financial losses but also indirect costs associated with investigations, legal proceedings, and the erosion of consumer trust.

This research addresses this critical issue by proposing a Gradient Boosting Classifier, a sophisticated machine learning algorithm specifically designed to handle imbalanced datasets and known for its high predictive accuracy. The algorithm's capacity to identify complex non-linear relationships within the data is particularly well-suited to the multifaceted nature of credit card fraud, offering the potential for significant improvements in detection accuracy and a reduction in the financial and societal costs associated with this pervasive criminal activity. This study provides a rigorous evaluation of the Gradient Boosting Classifier's performance in the context of credit card fraud detection, benchmarking its results against other established machine learning algorithms to demonstrate its efficacy in mitigating the risks inherent in the increasingly complex landscape of digital financial transactions.

2 Literature Survey

The landscape of credit card fraud detection research has undergone a significant evolution, reflecting both the increasing sophistication of fraudulent techniques and the rapid advancements in machine learning. Early approaches primarily relied on rule-based systems and expert systems, which encoded pre-defined rules based on known fraudulent patterns (Bolton & Hand, 2002). While these systems provided a straightforward approach, they proved inherently inflexible and struggled to adapt to the constantly evolving tactics of fraudsters. Their inability to generalize to unseen fraud patterns and their dependence on manual rule definition limited their effectiveness in the face of novel and increasingly complex fraudulent activities.

The limitations of rule-based systems spurred the adoption of statistical methods, including logistic regression and discriminant analysis (Weston et al., 2003). These approaches offered a more flexible and data-driven alternative, allowing for the modeling of complex relationships between transaction features and the probability of fraud. However, a persistent challenge emerged: the inherent class imbalance in credit card transaction datasets. The overwhelming preponderance of legitimate transactions compared to fraudulent ones introduced significant bias into the models, leading to high false negative rates and a diminished ability to accurately identify fraudulent instances. This imbalance skewed the model's performance metrics, often resulting in high overall accuracy despite poor detection of the critical minority class (fraudulent transactions).

The rise of machine learning brought a new wave of techniques aimed at addressing the limitations of both rule-based and purely statistical approaches. Support Vector Machines (SVMs) (Huang et al., 2007) gained prominence due to their ability to effectively handle high-dimensional data and model non-linear relationships between features. Decision trees (Phua et al., 2005) offered an advantage in terms of interpretability, allowing for a better understanding of the factors influencing the model's predictions. However, the class imbalance problem persisted, prompting the development and widespread adoption of ensemble methods, such as Random Forests (Breiman, 2001), bagging, and boosting. These ensemble techniques combine the predictions of multiple models, effectively reducing the impact of individual model biases and improving overall robustness and predictive accuracy.

The recent surge in computational power and the availability of large datasets have fueled the adoption of deep learning techniques in credit card fraud detection. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks (Baradwaj et al., 2020), have been extensively investigated due to their capacity to model temporal dependencies within sequences of transactions. This ability to capture the temporal context of transactions is crucial for identifying patterns indicative of fraud that unfold over time. Convolutional Neural Networks (CNNs) (Shrestha & Mahanti, 2021) have also been employed to exploit spatial patterns within the features of individual transactions. Deep learning models, however, are computationally intensive and require significant amounts of data for effective training. Furthermore, the risk of overfitting, particularly in the context of imbalanced datasets, necessitates careful model selection, regularization techniques, and hyperparameter tuning.

The persistent challenge of class imbalance has driven research into advanced data preprocessing and sampling methods. Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2002) and related techniques aim to address the imbalance by generating synthetic samples of the minority class, thereby creating a more balanced training dataset. Cost-sensitive learning approaches assign different misclassification costs to different classes, penalizing false negatives (missed fraudulent transactions) more heavily than false positives (incorrectly flagged legitimate transactions). Anomaly detection techniques, focusing on identifying deviations from established patterns of normal behavior, offer another valuable approach to identifying fraudulent activities (Chandola et al., 2009). These methods can be particularly effective in detecting novel fraud patterns that might not be captured by models trained on historical data.

The evaluation of fraud detection models necessitates the use of comprehensive performance metrics that go beyond simple accuracy. Precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (AUC) are commonly used to provide a more

ISSN: 2583-6129
DOI: 10.55041/ISJEM02166

nuanced assessment of model performance, particularly in the context of imbalanced datasets where simple accuracy can be misleading. Furthermore, there's a growing emphasis on model interpretability and explainability (Guidotti et al., 2018), which is crucial for building trust, understanding model limitations, and ensuring responsible deployment of these systems in sensitive financial applications. The ongoing research in credit card fraud detection continues to explore and refine these techniques, adapting to the ever-evolving landscape of fraudulent activities and pushing the boundaries of machine learning capabilities.

2.1 Existing System

Raghavan et.al defined an auto-encoder as an actual neural network. An auto-encoder can also encrypt the data the same way as it would decrypt the data. In this method, for no anomalous points, the auto-encoders are trained. According to the reconstruction error, it would present the anomaly ideas classify it as 'fraud' or `no fraud,' meaning that the system has not been trained, which is predicted to have a higher amount of anomalies. However, a slight value overhead the higher bound value or considers the threshold an anomaly. Carcillo et al. implemented a hybrid approach that utilizes unsupervised outlier scores to expand the set of features of the fraud detection classifier. Their main contribution was to implement and assess various levels of granularity for outlier score definition. Their experimental results indicate that their proposed approach is efficient and enhances detection accuracy.

3. Methodology

In this research study, we aim to detect credit card fraud using Support Vector Machine (SVM), a popular machine learning algorithm. The methodology used in this study is as follows:

a) **Data collection and preprocessing*:*** We collected the European card benchmark dataset, which contains a large number of transactions labeled as fraudulent or non-fraudulent. We preprocessed the data by removing duplicates, missing values, and outliers

b) **Feature selection:** We selected the most relevant features from the dataset using various feature selection techniques, such as correlation analysis and chi-square tests.

c) **Model training**: We trained the SVM model on the preprocessed data using a training set. We used a radial basis function kernel to map the data into a high-dimensional space and maximize the margin between the two classes.

d) **Model evaluation:** We evaluated the performance of the SVM model using a test set. We measured the performance in terms of accuracy, precision, recall, and F1 score.

e) **Model optimization*:*** We performed hyperparameter tuning to optimize the SVM model's performance. We used cross-validation techniques to find the optimal values for the parameters.

f) **Comparison with other algorithms:** We compared the performance of the SVM model with other popular machine learning algorithms, such as logistic regression, decision trees, and k-nearest neighbors.

g) **Balancing the dataset:** We also experimented with balancing the dataset to reduce the false negative rate and improve the overall performance of the SVM model.

The methodology used in this study provides a comprehensive approach to credit card fraud detection using SVM. By selecting relevant features, optimizing the model, and comparing it with other algorithms, we can evaluate the effectiveness of SVM in detecting fraudulent transactions.

MACHINE LEARNING PROCESS

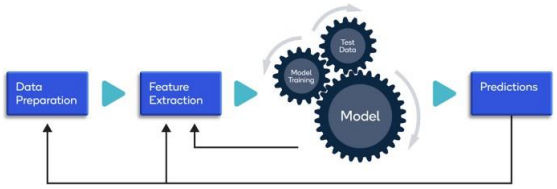

Fig 1:Machine learning process

Machine learning is a branch of artificial intelligence that focuses on developing algorithms and statistical models that enable systems to automatically improve their performance on a specific task, using data as the primary input.

Machine learning algorithms can learn from experience and adjust their parameters to improve their performance without being explicitly programmed.
This makes them useful for a wide range of applications, including image and speech recognition, natural language processing, and predictive analytics. The primary goal of this thesis is to apply machine learning techniques to solve a specific problem or develop a new approach for a particular application. This may involve data collection, preprocessing, feature extraction, model selection, training, and evaluation, as well as hyperparameter tuning and optimization. The effectiveness of the machine learning approach will be evaluated using various metrics, such as accuracy, precision, recall, and F1 score, and compared to existing methods or benchmarks. The outcome of this thesis will contribute to advancing the field of machine learning and potentially provide practical solutions to real- world problems.

International Scientific Journal of Engineering and Management
Volume: 03 Issue: 12 | Dec – 2024
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM02166

Proposed architecture

This paper proposes an intelligent approach for detecting fraudulent credit card transactions that uses support vector machine Classifier. In the proposed approach, the system is intelligently integrated to tune the parameters of the Gradient Boosting Classifier. The proposed approach is primarily concerned with discriminating between legitimate and fraudulent credit card transactions. The main contribution of our research is an intelligent approach for detecting fraud in credit card transactions using Gradient Boosting Classifier. The performance of the proposed intelligent approach is evaluated based on real-world data sets which is referred from kaggle and performance evaluation metrics are calculated. The proposed Gradient Boosting Classifier achieved training accuracy of 100% and test accuracy of 91%. The proposed intelligent approach for credit card fraud detection consists of following major steps, which are data collection, data pre-processing, applying the model, prediction result, performance analysis and graphical representation. The experiment was performed using an Intel Core i3 processor with 8GB RAM. The proposed approach and machine learning techniques were implemented and tested using Python and the web interface was developed using Flask.

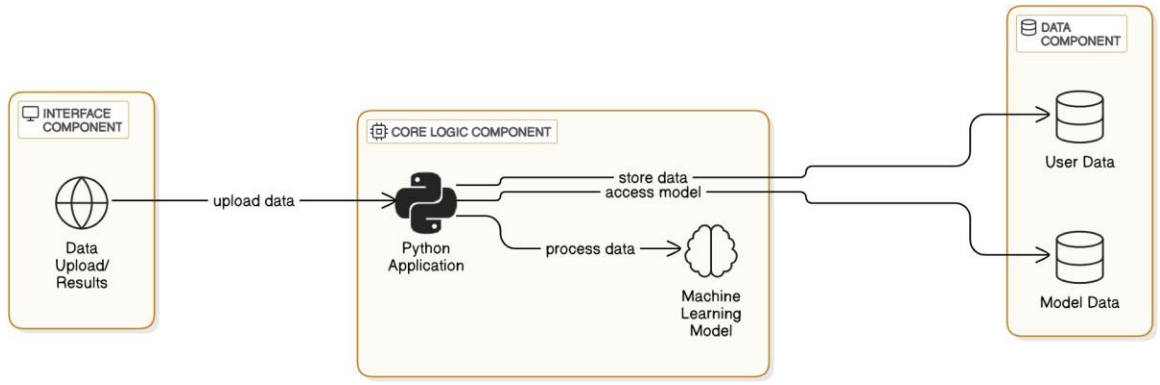

Fig 2 Proposed Architecture

3.1 Data Collection

Data Source:

Data for this fraud pulse project was created for this research, with simulated values meant to resemble real- world scenarios.

Dataset Variables:

The dataset encompasses a range of features related to credit card applications, which can be broadly classified into the following categories:

Applicant Financial Information: These variables describe the applicant's financial status and history:

Checking Account Status: (over_draft) Represents the status of the applicant's checking account, with categories such as <0, 0<=X<200, no checking, and >=200. These categories provide insights into the applicant's immediate access to funds.

Credit History: (credit_history) Provides information on the applicant's past credit performance, using categories like critical/other existing credit, existing paid, delayed previously, and no credits/all paid. This feature is vital for assessing the applicant's creditworthiness.

Savings/Asset Balance: (Average_Credit_Balance) Indicates the average balance held in savings or other asset accounts. This is presented in categories, such as no known savings, <100, 100<=X<500, 500<=X<1000, and >=1000

Current Outstanding Balance: (current_balance) Reflects the current balance the applicant owes on any existing loans.

Transaction Details: This section focuses on the specifics of the current credit application:

Credit Amount Requested: (credit_usage) Specifies the amount of credit the applicant is applying for.

Loan Purpose: (purpose) Indicates the intended use of the credit, categorized into various options (e.g., radio/tv, education, new car, business, furniture/equipment, used car, repairs, etc.). The specific categories were simulated to represent typical loan purposes.

Applicant Demographics & Personal Situation: These variables describe the applicant's personal circumstances:

Employment Status: (employment) Describes the applicant's length of current employment in string format which represents an underlying relationship.

Location: (location) - Location of applicant.

Personal Status: (personal_status) Represents the applicant's marital and gender status using categories like male single, female div/dep/mar, male div/sep, and male mar/wid.

Residence Length: (residence_since) Specifies how long the applicant has been living at their current residence. The timeframe is measured in years.

International Scientific Journal of Engineering and Management
Volume: 03 Issue: 12 | Dec – 2024
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM02166

Property Ownership: (property_magnitude) indicates type of property owned by the applicant

Applicant Age: (cc_age) Represents the age of the applicant at the time of credit application.

Housing Status: (housing) The type of housing the applicant has (rent, own, etc).

Job Category: (job) Describes the applicant's job, using categories like skilled, unskilled resident, high qualif/self emp/mgmt, unemp/unskilled non res.

Number of Dependents: (num_dependents) Specifies how many dependents the applicant has.

Telephone Ownership: (own_telephone) Indicates if the applicant has a landline phone (yes/no). **Foreign Worker Status:** (foreign_worker) A binary variable specifying whether the applicant is a foreign worker (yes/no).

3.2 Model Building

The predictive model was developed through a systematic process involving data preprocessing, model selection, training, and evaluation. Initially, the dataset underwent thorough cleaning, addressing missing values and encoding categorical features using a combination of one-hot, ordinal, and label encoding techniques, followed by standardization of numerical features. The data was then split into training, validation, and test sets, with considerations taken to balance the imbalanced class distribution using oversampling or undersampling strategies, if needed. A machine learning classifier, either a Decision Tree or a Support Vector Machine, was instantiated and trained on the preprocessed training data. Model performance was evaluated using metrics such as accuracy, precision, recall, and F1-score on the validation set, and the best performing model was then fine-tuned using hyperparameter optimization. Finally, the optimized model was serialized and saved using the pickle library, for seamless integration into a web-based fraud detection application.

Data Preprocessing and Preparation: The initial phase of model development involved meticulous data preprocessing to ensure optimal model performance. Raw transaction data was subjected to a series of cleaning and transformation steps. Missing values within relevant features were addressed via imputation, and outliers were handled based on their distribution. Categorical features underwent encoding using a combination of one-hot encoding for nominal data, ordinal encoding for features with inherent order, and label encoding for binary features. Numerical features were standardized to ensure all were in a similar scale. To mitigate potential bias arising from imbalanced classes, oversampling or undersampling techniques were applied during training, while stratified sampling was performed during data splitting.

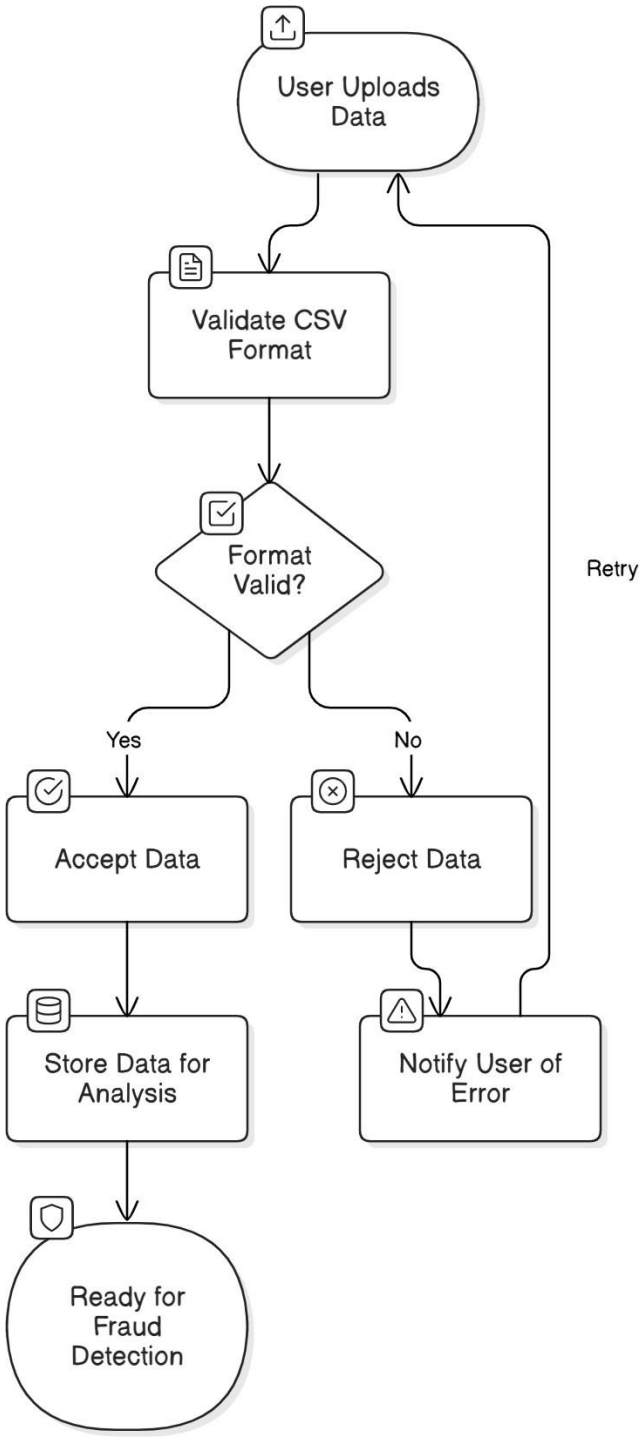

Fig 3 Support vector Machine (SVM)

Model Selection and Training: Given the binary nature of the fraud detection task, either a Decision Tree Classifier or a Support Vector Machine model was selected for model training and experimentation. Each model was initially instantiated with baseline parameters and subsequently trained using the processed training data. This step involved iteratively tuning model parameters and evaluating the performance using relevant metrics. Cross validation

ISSN: 2583-6129
DOI: 10.55041/ISJEM02166

was employed to ensure that the model is able to generalize well to unseen data and was not overfitting to the training data.

Model Evaluation and Tuning: The performance of the trained models was rigorously evaluated using appropriate metrics, such as accuracy, precision, recall, F1-score, AUC-ROC and confusion matrix. These metrics provided insight into the model's ability to correctly classify both fraudulent and legitimate transactions, allowing for a balanced view of the model's capability. Based on the evaluation, the models were fine-tuned using hyperparameter optimization techniques like grid search or randomized search to find the most optimal set of parameters that can achieve the highest accuracy on the held-out validation dataset.

Model Serialization and Deployment: Upon achieving a satisfactory level of performance on the validation set, the best-performing model was selected. This final model was then serialized and stored as a pickle file, which facilitates seamless integration into the web- based fraud detection application. The serialized model can now be loaded and used for making real-time predictions on new, unseen transaction data. The model is evaluated using a separate test set and the reported results represent the final model's performance.

3.3 Model Evaluation

The evaluation of the trained fraud detection model is a critical step to ensure its effectiveness and reliability. This process aims to quantify the model's ability to accurately classify credit card transactions as either legitimate ("good") or fraudulent ("bad"), and it involves several key components:

1. Evaluation Data:

Test Dataset: A held-out dataset (typically 15-20% of the original data) is used for evaluation. This dataset was *not* used during model training or hyperparameter tuning and acts as an unbiased representation of real- world data that the model has never seen before. This approach ensures the model's generalization ability and its performance on unseen data.

Independent Evaluation: It's crucial to ensure that the test dataset is entirely independent of the training and validation sets to prevent data leakage and provide an accurate assessment of the model's ability to generalize.

2. Performance Metrics:

The following metrics are calculated to quantify the model's performance on the test set:

Accuracy: The overall proportion of correctly classified transactions (both "good" and "bad"). While useful for a general overview, accuracy may be misleading for imbalanced datasets.

- Accuracy = (True Positives + True Negatives) / (Total Transactions)

Precision: Measures the proportion of transactions predicted as fraudulent that are actually fraudulent. This metric is crucial for minimizing false positives (i.e., incorrectly flagging legitimate transactions as fraud).

- Precision = True Positives / (True Positives + False Positives)

Recall (Sensitivity): Measures the proportion of actual fraudulent transactions that are correctly identified. Recall is vital for minimizing false negatives (i.e., failing to identify actual fraudulent transactions).

- Recall = True Positives / (True Positives + False Negatives)

F1-Score: The harmonic mean of precision and recall. F1-score provides a single metric that balances the trade-off between precision and recall, particularly useful in imbalanced datasets.

- F1-Score = 2 * (Precision * Recall) / (Precision + Recall)

Area Under the Receiver Operating Characteristic Curve (AUC-ROC): This is used when evaluating a binary classification model, and it measures the overall ability of the model to distinguish between the classes of good and bad. The ROC curve plots the true positive rate against the false positive rate at different classification thresholds, and the AUC represents the area under that curve.

Confusion Matrix: This metric provides a breakdown of the model predictions into four categories (True Positives, True Negatives, False Positives, and False Negatives). It gives a granular look at the model's performance, which helps in identifying areas where the model excels and where it could be improved.

3. Interpretation of Metrics:

The performance metrics provide a comprehensive view of the model's capabilities and limitations. Accuracy, while offering a general measure of correctness, should be interpreted cautiously given the imbalanced nature of fraud datasets, where a high accuracy might mask poor performance on the minority class (fraudulent transactions). Therefore, precision and recall were critical indicators: precision reflects the model's ability to avoid false alarms, i.e., minimizing the number of legitimate transactions wrongly flagged as fraud, whereas recall quantifies the model's sensitivity to detect all actual fraudulent activities. The F1-score, by balancing both precision and recall, serves as a holistic measure of the model's predictive capability, especially when dealing with imbalanced classes. The AUC-ROC score provides an assessment of the model's overall ability to distinguish between fraudulent and legitimate transactions, and the confusion matrix gives a detailed perspective on where the model tends to make classification errors, all of which helps in guiding iterative model development and deployment decisions.

International Scientific Journal of Engineering and Management
Volume: 03 Issue: 12 | Dec – 2024
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM02166

4 Flask Web Application Implementation Step

1: Model Loading:

A crucial initial step is loading the pre-trained fraud detection model (either a Decision Tree or SVM). This step assumes that the model has been previously trained and saved to a file (e.g., credit.pkl) using Python's pickle module. The application's load_model() function handles the model loading process. This function must be robust, handling situations where the model file may be missing or corrupted. It returns the loaded model if successful or None if loading fails, which can allow for more graceful handling of loading errors during run-time. Any potential exceptions during loading must be handled to avoid application failure and improve reliability.

Step 2: Route Configuration (using Flask):

Flask routes are defined to manage different functionalities within the application. The route , index or prediction is used to display the user input forms, and /predict is configured to process user input and display prediction results. Each route has explicit instruction as to what parameters and data formats are required. The data structures and variables used in the web forms reflect appropriate and user-friendly choices of labels based on the data and features described in the dataset.

Step 3: Input Handling and Data Validation:

The Flask route designated for processing input data (e.g., /predict) must document and implement explicit steps that must handle data submitted via web forms, outlining what data types and formats are expected. The validation of the user provided input is a critical part of handling input data. Each input field undergoes validation to prevent erraors arising from malformed or incorrect input. For example, numeric fields are verified to contain only numerical values and to be within a reasonable range for credit card information. Steps for sanitization to prevent injection attacks should be explicitly documented within code. All user input is treated as untrusted. The reasons for specific validations, and any limitations due to differences in data types used for model training must be documented as a justification for validation approach.

Step 4: Data Preparation (with Model):

After user input has been validated, the user submitted input must be correctly structured so that it can be understood by the machine learning model, which must be a 2D NumPy array to be used for prediction using scikit-learn models. This involves converting user-provided input into the correct data types and formats. String values might need to be converted to categorical codes (using one-hot or ordinal encoders) or other forms of data preprocessing. The data preparation steps done here are designed to mimic the preprocessing used during the model training to ensure compatibility. The features must be selected based on the previous steps in model training.

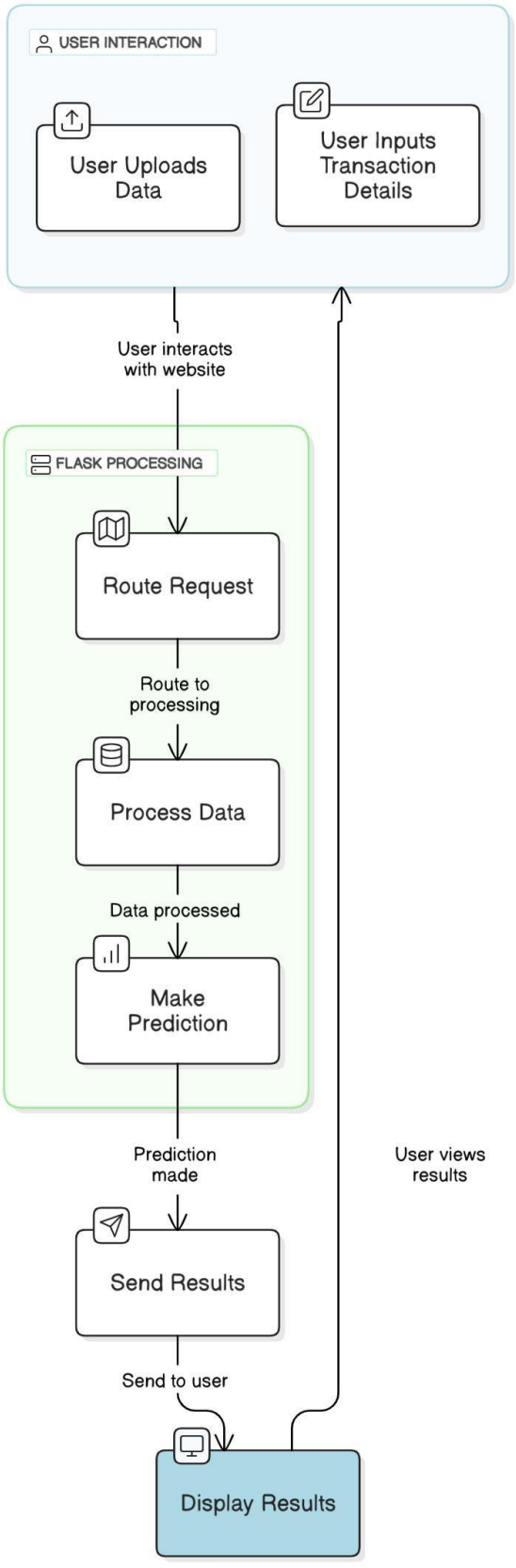

Fig 4 Flask Web Application

ISSN: 2583-6129
DOI: 10.55041/ISJEM02166

Step 5: Prediction and Result Formulation:

The load_model() function is called to retrieve the pre-trained model. The data is pre-processed using the specified steps and fed into the trained model for prediction. The prediction output of the machine learning model is a binary output, that is formatted to display a user friendly output. This format could be the string representations Fraud or Good based on the models prediction output. This information is then passed to the HTML template for a user-friendly result presentation. The results are provided to the user, which is shown within the web page

5 Conclusion

The Fraud Pulse system leverages a supervised machine learning algorithm, specifically either a Decision Tree or a Support Vector Machine (SVM), to perform credit card transaction classification. These algorithms are trained on a labeled dataset consisting of various features extracted from credit card applications and their corresponding outcomes ("good" or "bad"). The model learns to identify complex relationships and patterns between the input features and the likelihood of a transaction being fraudulent. During the training phase, the algorithm adjusts its internal parameters to minimize the classification error. Once trained, the model can then be used to classify new, unseen transactions as either "good" (legitimate) or "bad" (fraudulent), based on the patterns and rules it has acquired during training. The system's effectiveness hinges on the algorithm's ability to discern subtle differences between genuine and fraudulent activities. The trained model is then incorporated in the web application to analyze user's data.

Refrences

[1] Y. Abakarim, M. Lahby, and A. Attioui, ``An effficient real time model for credit card fraud detection based on deep learning," in Proc. 12th Int. Conf. Intell. Systems: Theories Appl., Oct. 2018, pp. 17, doi: 10.1145/3289402.3289530.

[2] H. Abdi and L. J. Williams, ``Principal component analysis," Wiley Interdiscipl. Rev., Comput. Statist., vol. 2, no. 4, pp. 433459, Jul. 2010, doi: 10.1002/wics.101.

[3] V. Arora, R. S. Leekha, K. Lee, and A. Kataria, ``Facilitating user authorization from imbalanced data logs of credit cards using artificial intelligence," Mobile Inf. Syst., vol. 2020, pp. 113, Oct. 2020, doi: 10.1155/2020/8885269.

[4] A. O. Balogun, S. Basri, S. J. Abdulkadir, and A. S. Hashim, ``Performance analysis of feature selection methods in software defect prediction: A search method approach," Appl. Sci., vol. 9, no. 13, p. 2764, Jul. 2019, doi: 10.3390/app9132764.

[5] B. Bandaranayake, ``Fraud and corruption control at education system level: A case study of the Victorian department of education and early childhood development in Australia," J. Cases Educ. Leadership, vol. 17, no. 4, pp. 3453, Dec. 2014, doi: 10.1177/1555458914549669.

[6] J. Bªaszczy«ski, A. T. de Almeida Filho, A. Matuszyk, M. Szelg , and R. Sªowi«ski, ``Auto loan fraud detection using dominance-based rough set approach versus machine learning methods," Expert Syst. Appl., vol. 163, Jan. 2021, Art. no. 113740, doi: 10.1016/j.eswa.2020.113740.

[7] B. Branco, P. Abreu, A. S. Gomes, M. S. C. Almeida, J. T. Ascensa˜o, and P. Bizarro, ``Interleaved sequence RNNs for fraud detection," in Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2020, pp. 31013109, doi: 10.1145/3394486.3403361.

[8] F. Cartella, O. Anunciacao, Y. Funabiki, D. Yamaguchi, T. Akishita, and O. Elshocht, ``Adversarial attacks for tabular data: Application to fraud detection and imbalanced data," 2021, arXiv:2101.08030.

[9] S. S. Lad, I. Dept. of CSE Rajarambapu Institute of Technology Rajaramnagar Sangli Maharashtra, and A. C. Adamuthe, ``Malware classification with improved convolutional neural network model," Int. J. Comput. Netw. Inf. Secur., vol. 12, no. 6, pp. 3043, Dec. 2021, doi: 10.5815/ijcnis.2020.06.03.

[10] V. N. Dornadula and S. Geetha, ``Credit card fraud detection using machine learning algorithms," Proc. Comput. Sci., vol. 165, pp. 631641, Jan. 2019, doi: 10.1016/j.procs.2020.01.057.

[11] I. Benchaji, S. Douzi, and B. E. Ouahidi, ``Credit card fraud detection model based on LSTM recurrent neural networks," J. Adv. Inf. Technol., vol. 12, no. 2, pp. 113118, 2021, doi: 10.12720/jait.12.2.113-118.

[12] Y. Fang, Y. Zhang, and C. Huang, ``Credit card fraud detection based on machine learning," Comput., Mater. Continua, vol. 61, no. 1, pp. 185195, 2019, doi: 10.32604/cmc.2019.06144.

[13] J. Forough and S. Momtazi, ``Ensemble of deep sequential models for credit card fraud detection," Appl. Soft Comput., vol. 99, Feb. 2021, Art. no. 106883, doi: 10.1016/j.asoc.2020.106883.

[14] K. He, X. Zhang, S. Ren, and J. Sun, ``Deep residual learning for image recognition," 2015, arXiv:1512.03385.

[15] X. Hu, H. Chen, and R. Zhang, ``Short paper: Credit card fraud detection using LightGBM with asymmetric error control," in Proc. 2nd Int. Conf. Artif. Intell. for Industries (AII), Sep. 2019, pp. 9194, doi:10.1109/AI4I46381.2019.00030.