ONLINE PAYMENT FRAUD DETECTION USING MACHINE LEARNING

Dr. M. Venkatesh[1]
Bhukya Keerthi Bai [2], Budati Bhargavi [3], Chilukoti Manasa [4], Dwarasala Mokshitha [5]

[1]*Associative Professor, Department of ECE, Vasireddy Venkatadri Institute of Technology, Namburu, Guntur, Andhra Pradesh, India*

[2-5]*Undergraduate Students, Department of ECE, Vasireddy Venkatadri Institute of Technology, Namburu, Guntur, Andhra Pradesh, India*

ABSTRACT

As we are approaching modernity, the trend of paying online is increasing tremendously. It is very beneficial for the buyer to pay online as it saves time, and solves the problem of free money. At the same time, the related network transaction fraud problem has become more significant. Our project aim is to enhance online payment security through the application of machine learning models for fraud detection. Machine learning models can analyze large volumes of transactional data more accurately and faster than manual inspection.

Keyword: - *Fraud Detection, Online Transactions, Random Forest, Machine Learning*

1. INTRODUCTION

In the ever-evolving landscape of the digital age, the rise of online payment fraud has emerged as a paramount concern, casting a shadow over businesses, financial institutions, and consumers. The spectrum of fraudulent activities encompasses identity theft, pilfered credit card information, and unauthorized transactions, all of which pose a substantial threat to the integrity of online financial transactions.

In response to this escalating challenge, the integration of machine learning algorithms has proven instrumental in fortifying online payment systems against illicit activities. These algorithms empower systems to meticulously scrutinize vast troves of historical transaction data, discern intricate patterns, and adapt by learning from these patterns. By doing so, they not only enhance the efficiency of fraud detection but also enable the timely prediction and flagging of potentially nefarious activities. In essence, the synergy between machine learning and online payment security stands as a crucial bulwark, fortifying digital transactions and safeguarding the interests of businesses, financial institutions, and consumers in this dynamic and interconnected era.

2. EXISTING SYSTEM

The existing online payment fraud detection model relies on the decision tree algorithm, which is a widely used supervised learning technique. The algorithm constructs a flowchart-like structure driven by input features, starting with the entire dataset as the root node and iteratively splitting based on optimal features until a stopping criterion is met. Leaf nodes are assigned labels based on the majority class, facilitating predictions. The model's implementation involves loading libraries, reading and preprocessing the dataset, splitting it for training and testing, handling missing values, building and training the decision tree, and making predictions. The flowchart outlines the key steps, emphasizing dataset loading, feature definition, data splitting, imputation, decision tree construction, and prediction. The model seeks to balance interpretability and adaptability, addressing fraud detection challenges while

considering decision tree limitations and implementing measures to enhance accuracy and prevent overfitting and underfitting.

2.1 Disadvantages in existing system

The decision tree classifier in your project has certain drawbacks. It is prone to overfitting, especially when the tree is deep, which captures noise and leads to poor generalization. Decision trees are sensitive to small variations in the training data, resulting in different tree structures with slight changes. They can be biased toward features with more levels, lack global optimality, and struggle with imbalanced datasets. While decision trees offer simplicity and interpretability for shallow trees, deep trees are complex and challenging to interpret. In addition However they may not capture complex relationships as effectively as more advanced models such as neural networks. Consideration of these limitations is essential, and alternative algorithms such as random forests, gradient boosting, or neural networks might be explored based on the specific characteristics of the data and the project's objectives.

3. PROPOSED SYSTEM

The provided code implements an interactive fraud detection application using a pre-trained random forest classifier. The model is loaded from the .sav file, and an IPython widget-based user interface is created, which allows users to input transaction details. These details include account age, number of items, local time, payment method, and payment method age. Upon clicking the "Check Fraud" button, the code processes the user-entered data through the Random Forest model and displays the prediction in the output area, indicating whether the transaction is classified as "Fraud" or "Not Fraud." This application provides a convenient and intuitive means for users to leverage machine learning for fraud detection in real-time scenarios.

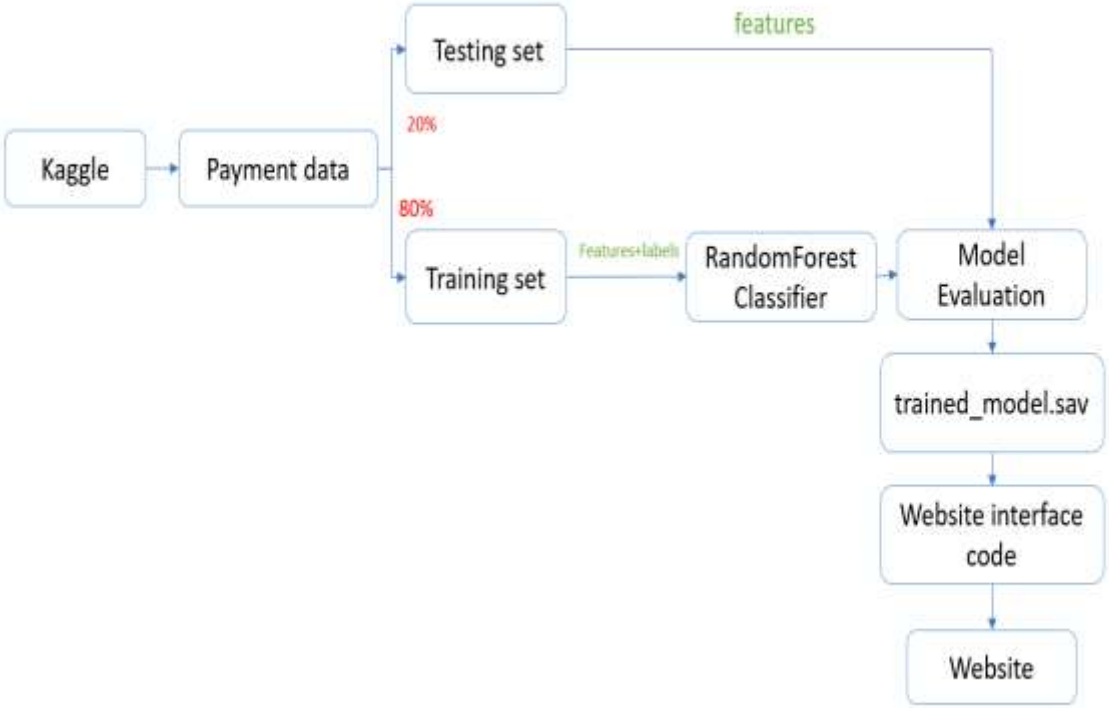

Fig-1: Block Diagram of online payment fraud detection system

3.1 Prerequisites & Environment

- **Jupyter Notebook:** Jupyter Notebook is an open-source web application that allows users to create and share documents containing live code, equations, visualizations, and narrative text. It supports various programming languages, including Python, R, and Julia, making it versatile for data analysis, machine learning, scientific computing, and more. Jupyter Notebook offers a flexible and interactive environment for data analysis, machine learning, and scientific computing, with features that promote collaboration, reproducibility, and productivity.
- **Anaconda:** Node.js is a runtime environment that allows developers to run JavaScript code outside of a web browser, making it particularly suitable for server-side applications. In the context of building a network using Hyperledger Fabric, Node.js is essential for developing smart contracts and building client applications to interact with the Fabric network.
- **Spyder:** Spyder, an integral facet of the Anaconda distribution, seamlessly integrates the Jupyter notebook environment with Spyder's IDE capabilities. Through its user-friendly interface, practitioners engage in interactive Python coding, data visualization, and documentation using markdown cells. This versatile tool empowers users with a flexible environment for scientific computing and data exploration, fostering efficiency and precision within the Anaconda ecosystem.

3.2 Libraries

- **sickit-learn:** Scikit-learn, often abbreviated as sklearn, is a popular open-source machine learning library for Python. It provides several tools for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, and preprocessing. The library provides tools for model evaluation, including functions for cross-validation, hyperparameter tuning, and various performance metrics ,such as accuracy, precision, recall, F1-score, and ROC curves.
- **Flask:** Flask is a lightweight and flexible Python web framework that provides tools and libraries for building web applications. It is particularly useful for developing web applications that require handling HTTP requests, routing, rendering templates, and interacting with databases**.** With its minimalist design and modular architecture, Flask provides developers with the freedom to customize and extend their applications according to their specific needs.

3.3 Methodology

While the number of online auctions continues to grow, the number of online auction scams is also increasing. To avoid detection, scammers often disguise their normal trading behavior by disguising themselves as honest participants. Therefore staying vigilant is not enough to prevent scams. Online auction participants require a more proactive approach to protecting their interests, such as an early fraud detection system.

The steps to implement are as follows:

- Install required libraries and dependencies for data preprocessing and model evaluation in jupyter.
- Install the online payment transaction dataset from Kaggle.
- Clean the dataset by handling missing values, outliers, and inconsistencies and converting payment types from categorical labels to numerical labels.
- Split the dataset into training and testing sets to evaluate model performance. Using a random forest classifier and train the model.
- Evaluate the trained model’s performance on the testing dataset using metrics such as accuracy, precision, recall, and F1 score. Analyze the confusion matrix to understand the model's ability to detect fraudulent and non-fraudulent transactions.
- Save the trained model to a file with .sav extension for deployment. Here, the Flask library is used for deployment, as Flask is a micro web framework for Python, specifically designed for building web applications.
- Using the spyder application, we developed Python code using a flask and created a web app.

4. RESULT

```
Accuracy
0.9997108109552354

classification report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00   1588610
           1       0.98      0.79      0.88      2045

    accuracy                           1.00   1590655
   macro avg       0.99      0.90      0.94   1590655
weighted avg       1.00      1.00      1.00   1590655
```

FIG-1: Model Performance

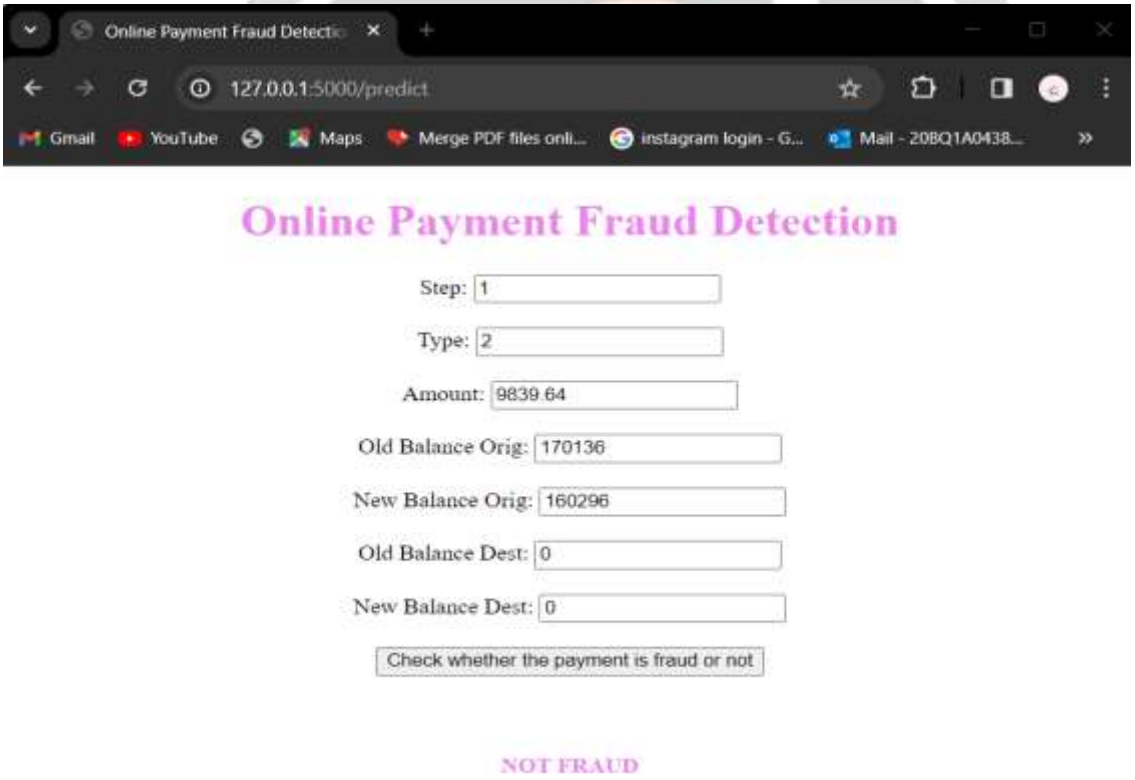

FIG-2: Original Transaction

FIG-3: Fraud transaction

5. CONCLUSIONS

In conclusion, the outlined methodology provides a structured approach for building and deploying a machine learning model for online fraud detection. By following these steps, organizations can effectively leverage data-driven techniques to mitigate the risks associated with fraudulent online transactions.

Through the collection and preprocessing of relevant data coupled with the application of suitable machine learning algorithms, organizations can develop models capable of distinguishing between normal and fraudulent patterns. Rigorous training, testing, and validation processes ensure the accuracy and reliability of the model before deployment into real-world environments.

Once deployed, the model becomes an integral component of the online fraud detection system, continuously analyzing transactions in real time to identify suspicious activities. Its effectiveness hinges on ongoing monitoring and periodic updates to adapt to evolving fraud tactics and maintain optimal performance.

Ultimately, the adoption of machine learning for fraud detection not only enhances security but also contributes to the overall trust and integrity of online transactions, safeguarding both businesses and consumers against financial losses and reputational damage.

6. REFERENCES

[1]. Abdallah, Aisha, Mohd Aizaini Maarof & Anazida Zainal. (2016). Fraud detection system: A survey. Journal of Network and Computer Applications, 68, 90-113.
[2]. Minastireanu, Elena-Adriana & Gabriela Mesnita. (2019). An analysis of the most used machine learning algorithms for online fraud detection. Informatica Economica, 23(1)
[3]. Zhang, Zhaohui, et al. (2018). A model based on convolutional neural network for online transaction fraud detection. Security and Communication Networks.
[4]. Minastireanu, Elena-Adriana & Gabriela Mesnita. (2019). Light gbm machine learning algorithm to online click fraud detection. J. Inform. Assur. Cybersecur, 263928.