RAJALAKSHMI ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

Department of Artificial Intelligence and Data Science

Intelligence fraud detection: Leveraging Deep Learning For Credit Card Transactions

Mrs.Renuka Devi S
M.E(Asst. Prof)

Faleel Mohsin 221801010
Kawshik 221801025

Problem Statement and Motivation

- □ A credit card company is aiming to resolve credit card fraud anomalies by analyzing transaction patterns and customer behavior to minizine a potential credential fraud in the system.

- □ The motivation of this problem statement is to minimize financial losses, gaining customers trust and upholding the brand reputation.

Objectives

The primary goal of this project is to reduce customers financial losses by analyzing the transaction process and customer behaviour to detect fradulent activities within the system.

Abstract

□ This project aims to enhance credit card fraud detection capabilities by analyzing real-time and past transaction patterns and customer behavior. By leveraging advanced data analytics and machine learning techniques, the system will identify and minimize potential fraudulent activities. The project will focus on developing adaptive models that can detect anomalies in real-time, reduce false positives, and improve the overall security of the credit card system. This approach will ensure a more robust and efficient fraud detection system, ultimately safeguarding customers and reducing financial losses.

Introduction and Overview of the Project

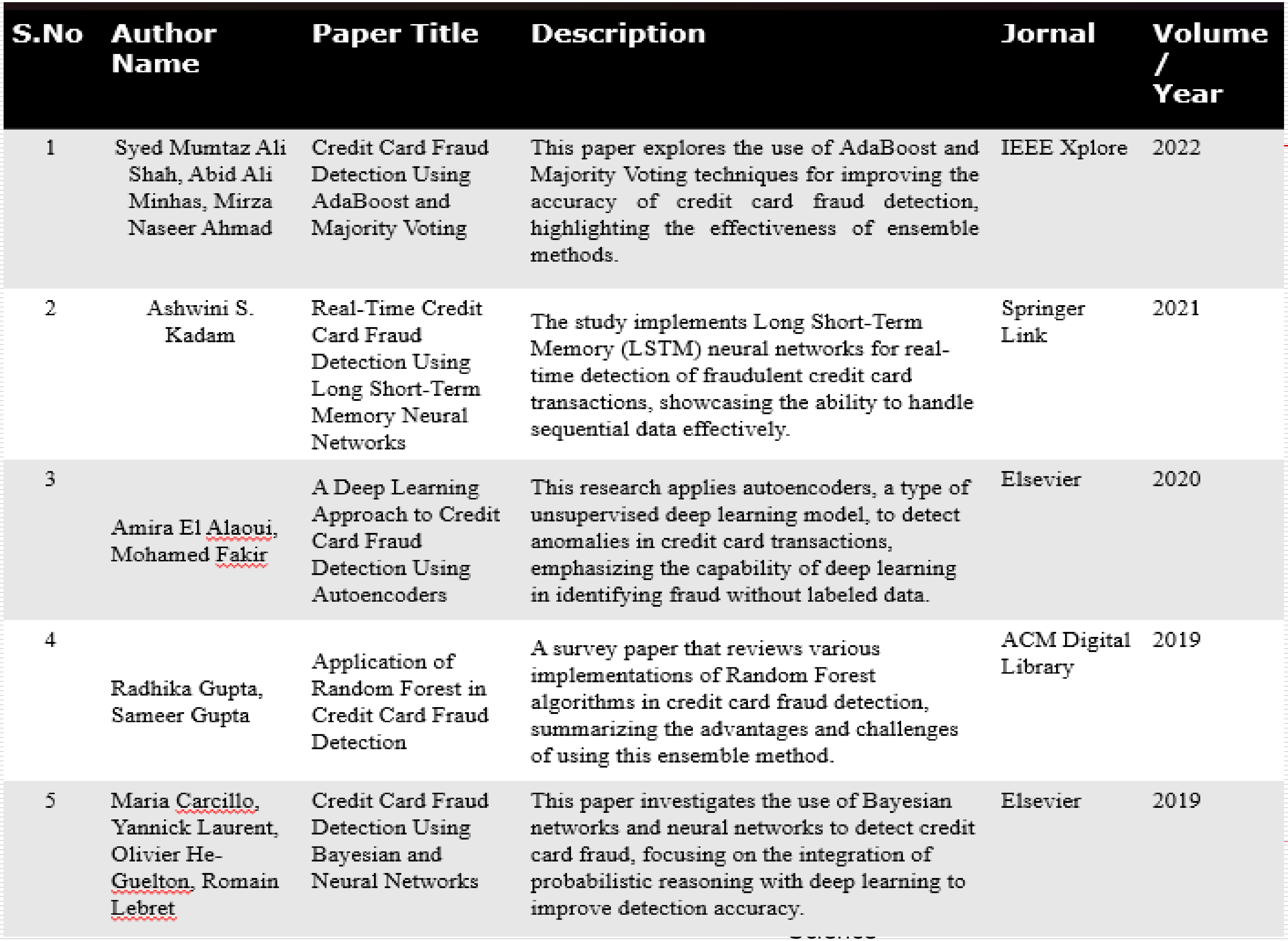
Introduction:

Credit card is a widely used payment method for online and offline transactions. However, the mass evolution of credit cards paved the way for different fraudulent actions.

Overview:

The aim of this project is to create a robust system for real-time identification of fraudulent credit card transactions. Using advanced data analysis and machine learning, the system will analyze transaction data and customer behavior to detect anomalies fraud patterns

Literature Survey



S.No	Author Name	Paper Title	Description	Jornal	Volume / Year
1	Syed Mumtaz Ali Shah, Abid Ali Minhas, Mirza Naseer Ahmad	Credit Card Fraud Detection Using AdaBoost and Majority Voting	This paper explores the use of AdaBoost and Majority Voting techniques for improving the accuracy of credit card fraud detection, highlighting the effectiveness of ensemble methods.	IEEE Xplore	2022
2	Ashwini S. Kadam	Real-Time Credit Card Fraud Detection Using Long Short-Term Memory Neural Networks	The study implements Long Short-Term Memory (LSTM) neural networks for real-time detection of fraudulent credit card transactions, showcasing the ability to handle sequential data effectively.	Springer Link	2021
3	Amira El Alaoui, Mohamed Fakir	A Deep Learning Approach to Credit Card Fraud Detection Using Autoencoders	This research applies autoencoders, a type of unsupervised deep learning model, to detect anomalies in credit card transactions, emphasizing the capability of deep learning in identifying fraud without labeled data.	Elsevier	2020
4	Radhika Gupta, Sameer Gupta	Application of Random Forest in Credit Card Fraud Detection	A survey paper that reviews various implementations of Random Forest algorithms in credit card fraud detection, summarizing the advantages and challenges of using this ensemble method.	ACM Digital Library	2019
5	Maria Carcillo, Yannick Laurent, Olivier He-Guelton, Romain Lebret	Credit Card Fraud Detection Using Bayesian and Neural Networks	This paper investigates the use of Bayesian networks and neural networks to detect credit card fraud, focusing on the integration of probabilistic reasoning with deep learning to improve detection accuracy.	Elsevier	2019

Existing System

□ **Rule-Based Systems**

- Rule-based systems rely on predefined rules and thresholds to detect fraudulent transactions.

□ **Machine Learning Techniques**

- Supervised Learning : Models are trained on historical labeled data (fraudulent and non-fraudulent transactions).
- Unsupervised Learning : Models identify patterns and anomalies without labeled data.

Drawback of Existing System

❑ **Rule-Based Systems**

- Rule-based systems are inflexible and need manual updates to handle new fraud patterns. They become complex and slow as rules increase, require high maintenance, and often miss sophisticated fraud.

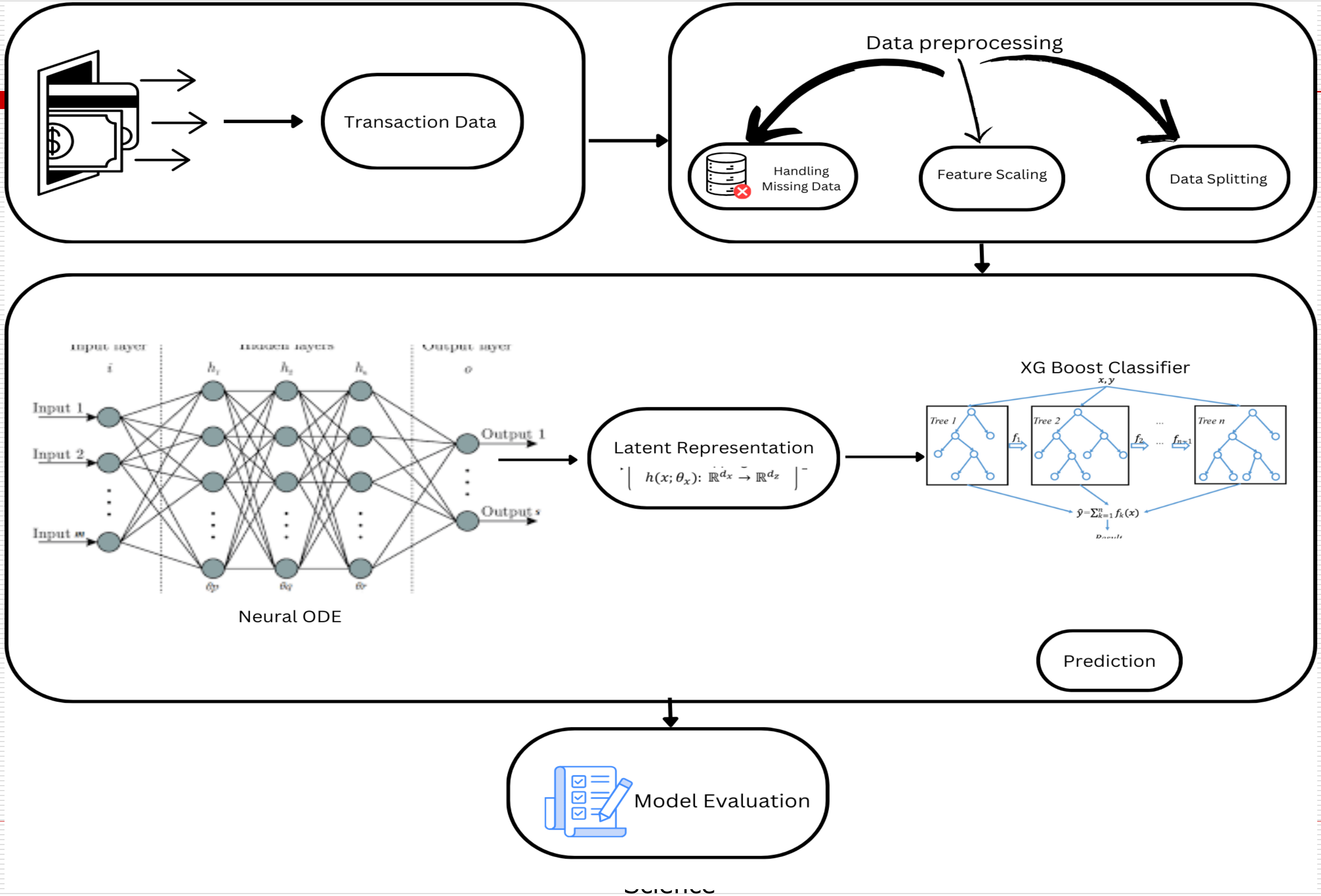
❑ **Machine Learning**

- ML models rely on large amounts of high-quality, labeled data and require significant manual effort for feature engineering. They can suffer from overfitting or underfitting and are often difficult to interpret.

Proposed System

1. **System Integration:** Combines Neural OED and XGBoost for credit card fraud detection.
2. **Feature Engineering:** Neural OED is used to identify significant features from transaction data.
3. **Classifier Training:** XGBoost is trained using the optimized features.
4. **Model Optimization:** Hyperparameters of XGBoost are fine-tuned and validated with cross-validation.
5. **Real-Time Deployment**: The system is deployed for real-time fraud detection.6. Continuous Monitoring: The system is monitored and updated based on new data and feedback.

System Architecture



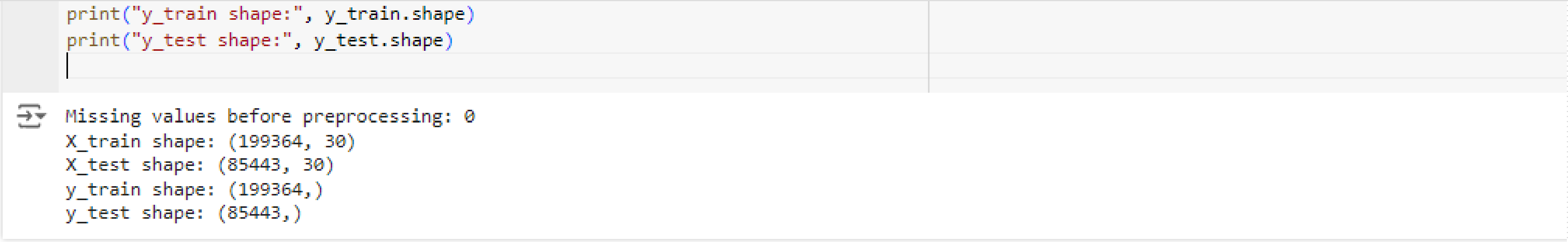
List of modules

- Module 1: Data Pre-processing Module.
- Module 2: Data balancing Module
- Module 2: Implementing neural Module
- Module 3: Implementation of XGBoost
- Module 4: Evaluation Module
- Module 5: Fraud Detection Module.

Data Pre-Processing Module

- Handling Missing Data: Rows with missing values are dropped to ensure a clean dataset for accurate analysis.
- Feature Scaling: Features are standardized using StandardScaler, ensuring consistent scaling across all features.
- Data Splitting: The dataset is split into training and testing sets, maintaining class distribution with stratified sampling.

Output:

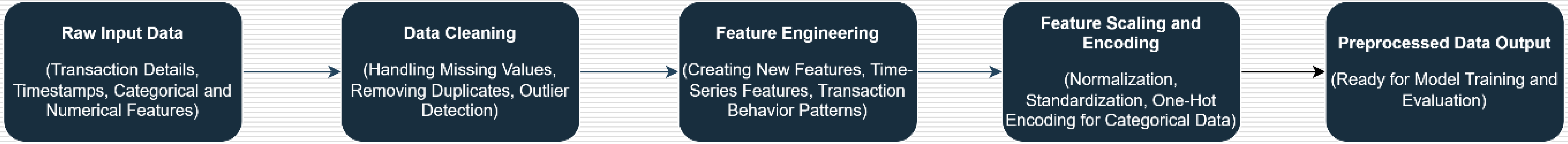


```
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

```
Missing values before preprocessing: 0
X_train shape: (199364, 30)
X_test shape: (85443, 30)
y_train shape: (199364,)
y_test shape: (85443,)
```

Data Pre-Processing Module:

- Data Flow Diagram:

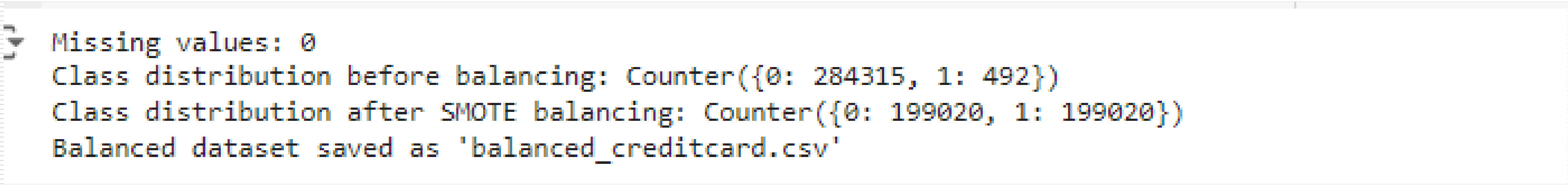


Raw Input Data
(Transaction Details, Timestamps, Categorical and Numerical Features)
Data Cleaning
(Handling Missing Values, Removing Duplicates, Outlier Detection)
Feature Engineering
(Creating New Features, Time-Series Features, Transaction Behavior Patterns)
Feature Scaling and Encoding
(Normalization, Standardization, One-Hot Encoding for Categorical Data)
Preprocessed Data Output
(Ready for Model Training and Evaluation)

Data Balancing Module

- Synthetic Oversampling: SMOTE generates synthetic samples for the minority class (fraud) by interpolating between existing data points, rather than simply duplicating them.
- Maintains Class Distribution: It ensures that both the majority and minority classes have equal representation in the training set, addressing the issue of class imbalance.
- Focused on Training Data: SMOTE is applied only to the training set, preventing information leakage into the testing set while ensuring the model is trained on balanced data.

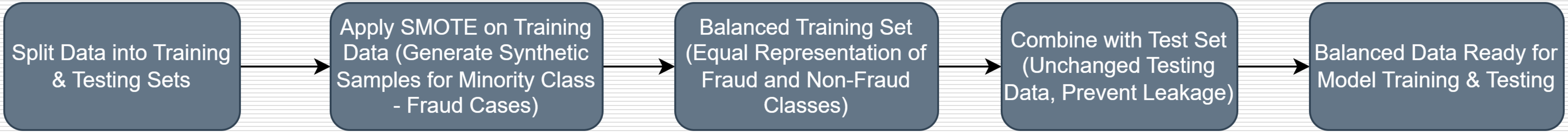
Output:



```
Missing values: 0
Class distribution before balancing: Counter({0: 284315, 1: 492})
Class distribution after SMOTE balancing: Counter({0: 199020, 1: 199020})
Balanced dataset saved as 'balanced_creditcard.csv'
```

Data Balancing Module

- Data Flow Diagram:



Split Data into Training & Testing Sets

Apply SMOTE on Training Data (Generate Synthetic Samples for Minority Class - Fraud Cases)

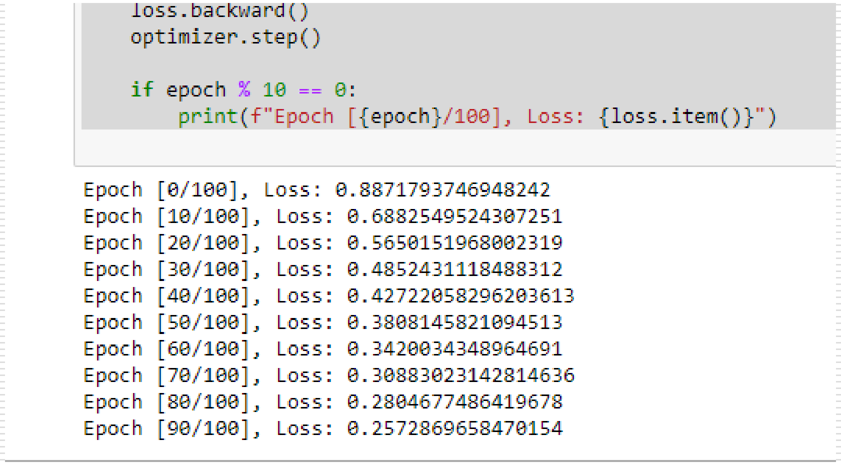
Balanced Training Set (Equal Representation of Fraud and Non-Fraud Classes)

Combine with Test Set (Unchanged Testing Data, Prevent Leakage)

Balanced Data Ready for Model Training & Testing

Implementation of Neural ODE Module:

- **Optimization:** The model uses the Adam optimizer, which adjusts the weights to reduce prediction errors (using binary cross-entropy loss) during training, helping the model improve its fraud detection.
- **Learning Through ODE:** Gradients (how much to adjust weights) are computed through the ODE block, allowing the model to learn how features evolve over time and capture complex relationships in the data.
- **Iterative Training:** The model processes the training data multiple times (epochs), gradually refining its ability to classify fraudulent and non-fraudulent transactions with each pass.
- Output:



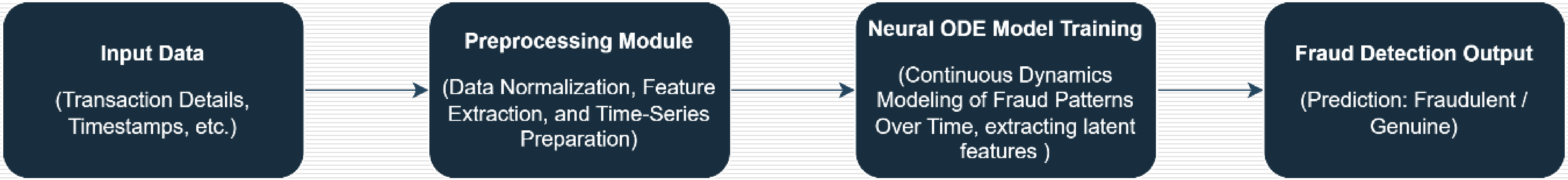
```
    loss.backward()
    optimizer.step()

    if epoch % 10 == 0:
        print(f"Epoch [{epoch}/100], Loss: {loss.item()}")
```

```
Epoch [0/100], Loss: 0.8871793746948242
Epoch [10/100], Loss: 0.6882549524307251
Epoch [20/100], Loss: 0.5650151968002319
Epoch [30/100], Loss: 0.4852431118488312
Epoch [40/100], Loss: 0.42722058296203613
Epoch [50/100], Loss: 0.3808145821094513
Epoch [60/100], Loss: 0.3420034348964691
Epoch [70/100], Loss: 0.30883023142814636
Epoch [80/100], Loss: 0.2804677486419678
Epoch [90/100], Loss: 0.2572869658470154
```

Implementation of Neural ODE Module:

- Data Flow Diagram:



Input Data

(Transaction Details, Timestamps, etc.)

Preprocessing Module

(Data Normalization, Feature Extraction, and Time-Series Preparation)

Neural ODE Model Training

(Continuous Dynamics Modeling of Fraud Patterns Over Time, extracting latent features)

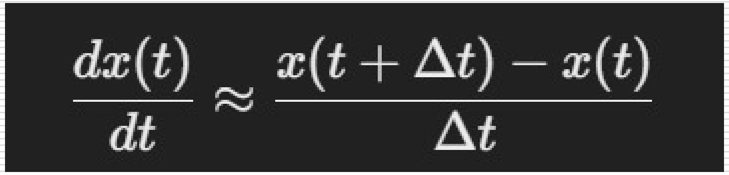
Fraud Detection Output

(Prediction: Fraudulent / Genuine)

Formulas:

□ **Ordinary Differential Equation (ODE):**

The dynamics of the system are modeled using the equation:

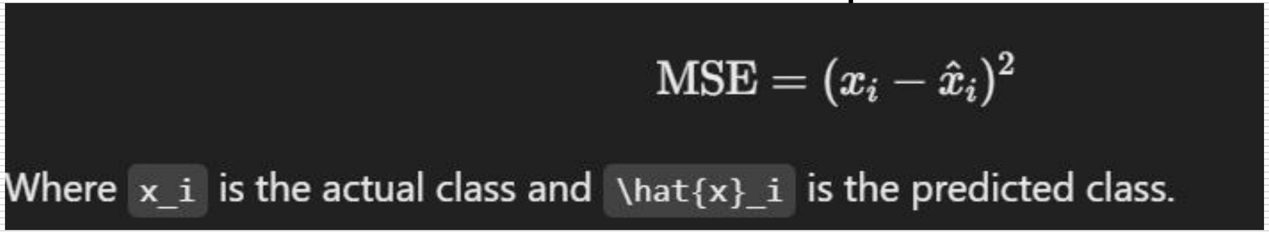


$$\frac{dx(t)}{dt} \approx \frac{x(t+\Delta t) - x(t)}{\Delta t}$$

Here, x(t) is the state, $f$is a neural network representing the rate of change, and $\theta$are the parameters of this network.

□ **Training Objective:**

The model is trained by minimizing the loss function, which measures the difference between observed and predicted values:



$$\mathrm{MSE} = (x_i - \hat{x}_i)^2$$

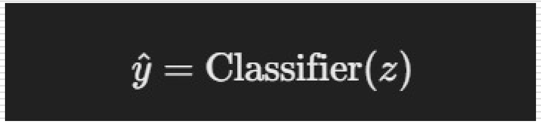
Where `x_i` is the actual class and `\hat{x}_i` is the predicted class.

Formula:

Common choices for Loss datainclude mean squared error (MSE) or binary cross-entropy for classification tasks.

□ **Classifier Integration:**

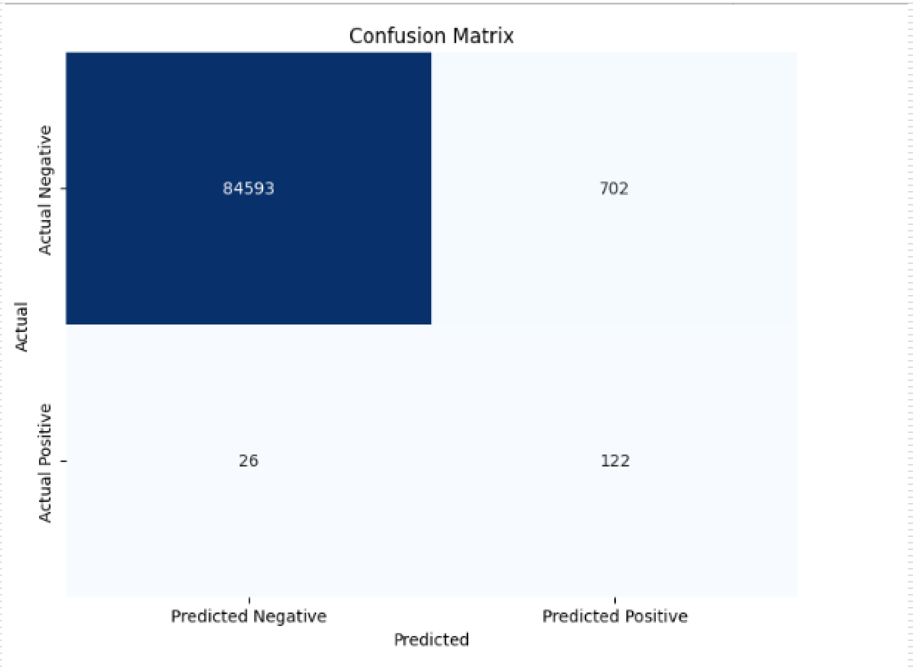
After obtaining the latent representation z, you can feed it into a classifier (e.g., XGBoost) to classify transactions as fraudulent or not:

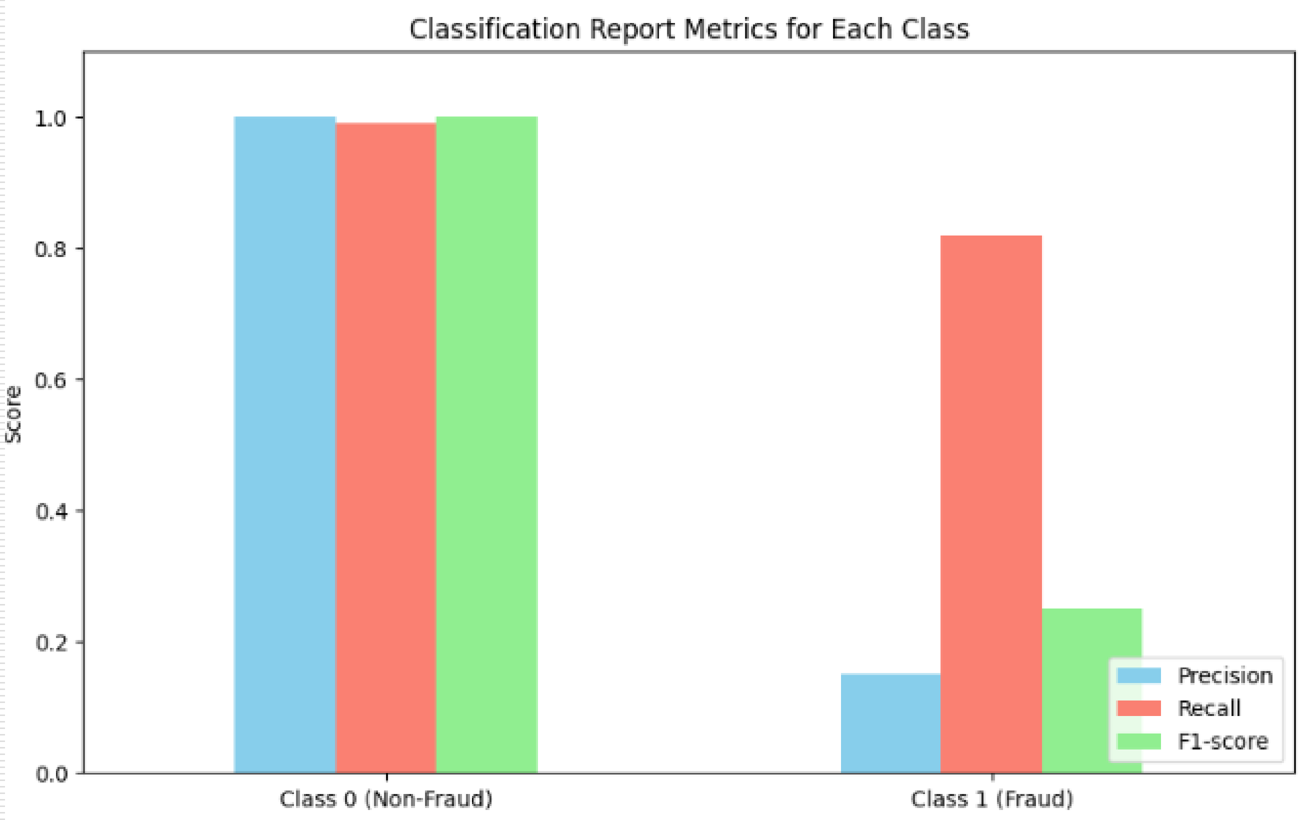


$$\hat{y} = \text{Classifier}(z)$$

The classifier provides predictions based on the latent representation derived from the ODE.

Output:

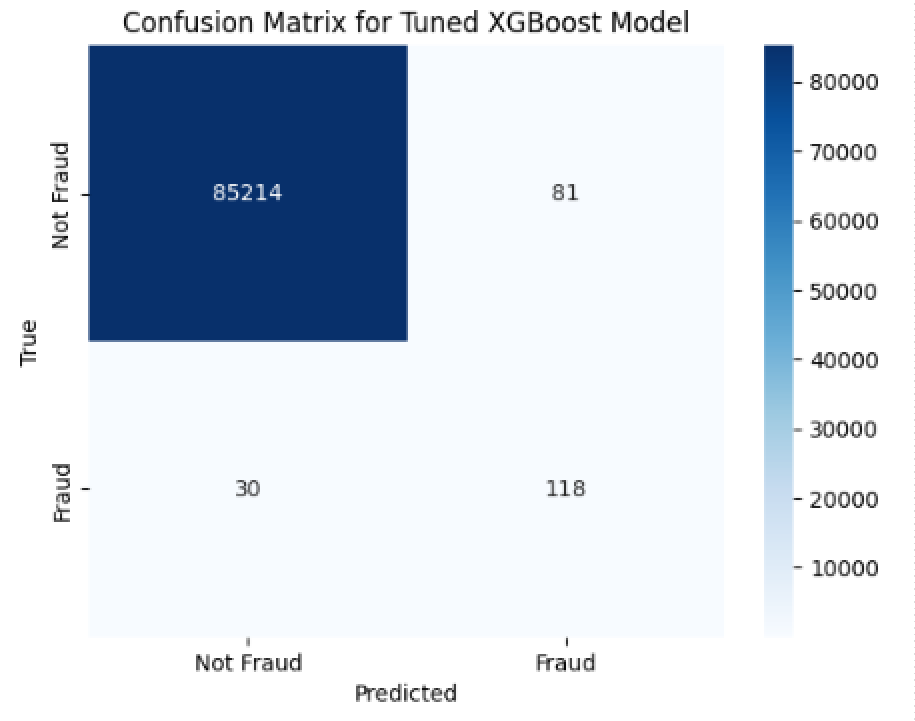




- **Untuned XGBoost classifier using latent representation from NOED**

- **Precision, Recall and F1 Score Metrics Evaluation**

Output:


Confusion Matrix for Tuned XGBoost Model
True
Not Fraud
Fraud
85214
81
30
118
Not Fraud
Fraud
Predicted
80000
70000
60000
50000
40000
30000
20000
10000

Conclusion:

- In conclusion, we implemented two advanced models, Neural ODE and XGBoost, to tackle credit card fraud detection. To address the significant class imbalance in the dataset, we applied SMOTE for synthetic oversampling, ensuring an even representation of fraud and non-fraud cases in the training set. This preprocessing step allowed both models to better capture the subtle patterns of fraudulent transactions. By combining dynamic feature learning with Neural ODE and the powerful gradient-boosting capabilities of XGBoost, we aimed to create a robust, effective solution for identifying fraud with improved accuracy.

References

- □ Srivastava, A., Kundu, A., Sural, S., & Majumdar, A. K. (2008). Credit Card Fraud Detection Using Hidden Markov Model. IEEE Transactions on Dependable and Secure Computing, 5(1), 37-48.
- □ Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3), 602-613.
- □ Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. Artificial Intelligence Review, 34(1), 1-14.

Thank You