



## **Adaptive AI for Enhanced Credit Card Fraud Detection**

Prepared By

**Name:** Bothina Ahmed Hessen Osman

**ID No.:** 243002096

Submitted in fulfillment of the requirements for the degree Master of  
Science in Artificial Intelligence

At

Collage of Informatics

Midocean University

UAE

**Supervisor Name:** Dr.Hagar Saleh

Collage of Informatics, Midocean University, Comoros

## Acknowledgements

First and foremost, I extend my deepest gratitude to Allah Almighty for granting me the strength, clarity, and perseverance needed to complete this research journey.

I would like to express my sincere appreciation to my supervisor, Dr. Hagar Saleh, for her continuous guidance, encouragement, and insightful feedback throughout the development of this work. Her expertise and support were essential in shaping the direction and quality of this thesis.

My heartfelt thanks go to my family—especially my mother—for her prayers, endless support, and unwavering belief in me. I am also profoundly grateful to my husband for her patience, understanding, and motivation during every stage of this project.

Finally, I wish to acknowledge my colleagues and friends for their valuable discussions, feedback, and moral support, which greatly contributed to completing this work successfully

# Abstract

Banks and other financial institutions are increasingly requiring smart systems that can spot fake credit card transactions in real time. The purpose is to cut down on money lost and help people make safe choices. We present a hybrid fraud-detection framework in this thesis that uses both Machine Learning (ML) and Deep Learning (DL) models to make classification more accurate in datasets that are very unbalanced. To address the significant issue of underrepresentation of minority classes, a critical challenge in fraud detection tasks, four imbalance-handling techniques were employed: random oversampling, SMOTE, random undersampling, and Tomek Links.

The proposed framework employs a methodical pipeline encompassing data preprocessing, normalization, feature engineering, model optimization via GridSearchCV, and comprehensive evaluation using standard metrics. We used six machine learning algorithms as baseline learners: logistic regression, K-nearest neighbors, decision tree, random forest, Naïve Bayes, and XGBoost. We changed their hyperparameters to achieve the best AUC and F1-score results. We also used three deep learning (DL) models: Deep Neural Network (DNN), Simple Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM). These models were able to identify patterns of fraud that regular machine learning (ML) models might miss because they are nonlinear, sequential, or temporal. Experimental findings demonstrate that imbalance-handling techniques significantly influence model performance, with SMOTE consistently improving detection results in both ML and DL models. The LSTM and SMOTE worked together the best of all the methods tested. They were the best at finding rare fraudulent transactions. The full evaluation using

accuracy, precision, recall, F1-score, AUC, and Confusion Matrices shows that the proposed hybrid ML-DL framework is better than traditional standalone models. It is a reliable and scalable way to identify financial fraud in real-world situations confusion matrices.

# Contents

|                                                             |           |
|-------------------------------------------------------------|-----------|
| <b>Acknowledgements.....</b>                                | <b>2</b>  |
| <b>Abstract.....</b>                                        | <b>3</b>  |
| <b>List of Abbreviations .....</b>                          | <b>6</b>  |
| <b>Chapter 1: Introduction .....</b>                        | <b>8</b>  |
| <b>Chapter 2: Overview .....</b>                            | <b>15</b> |
| <b>Chapter 3: Related Work.....</b>                         | <b>35</b> |
| 3.1 Introduction .....                                      | 35        |
| 3.2 History .....                                           | 35        |
| <b>Chapter 4: Proposed Approach.....</b>                    | <b>41</b> |
| 4.1 Introduction .....                                      | 41        |
| 4.2 Dataset.....                                            | 43        |
| 4.3 Preprocessing Dataset .....                             | 44        |
| 4.4 Feature Selection Phase and Method .....                | 47        |
| 4.5 Model training phase.....                               | 49        |
| 4.6 Summary .....                                           | 55        |
| <b>Chapter 5: Experimental Results and Discussion .....</b> | <b>57</b> |
| 1.1 Introduction.....                                       | 57        |
| 1.2 Results of Individual Models.....                       | 58        |
| 1.3 Results of Proposed Stacking Framework.....             | 60        |
| 1.4 Discussion of Findings .....                            | 68        |
| 1.5 Limitations of The Proposal Work .....                  | 70        |
| 1.6 Summary of the Chapter .....                            | 71        |
| <b>Chapter 6: References .....</b>                          | <b>73</b> |

## List of Abbreviations

| Abbreviation | Meaning                                   |
|--------------|-------------------------------------------|
| <b>AI</b>    | Artificial Intelligence                   |
| <b>ML</b>    | Machine Learning                          |
| <b>DL</b>    | Deep Learning                             |
| <b>DNN</b>   | Deep Neural Network                       |
| <b>RNN</b>   | Recurrent Neural Network                  |
| <b>LSTM</b>  | Long Short-Term Memory                    |
| <b>NN</b>    | Neural Network                            |
| <b>ANN</b>   | Artificial Neural Network                 |
| <b>MLP</b>   | Multi-Layer Perceptron                    |
| <b>SMOTE</b> | Synthetic Minority Oversampling Technique |
| <b>ROS</b>   | Random Oversampling                       |
| <b>RUS</b>   | Random Undersampling                      |
| <b>TL</b>    | Tomek Links                               |
| <b>CV</b>    | Cross-Validation                          |
| <b>GS</b>    | Grid Search                               |
| <b>XGB</b>   | Extreme Gradient Boosting (XGBoost)       |
| <b>DT</b>    | Decision Tree                             |
| <b>RF</b>    | Random Forest                             |
| <b>KNN</b>   | K-Nearest Neighbors                       |
| <b>LR</b>    | Logistic Regression                       |
| <b>NB</b>    | Naïve Bayes                               |

| <b>Abbreviation</b> | <b>Meaning</b>                                  |
|---------------------|-------------------------------------------------|
| <b>FS</b>           | Feature Selection                               |
| <b>EDA</b>          | Exploratory Data Analysis                       |
| <b>FN</b>           | False Negative                                  |
| <b>FP</b>           | False Positive                                  |
| <b>TN</b>           | True Negative                                   |
| <b>TP</b>           | True Positive                                   |
| <b>PCA</b>          | Principal Component Analysis                    |
| <b>AUC</b>          | Area Under the Curve                            |
| <b>ROC</b>          | Receiver Operating Characteristic               |
| <b>F1-score</b>     | Harmonic Mean of Precision and Recall           |
| <b>API</b>          | Application Programming Interface               |
| <b>CSV</b>          | Comma-Separated Values                          |
| <b>SHAP</b>         | SHapley Additive exPlanations                   |
| <b>LIME</b>         | Local Interpretable Model-Agnostic Explanations |
| <b>GPU</b>          | Graphics Processing Unit                        |
| <b>CPU</b>          | Central Processing Unit                         |

# Chapter 1: Introduction

## Introduction

Over the last ten years, preventing credit card fraud has emerged as a major financial institution's priority across the globe, as the explosion of online transactions and digital payment platforms has made fraud even more sophisticated and difficult to detect in real time. Banks and platforms such as these are using precise and timely fraud-prediction systems in order to help reduce the scope of loss to banks, increase customer confidence, and maintain safe and secure transaction environments. The proliferation of e-commerce platforms and mobile banking apps has also led to the emergence of large transaction logs consisting of detailed electronic traces for each purchase made, login attempt, card swipe, or the digital move of a device that can display behavioral characteristics of valid or inappropriate behavior (Sahin et al., 2023). [1]. Artificial intelligence (AI)—especially machine learning (ML) and deep learning (DL)—has been tremendously successful at detecting financial fraud, as such models are also able to reveal intricate relations that do not follow a linear structure on large transactional volumes of data, which conventional statistical methods usually cannot capture. Machine learning (ML) models learn from past transaction data to classify new data, and neural networks like recurrent neural networks represent complex time and sequence relations, which typically characterize fraud behavior (Bhattacharyya et al., 2011). [2]. Further, sophisticated DL architectures such as Long Short-Term Memory (LSTM) networks have shown the ability of accurately detecting subtle deviations in transactional streams, which is important when it comes to fraud analytics (Zheng et al., 2018) [3]. Yet, a common problem in fraud detection is that legitimate and fraudulent transactions are extremely imbalanced, and fraud cases comprise less than 0.2% of all data. This imbalance can lead to ML and DL models

biased to predict non-fraud. Therefore, applying resampling methods such as Random Oversampling, SMOTE, Random Undersampling, and Tomek Links is important to increase the minority-class sensitivity to minority patterns and improve general classification performance (He & Garcia, 2009). [4]. In addition, correct feature preprocessing—normalization and dimensionality reduction, attenuating noise, or handling the noise—is vital for achieving the best quality of model learning and reducing misclassification rates. Financial fields are particularly sensitive, where mistakes in fraud predictions can result in enormous monetary losses, customer dissatisfaction, operational disruptions, and so forth. So, explainable artificial intelligence (XAI) has been growing in significance and importance to offer the transparency and trust of fraud-detection systems. Tools like SHAP and LIME help investigators make sense of the reasons behind each model decision, allowing financial institutions to justify the fraud alerts made, satisfy regulatory obligations, and optimize the risk strategies accordingly (Carcillo et al., 2021). [5]. Using ML, DL, imbalance-handling techniques, and explainable AI, this study aims to design a comprehensive and powerful framework for fraud detection, allowing high accuracy in the detection of rare fraudulent transactions. The idea behind this research is enabling financial institutions to move from reactive fraud management to an effective preventive tool for data-based fraud so as to ensure customer protection for consumers while strengthening the overall digital security and integrity of financial systems.

## **1.1 Problem Definition**

Despite recent advancements in financial technology, credit card fraud remains a significant threat to global financial organizations, resulting in

annual losses of billions of dollars. Conventional fraud-detection systems primarily focus on preset rules and manual inspection with static threshold detection methods and thus do not always recognize changes in both fraud patterns and attack strategies. Given the growing number of transaction volumes and the increased adaptable nature of fraud, traditional detection mechanisms generally fail to detect abnormal behaviors in real time, leading to delayed detection and more frequent false positives. In credit card fraud detection, a main difficulty is the extreme class imbalance between legitimate and fraudulent transactions; that is, typical fraudulent transactions constitute less than 0.2% of the data. This terrible class imbalance leads to machine learning classifiers becoming biased toward predicting non-fraud cases, thereby lowering their sensitivity to rare fraudulent occurrences and making it more likely these actions will not be detected. Also, most transactional datasets are high-dimensional and heterogeneous, with transactions with intricate features such as spending patterns, time series information, device identifiers, and geographic information—making it difficult to isolate the most relevant factors in discriminating legitimate from malicious transactions. Bringing irrelevant and noisy patterns into predictive models also increases the tendency to overfit the model, reduces the extent to which the classification may be representative, and makes interpretation of models difficult. Although machine learning (ML) and deep learning (DL) methods have proven ability to observe non-linear and sequential patterns of fraud, their accuracy is very dependent on the successful handling of imbalance, high-quality feature preprocessing, and transparent interpretability that are suitable for the real-life financial domain. Thus, this study answers the challenge to have a robust and data-based fraud detection framework combining optimized ML and DL models with a dynamic, advanced resampling approach. The aim is to

facilitate the classification of fraud, improve the detection of minority-class frauds, and for financial institutions involved to be more accurate when making decisions, to enable them to take decisions based on evidence, and to understand more clearly the indicators of behavior that would be associated with fraudulent transactions.

## **1.2 Thesis Objectives**

The main aim of this thesis is to develop an intelligent fraud-detection framework using applied artificial intelligence methodologies and to evaluate the success of using machine learning (ML) and deep learning (DL) models for the identification of credit card transactions as valid or fraudulent. Through integrating the various methods for handling imbalances, such as random oversampling, SMOTE, random undersampling, and Tomek links, this research will improve the detection of minority classes by mitigating the performance degradation associated with very imbalanced financial datasets. Further, the work will analyze and compare a variety of ML and DL classifiers to analyze their performance (both separately and together) over complex, nonlinear, and sequential fraud patterns. By carefully tuning for hyperparameters and using strict validation and evaluation metrics, the thesis will strive to find a reliable and scalable model for real-world fraud detection scenarios. The ultimate aim is to construct a comprehensive, data-driven detection framework to enable early identification of fraudulent activities, reduce false positives, improve financial security, and empower institutions to make data-driven decisions to protect customers and virtual payment systems.

### 1.3 Thesis Contribution

Major contributions of this thesis are as follows:

1. Suggested an all-encompassing model-based data preprocessing and imbalance-handling pipeline that combines four resampling methods (Random Oversampling, SMOTE, Random Undersampling and Tomek Links) to tackle the drastic class imbalance associated with credit card fraud datasets. This serves to increase minority-class representation, balance out the influence of legitimate transactions, and to allow models to capture rare patterns of fraud, while maintaining efficiency and representational clarity.
2. We utilized six different types of machine learning (ML) classifiers in order to find complex transactional behaviors tied to fraud using Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest, Naïve Bayes, and XGBoost. Both linear and non-linear ML models are used to achieve good predictive capabilities while using multiple fraud scenarios, thereby ensuring robust pattern discovery and increasing predictive power on high-dimensional financial datasets.
3. Designed and tested 3 state of the art Deep Learning (DL) methods – Deep Neural Network (DNN), Simple Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) – to model complex networks of dynamic temporal, sequential, and nonlinear dependencies in a stream of transactions. These models extend the utility of the framework by enhancing the predictive power of the model and analyzing dynamic patterns of behavior which classical ML analysis tends to overlook.
4. Using GridSearchCV, all ML models were optimised to determine best hyperparameters for each imbalance-handling technique, yielding pronounced improvements in the F1-score, AUC, and fraud-detection sensitivity. The vast

comparison over 24 ML setups and three DL architectures represents a rigorous benchmark to inform the best models–methods pairing for real life credit card fraud detection.

5. Used a unified evaluation framework with the Accuracy, Precision, Recall, F1-score, ROC–AUC, and Confusion Matrices in model evaluation to thoroughly evaluate its performance. This standardisation ensures fair comparison between ML and DL results and shows that SMOTE-augmented LSTM can effectively classify small minority-class variations with a high degree of discrimination capability.
6. The proposed framework combines several ML/DL models, with advanced imbalance treatment, results in an efficient fraud detection system and is easily extendable toward financial institutions. The contributions help to detect fraud quickly and efficiently, manage operational risks and provide evidence-based support to create more secure, transparent and scalable fraud-detection solutions.

## 1.4 Thesis Organization

This dissertation is composed as follows:

**Chapter 2** summarizes main theoretical concepts relevant to the field of Artificial Intelligence (AI), which can be classified into Machine Learning (ML) and Deep Learning (DL), and discusses their importance in the sphere of financial fraud detection. It covers basics such as binary classification, sequential modeling, anomaly detection and problems related to data imbalance. The chapter focuses on preprocessing and feature engineering in transactional analytics, modern fraud detection techniques and tools

**Chapter 3** discusses recent academic literature, landmark studies, and comparative studies in credit card fraud detection. The paper investigates the effectiveness of different ML and DL models, different imbalance-handling strategies, and the existing

research gaps—especially in model optimization and temporal modeling, and the demand of more robust evaluation frameworks. From this perspective, the present work situates this thesis in the context of fraud analytics research more generally.

**Chapter 4** research design and methodology adopted in this study is described. It details dataset, preprocessing steps, normalization methodology, and implementation of the 4 imbalance-handling methods (Random Oversampling, SMOTE, Random Undersampling, Tomek Links). The chapter further justifies the classification and setup of ML/DL models, hyperparameter optimizations using GridSearchCV, and metrics used to evaluate model performance.

**Chapter 5** “Experimental setup” which provides the experimental setting and results obtained with different imbalance-handling arrangements of all ML and DL classifiers. It contrasts this with Accuracy, Precision, Recall, F1-score, AUC, and Confusion Matrices, detailing how they can be analyzed. The chapter elaborates on the effect of different resampling techniques on model performance, the efficiency of deep learning in learning sequential fraud models and proposes the best models when used in all combinations.

Finally, in **Chapter 6** key findings, contributions, and implications are summarized. It emphasizes the importance of combining ML, DL, and imbalance-handling methods for increasing detection of fraud in financial transactions. The chapter closes by providing functional recommendations for financial institutions and directions for future work such as real-time fraud prediction, explainable AI integration, model deployment in a production setting.

# Chapter 2: Overview

## 2.1 Introduction

AI (Artificial Intelligence) and its related sub-disciplines are growing areas in supporting and enriching the emerging trend in fraud analytics, especially in detecting credit card fraud. AI includes fundamental topics related to the analysis of financial data, including Machine Learning (ML), Deep Learning (DL), Large Language Models (LLMs), and Explainable AI (XAI). These various technologies are helping us interpret and forecast the patterns of fraud in these increasingly complicated financial markets. ML techniques can help build predictive classification models that provide a better classification of legitimate and fraudulent transactions, whereas DL models have been used to learn the depth, non-linearity, and sequential characteristics present in large datasets of transactions. LLMs come into play by being able to analyze raw financial text (customer complaints, fraud reports, etc.) to provide more context. XAI can enhance transparency around model decisions where analysts and compliance teams can understand fraud-detection decisions, promoting accountability and trust. In this chapter, it begins by establishing the basic principles of fraud studies, emphasizing that AI plays a vital role in enabling these data-driven detection systems to operate more effectively. It includes details of binary classification, anomaly detection, and sequential algorithms applicable to finance, as well as dilemmas arising from dealing with highly imbalanced fraud datasets. We go through the main ML and DL algorithms for a fraud classification procedure and summarize those basic preprocessing steps—normalization, feature engineering, and imbalance-handling techniques—that promote the development of high-quality and stable detection methodologies in this domain. At last, the chapter reviews some current trends about AI-based fraud detection and demonstrates

how AI is enabling us to move from traditional reactive to proactive and intelligent solutions for real-time finance protection.

## **2.2 Artificial Intelligence (AI)**

Artificial intelligence/AI is a vast, evolving field that tries to mimic people's intelligence and decision making through tech. For the financial industry, AI is a critical enabler of reliable, secure, and accurate systems in fraud detection. It gives automatic processes the freedom to mine huge volumes of transactional information, identify suspicious transactions, and assist banks and financial institutions in preventing unauthorized transactions without human intervention. AI encompasses a number of categories of topics, all driving fraud analytics, including Machine Learning (ML), Deep Learning (DL), and Large Language Models (LLMs). Machine Learning (ML) approaches enable systems to develop their own knowledge of observed transaction patterns and to identify whether these behaviors are good or bad. DL models expand further, learning richer non-linear relationships that extend the feature set of deep learning models over more complex, non-linear sequences, which characterize sophisticated forms of fraudulent acts. LLMs can interpret unstructured text-based data such as fraud reports, customer complaints, and financial documents, which provide useful contextual information about the fraud behavior. This way, we can witness how the AI subfields in their entirety serve as the underpinning for modern intelligent systems that shield financial systems as they battle shifting fraud schemes. Fig. 1

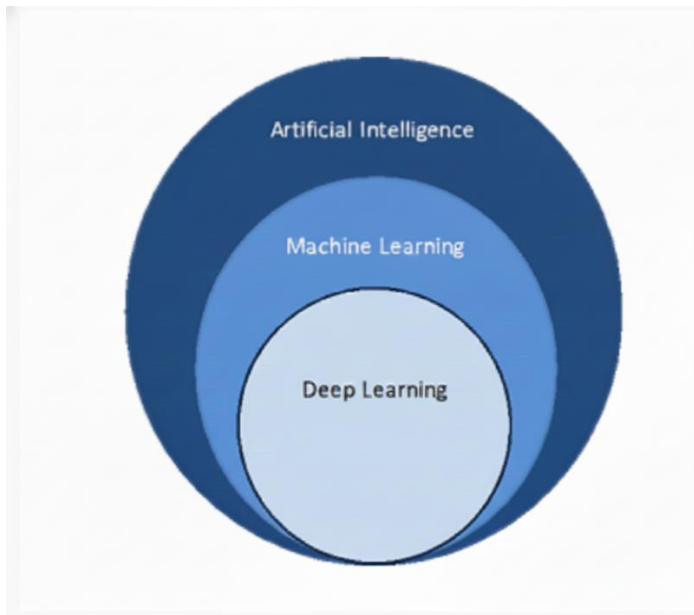


Fig. 1 Artificial Intelligence (AI)

### 2.3 Machine Learning (ML)

Machine Learning (ML) is an essential domain in Artificial Intelligence and enables systems to identify patterns of behavior in stored data and predict behavior without explicit instructions. One of these ML methods is especially well-suited for working with structured tabular data types such as data on financial transaction logs, used to detect credit card fraud. ML models can be trained based on data and metrics about the statistical association of behaviors and classify deals as legitimate or fraudulent and support financial institutions in real-time detecting suspicious activities. ML usually has two flavors: supervised learning — as models work on labeled data to perform classification and regression — and unsupervised learning, in which learning finds hidden structures in unlabeled data, for example clustering similar transaction behaviors. Supervised learning (such as classification) is more widely utilized in fraud detection and is of greater importance in distinguishing between genuine and fraudulent activities based on past transactions. Fig. 2

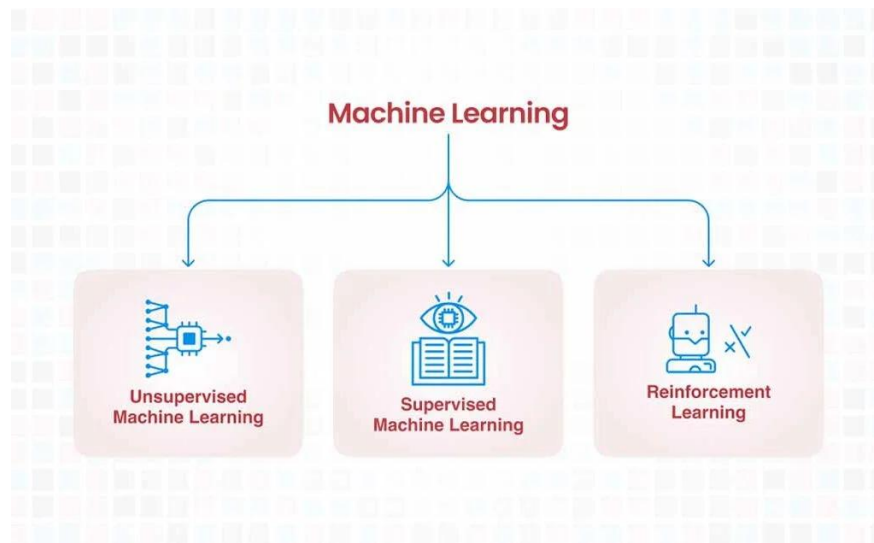


Fig. 2 Machine Learning (ML)

## 2.4 Deep Learning (DL)

Deep Learning (DL) is a subfield of machine learning that involves the creation of multiple layers of artificial neural networks consisting of several hidden layers. Deep learning architectures are tailored for exploiting the features that emerge from raw data, which is a much-admired approach to sequential, high-dimensional, or unstructured problems. The financial domain relies heavily on DL, as this area is where fraud detection within credit cards lies at the heart of it, as there is no static representation of complexity, and the detected fraudulent behavior usually follows the pattern of its very unique dynamic nature that some existing ML models may find difficult to detect. Models including Deep Neural Networks (DNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) are widely used to model transaction patterns, time dependencies, and anomalies in users' spending behaviors. DL models pay attention to history-based transaction flow; hence, learning is valuable for recognizing increasingly sophisticated and evolving fraud attempts across time.

Fig. 3

# Deep Learning

A subfield of Machine Learning based on deep neural networks

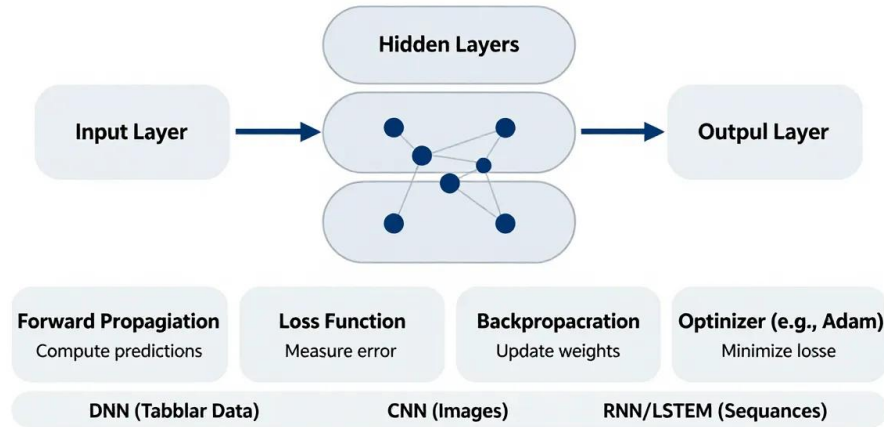


Fig. 3 Machine Learning (ML)

## 2.5 ML Models Used in Fraud Detection

**Logistic Regression (LR):** is a supervised learning algorithm used for binary classification problems. The process relies on a logistic (sigmoid) function to model the probability that a transaction belongs to the fraud class. While being simple, LR works at a great level for linearly separable data with high interpretability and is one of the frequent baseline models in fraud detection. Fig.

4

$$L = -\frac{1}{m} \sum_{i=1}^m \left[ y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$

Fig. 4 Logistic Regression (LR)

**K-Nearest Neighbors (KNN):** is an instance-based learning protocol that evaluates a transaction by looking into the labels of the nearest neighbors in the dataset. The model also uses distance measures to calculate similarity. In the context of fraud detection, KNN could learn local patterns of behaviors,

but it might struggle with high-dimensional data and large datasets at a computational cost. Fig.5

$$d(x, y) = \sqrt{\sum_{j=1}^n (x_j - y_j)^2}$$

Fig. 5 K-Nearest Neighbors (KNN)

**Decision Tree (DT):** It learns simple decision rules based on all features in decision trees used to classify that data. They divide transaction data into hierarchical branching, which results in classification outcomes. DTs can be read intuitively and learn nonlinear relationships but often have a high probability of overfitting if not pruned. Fig.6

$$Information\ Gain = Entropy(parent) - \sum_j \frac{N_j}{N} Entropy(j)$$

Fig. 6 Decision Tree (DT)

**Random Forest (RF):** is an ensemble model that combines multiple decision trees trained on random subsets of data and features. It improves prediction precision and reduces overfitting. RF is commonly used for fraud detection because of its robustness, its capability of dealing with imbalanced data, and its ability to reveal the most critical features. Fig. 7

$$\hat{y}(x) = \frac{1}{B} \sum_{b=1}^B \hat{y}^{(b)}(x)$$

Fig. 7 Random Forest (RF)

**Naïve Bayes (NB):** is a probabilistic classifier based on Bayes' theorem—the assumption that features are conditionally independent. With real-world fraud data, this assumption is rarely the case. NB is computationally efficient and applies extremely nicely when the dataset is high-dimensional or noisy. Fig. 8

$$P(C_k | X) = \frac{P(X | C_k) P(C_k)}{P(X)}$$

Fig. 8 Naïve Bayes (NB)

**XGBoost (Extreme Gradient Boosting):** is a strong boosting algorithm and builds trees sequentially, with each new tree correcting previous mistakes. It offers accuracy, speed, and regularization. In a particular application, fraud detection, XGBoost has been effective in capturing intricate relations and recognizing very subtle fraudulent behaviors. Fig. 9

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), \quad f_k \in \mathcal{F}$$

Fig. 9 XGBoost (Extreme Gradient Boosting)

## 2.6 DL Models Used in Fraud Detection

**Deep Neural Network (DNN):** A DNN is a type of neural network that has many fully connected layers that learn how to represent attributes in a hierarchical way. DNNs are good for fraud detection tasks that need a deeper understanding of patterns because they can find complex nonlinear relationships in transactional data. Fig. 10

### Deep Neural Network for Credit Card Fraud Detection

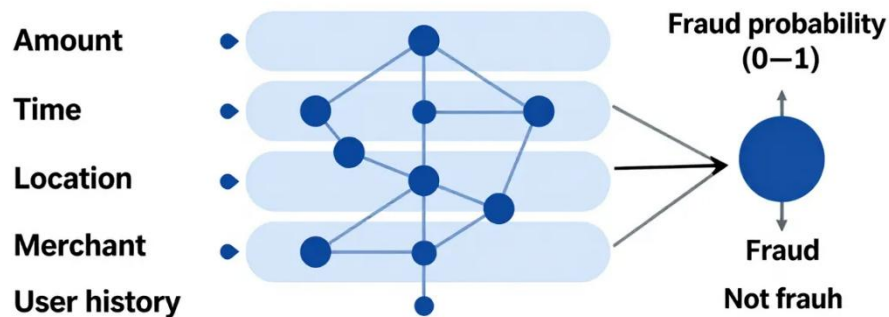


Fig. 10 Deep Neural Network (DNN)

**Recurrent Neural Network (RNN):** A neural network architecture designed for sequential data processing, where connections between nodes form directed cycles, allowing information to persist across time steps. RNNs maintain internal memory states that capture temporal dependencies in sequences, making them suitable for time-series analysis, natural language processing, and sequential

decision-making tasks. However, traditional RNNs suffer from vanishing gradient problems in long sequences, often requiring advanced variants like LSTM or GRU for better performance.

relationships in transactional data. Fig. 11

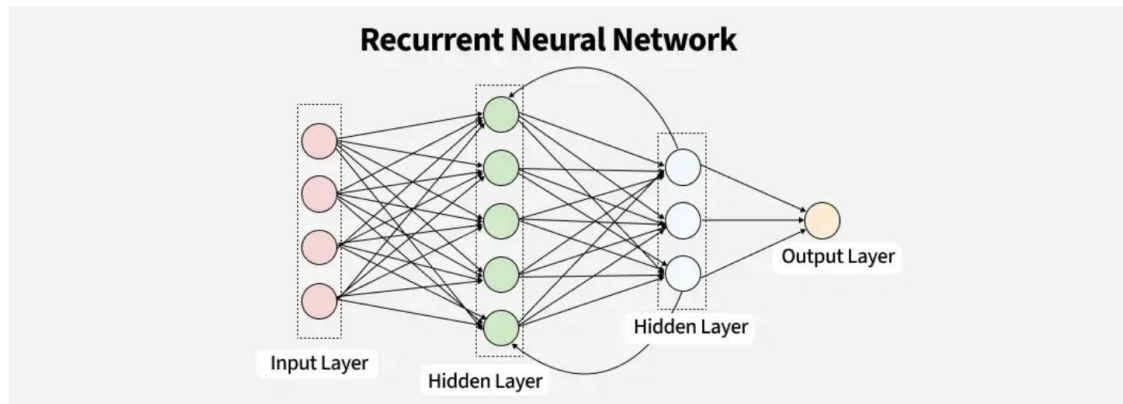


Fig. 11 Neutral Network (RNN)

**Long Short-Term Memory (LSTM):** is a complex RNN structure that can find long-term dependencies using gating mechanisms. LSTMs are better than regular RNNs at finding fraud because they can find patterns of fraud that change over time. Fig. 12

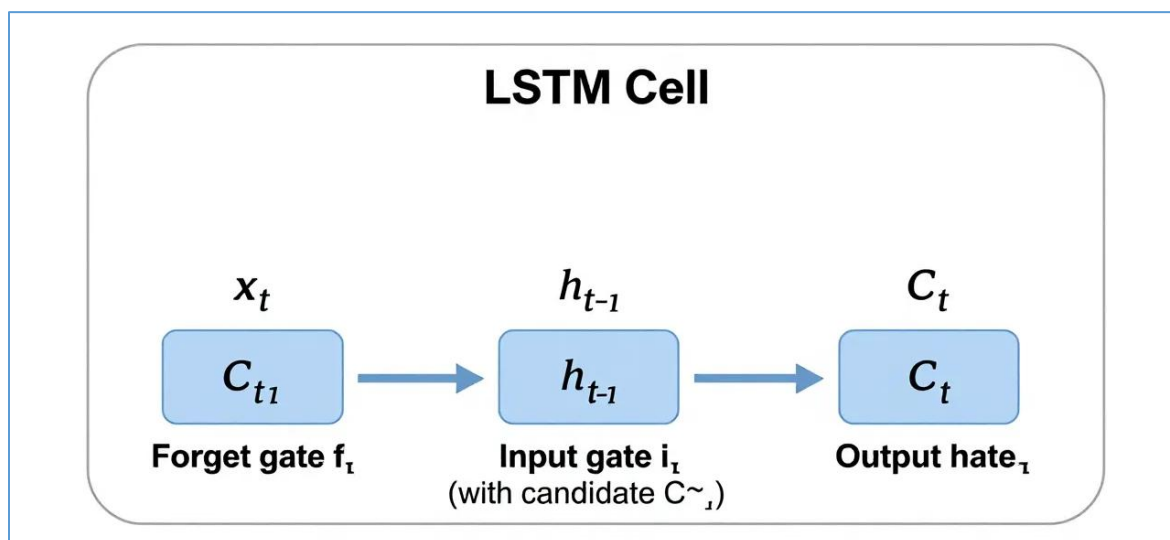


Fig. 12 Long Short-Term Memory (LSTM)

## 2.7 Imbalance Handling Methods Used in the Project

Fraud datasets contain an extremely imbalanced distribution, where fraudulent transactions only account for a small number of transactions. Thus, resampling methods are critical Fig. 13

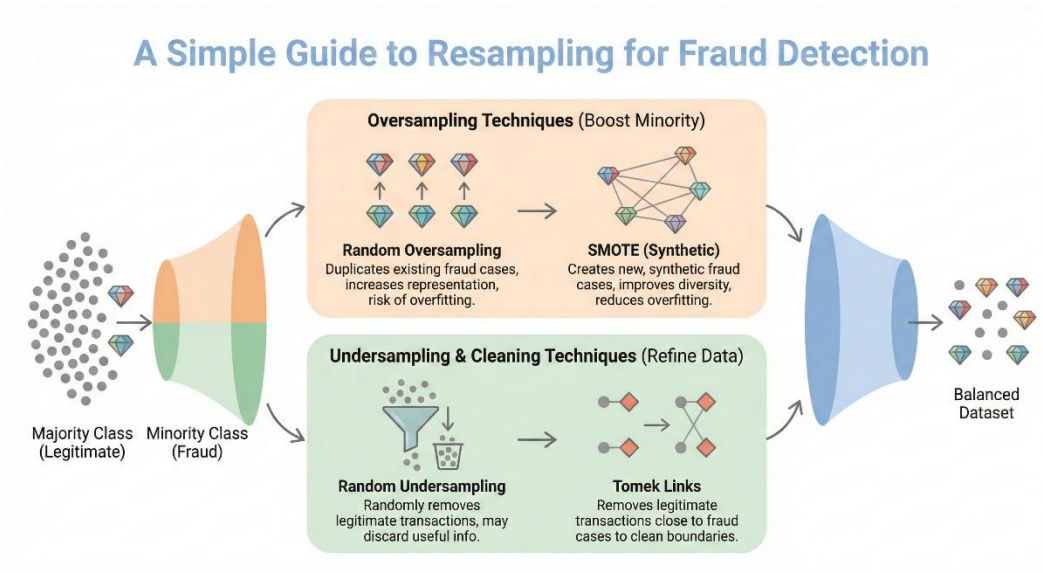


Fig. 13 Simple Guide to Resampling for Fraud Detection

**Random Oversampling:** This approach duplicates minority-class samples (fraud cases) to create balance in the dataset. This increases the sensitivity to fraud while increasing the likelihood of overfitting since the duplicated samples add no new information. Fig. 14

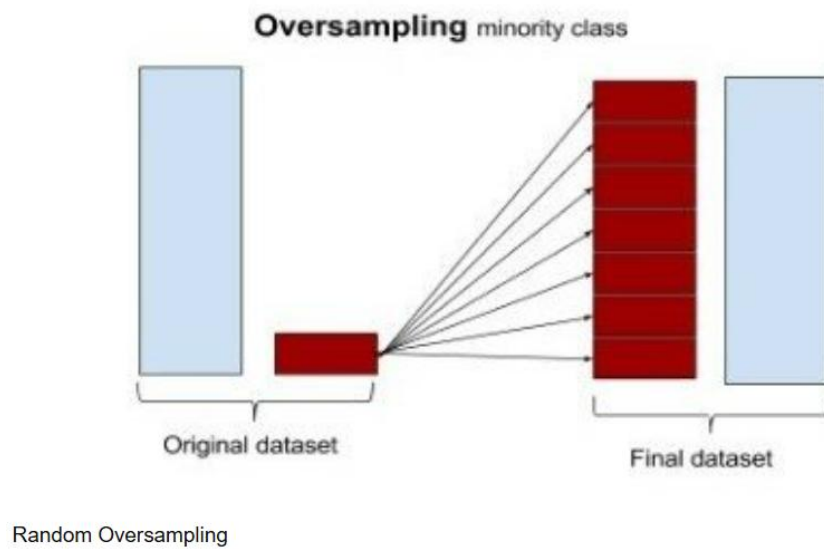


Fig. 14 Random Oversampling

**SMOTE:** Creates synthetic minority-class samples by interpolating between existing fraud cases and their nearest neighbors. It improves diversity of the minority class while decreasing overfitting, compared to basic oversampling. SMOTE is widely considered one of the most efficient methods for fraud detection. Fig. 15

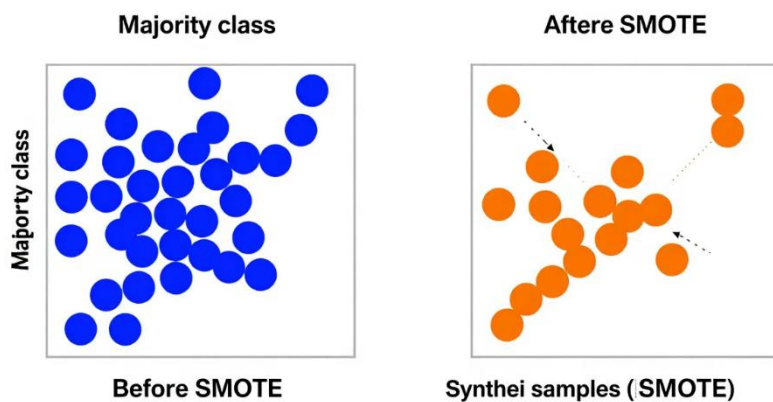


Fig. 15 SMOTE

**Random Undersampling:** This reduces the majority (legitimate transactions) class randomly by removing samples. At the same time, it keeps the class balance with less compute expenses but also discards useful information, causing model performance to degrade. Fig. 16

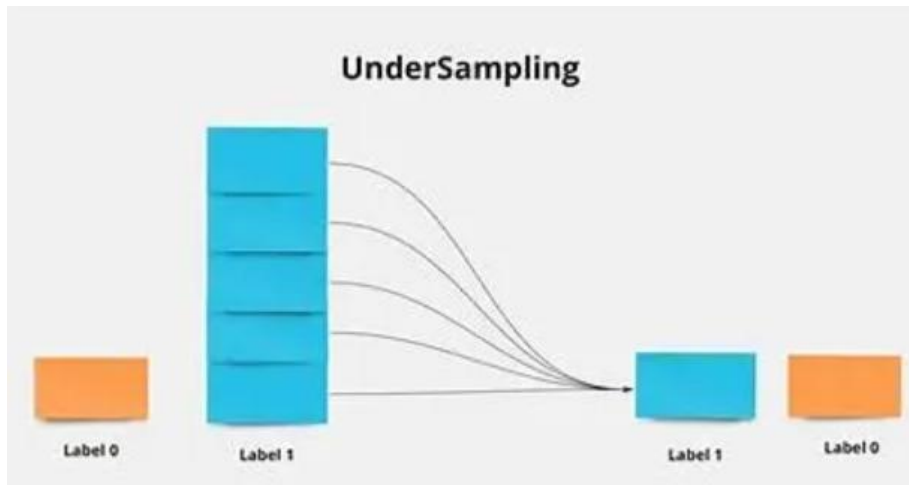


Fig. 16 UnderSampling

### 1. Tomek Links

Refer to pairs of nearest neighbors from opposite classes that are very close in feature space. Removing the majority-class instance cleans up your dataset and makes class boundaries separate. This has been commonly implemented as a data-cleaning and boundary-refinement approach in fraud detection. Fig. 17

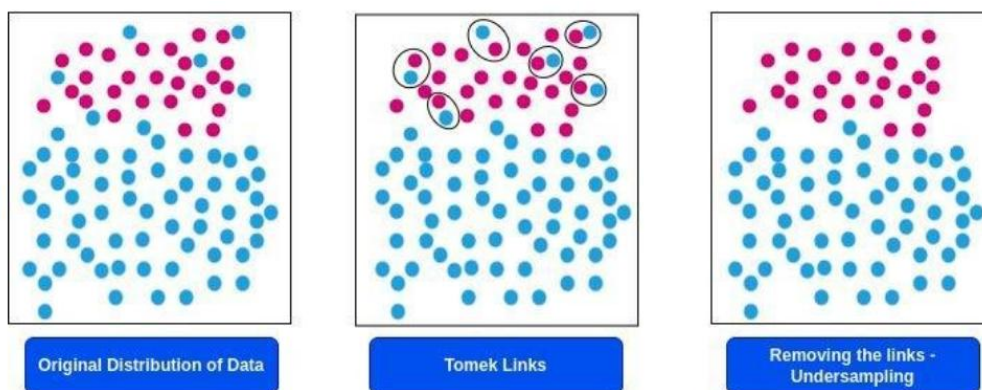


Illustration of Tomek Links.

Fig. 17 Tomek Links

## 2.8 Explainable AI (XAI)

Explainable AI (XAI) refers to AI systems that offer transparent decision-making processes. XAI is crucial in education to ensure that predictions made by ML and DL models can be understood and trusted by educators and students. It helps in identifying which features influenced a prediction and supports ethical use of AI. Fig. 18

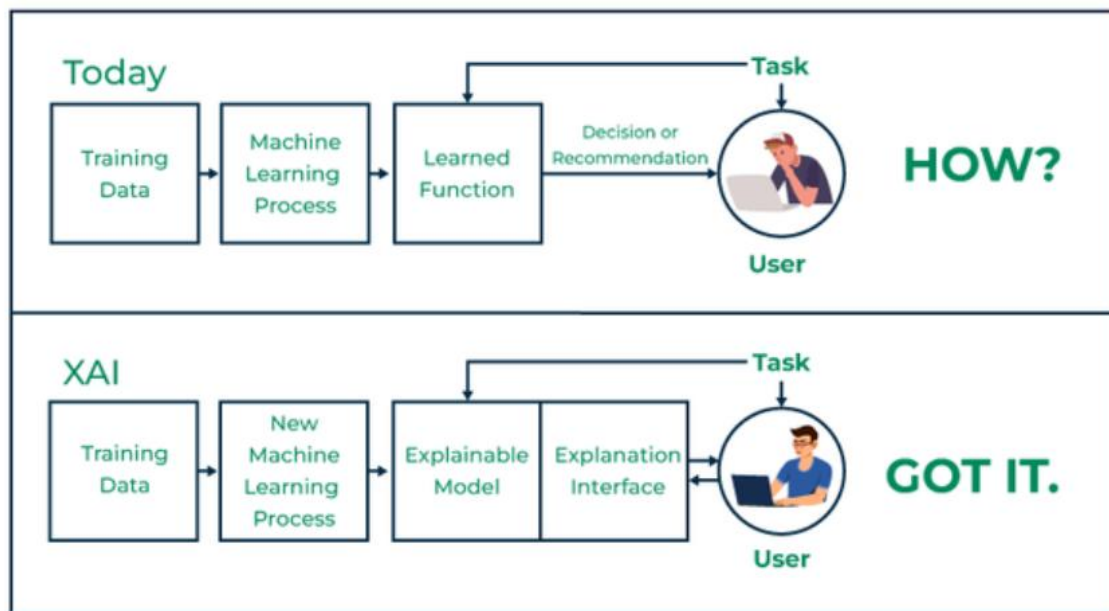


Fig. 18 XAI

**SHAP (SHapley Additive exPlanations):** is one of the most widely used explainability methods for understanding how prediction models make their decisions. It is based on cooperative game theory, where each feature contributes a certain “value” to the prediction, similar to how players contribute to the outcome of a game. In fraud detection, SHAP provides a clear interpretation of which transaction features (such as amount, time, merchant category, or spending patterns) influence the classification of a transaction as fraudulent. SHAP summary plots and feature importance visualizations help financial analysts and compliance teams verify model

decisions, improve trust, and detect potential false positives by understanding the factors behind each prediction. Fig.19

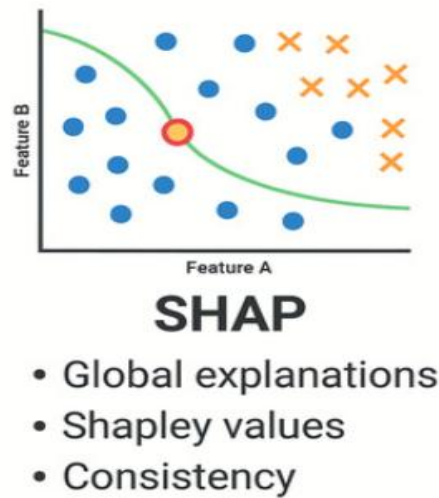
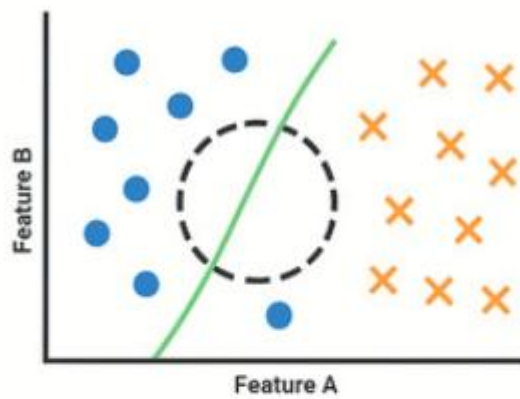


Fig. 19 XAI (SHAP)

### **LIME (Local Interpretable Model-agnostic Explanations):**

is an explainability method designed to interpret individual predictions rather than the entire model. It works by building a simple, local surrogate model around a specific instance to explain why the classifier labeled a particular transaction as fraud or legitimate. In credit card fraud detection, LIME is especially useful for analyzing high-risk transactions, as it highlights the features that contributed the most to the model's decision. This makes it easier for fraud investigators to validate alerts, understand unusual spending behavior, and take proper actions with improved transparency. Fig.20



## LIME

- Local explanations
- Linear approximation
- Instability

Fig. 20 XAI (LIME)

### 2.9 Hyperparameter Optimization (GridSearchCV):

is a systematic method used to identify the best hyperparameter configuration for a given machine learning model. It performs an exhaustive search over a predefined set of hyperparameter values and evaluates each combination using cross-validation. This ensures that the model does not overfit and achieves strong generalization performance on unseen data.

In fraud detection, hyperparameter optimization is crucial because models must balance sensitivity to fraudulent transactions (recall) and the ability to avoid false alarms (precision). By tuning parameters such as the number of trees in Random Forest, learning rate in XGBoost, or number of neighbors in KNN, GridSearchCV ensures that the model achieves optimal performance when dealing with highly imbalanced and noisy financial datasets. Fig.21

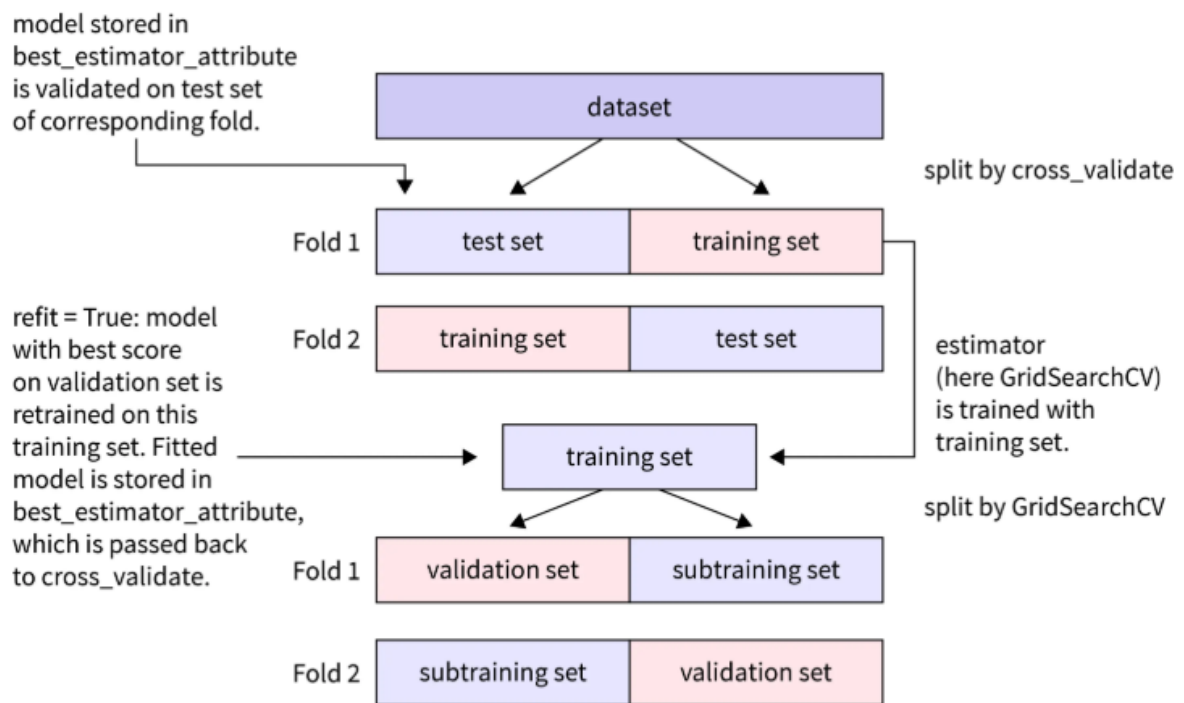


Fig. 21 GridSearchCV

## 2.10 Evaluation Metrics for Fraud Detection

Metrics for Fraud Detection. Because fraud datasets are highly imbalanced, traditional accuracy isn't enough. Rather, the effectiveness of a model to identify fraudulent cases is determined by specific metrics.

**Accuracy** is the ratio of all predictions that were classified correctly by the model. Useful as it is in many domains, especially in fraud detection where legitimate transactions dominate, it can cause a misleading result, as a model could score high accuracy with no fraud detection. Fig.22

## Accuracy Evaluation Methods

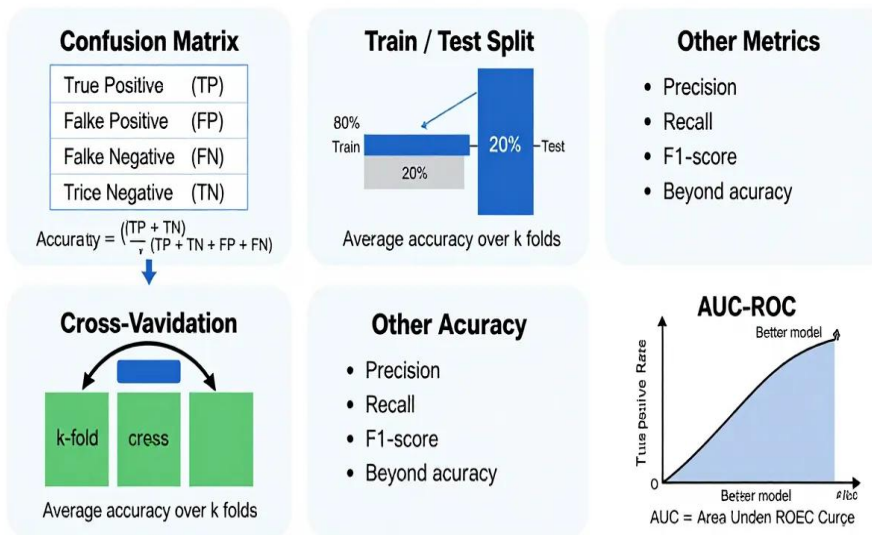


Fig. 22 Accuracy Evaluation Methods

**Precision:** indicates how many of the transactions predicted as fraud are fraudulent in action. This high precision indicates that few false alarms are generated by the model, which may be required to prevent customers from being disrupted unnecessarily. Fig.23

## Precision in Clasciification

|                     |                    |
|---------------------|--------------------|
| Positive Prediction | True Positive (TP) |
| Negative Prediction | FP                 |

$$\text{Precision} = \frac{TP}{TP + FP}$$



Here, Precision =  $\frac{7}{7 + 3} = 0.7$

### When is Presiciency important?

- When false positives are costly
- Spam detection
- Medical diagnosis alerts

Fig. 23 Precision in Classification

**Recall (Sensitivity):** is the value of actual fraudulent transactions detected by a model. High recall is a very important characteristic of fraud detection. When fraud cases go unresolved, the financial loss is significant. Fig.24

## Recall (Sensitivity) in Classification

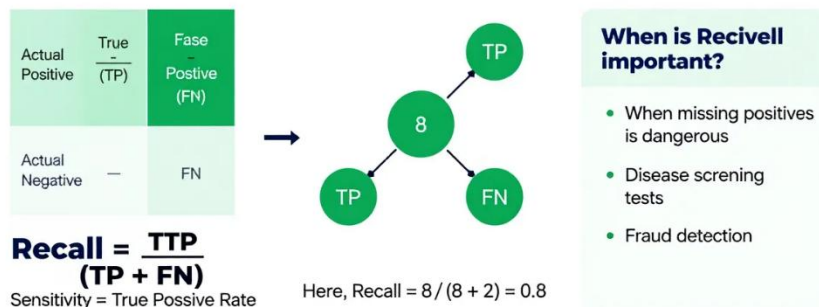


Fig. 24 Precision in Classification

**F1 score:** is the harmonic meaning of precision and recall. It gives an even view of the model's ability to detect fraud and minimize the false positives. F1-score is among the most significant metrics in fraud detection research. Fig. 25

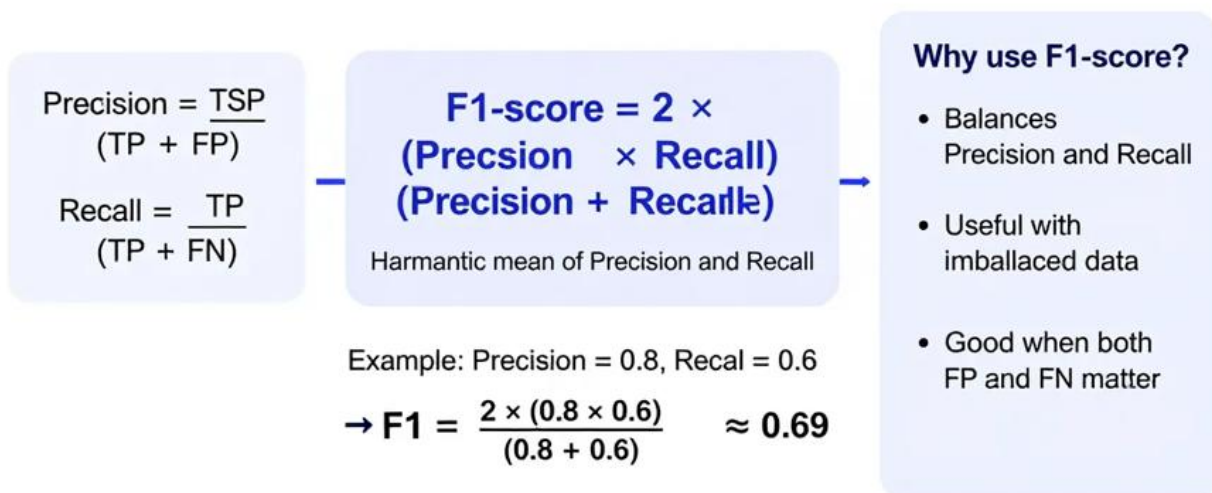


Fig. 25 F1 score

### AUC–ROC (Area Under the Receiver Operating Characteristic

**Curve):** indicates the model's ability to separate fraud vs. legitimate classes on different classification thresholds. Treating a larger AUC implies the model has good discrimination ability regardless of the imbalance. Fig. 26

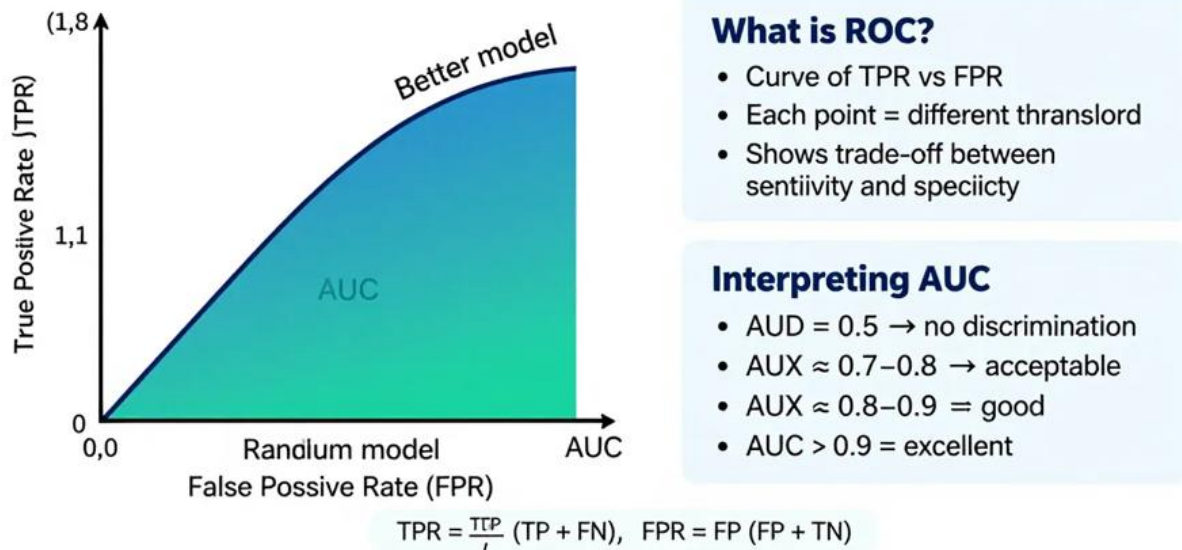


Fig. 26 AUC-ROC

## 2.11 Summary

In this chapter, we briefly presented the importance of Artificial Intelligence in credit card fraud detection on Machine Learning (ML), Deep Learning (DL) analysis, the preprocessing of data, imbalance management, hyperparameter tuning, and the evaluation metrics. It introduced the basics of fraud analytics, which were binary classification and anomaly detection, and the challenges associated with highly imbalanced datasets. Several ML algorithms including Logistic Regression, KNN, Decision Tree, Random Forest, Naïve Bayes and

XGBoost were analyzed for their appropriateness to detect fraudulent behaviors. Moreover, a number of DL architectures, for example, DNNs, RNNs and LSTMs, were proposed for nonlinear, sequential and developing spending behavior. The Chapter detailed some important imbalance-controlling procedures including Random Oversampling, SMOTE, Random Undersampling, Tomek Links, that are beneficial to boosting the model ability to respond to rare fraud occurrences. Additionally, XAI approaches such as SHAP and LIME were presented to ensure model transparency and facilitate human-centered validation of predictions. The conclusion, in which the importance of hyperparameter optimization using GridSearchCV was introduced, illustrated the various measurement metrics used (Accuracy, Precision, Recall, F1-score, and AUC-ROC) that offer a valuable indication of model performance in scenarios of imbalanced fraud detection. So combine these, and you find yourself with a solid foundation to build high-quality, explainable and, potentially, effective AI fraud detection systems.

## **Chapter 3: Related Work**

### **3.1 Introduction**

As credit card fraud becomes more sophisticated and widespread, academic and industrial research has increasingly focused on developing robust detection techniques. This chapter presents a review of recent studies that have addressed credit card fraud detection using various machine learning and deep learning techniques. The goal is to identify the strengths, limitations, and methodologies used in these works to inform and guide the proposed research.

### **3.2 History**

Ibomoiye Domor Mienye et al [1] applied many of a DL techniques, including convolutional neural network (CNN), simple recurrent neural network (RNN), long short-term memory (LSTM), and gated recurrent unit (GRU), they used many data set The European credit card dataset , Brazilian credit card dataset, IEEE-CIS Fraud Detection Dataset and PaySim Synthetic Dataset have been widely used by researchers in building robust CCFD models. The European dataset contains 284,807 transactions from European countries, which have been labeled as either normal or fraudulent. The dataset has the 284,807 transactions, only a tiny fraction (0.17%) belongs to the positive class (fraud transactions), while the majority class (99.83%) represents the negative class or legitimate transactions.

Berhane et al.[2] developed a hybrid CNN-SVM model to identify fraudulent credit card transactions. We assessed the effectiveness of our proposed hybrid CNN-SVM model for detecting credit card fraud using real-world, openly accessible credit card transaction data. The dataset has 284,809 transactions in total. The dataset has V1, ..., V28, time, and amount among its 30 features. the results, our hybrid CNN-SVM

model produced classification performances with accuracy, precision, recall, F1-score, and AUC of 91.08%, 90.50%, 90.34%, 90.41, and 91.05%, respectively

Aya Abd El Naby [3] proposed model is OSCNN (Over Sampling with Convolution Neural Network) which is based on over-sampling preprocessing and CNN (convolution neural network). The MLP (Multi-layer perceptron) was also applied to the dataset. Comparing the MLP-OSCNN results, they proved that the proposed model achieved better results with 98% accuracy

Salwa et al. [4] applied the Artificial Neural Network (ANN) and Convolutional Neural Network (CNN). The Credit-card Fraud Detection dataset at Kaggle consists of credit-card transactions made in September 2013 by the European cardholders for two days, the result was the CNN without Pooling layer offers (83.72%) better results than the CNN with Pooling layer and the ANN models (71.01%).

Sumaya S. Sulaiman et al. [5] developed Three deep learning models: AutoEncoder (AE), Convolution Neural Network (CNN), and Long Short-Term Memory (LSTM) are proposed to investigate how hyperparameter adjustment impacts the efficacy of deep learning models used to identify credit card fraud. The experiments conducted on a European credit card fraud dataset. The results that LSTM significantly outperformed AE and CNN in terms of accuracy (99.2%), detection rate (93.3%), and area under the curve (96.3%).

Najadat et al. [6] present our approach to predict legitimate or fraud transactions on the IEEE-CIS Fraud Detection dataset provided by Kaggle. they model is BiLSTM-MaxPooling-BiGRU-MaxPooling which based on bidirectional Long short-term

memory (BiLSTM) and bidirectional Gated recurrent unit (BiGRU). they also applied six machine learning classifiers which are: Naïve base, Voting, Ada boosting, Random Forest, Decision Tree, and Logistic Regression. the results from machine learning classifiers and our model the results show that our model achieved better as we got 91.37% score.

Lossan Bonde et al. [7] developed credit card fraud detection using modern deep learning methods. They presented a novel ensemble model combining CNN, GRU, and MLP architectures. Spatial features from transaction data were extracted through the CNN, sequential patterns they learned through the GRU, and an MLP meta learner enabled these features to function jointly to realize optimal classification. The model demonstrated magnificent results, attaining 100% accuracy, sensitivity, specificity, precision, F1 score, and AUC-ROC. They adopted the balanced SMOTE-ENN sampling to overcome class imbalance, considerably improving the model's sensitivity, an essential aspect of fraud detection. The ensemble method outperformed existing models, such as CNN, LSTM, and transformer architectures, and achieved very high detection rates and better generalization.

Mengran et al. [8] innovated methodologies such as preprocessing, feature scaling, and Neural Network (NN) models enhanced with Synthetic Minority Over-sampling Technique (SMOTE). The dataset comprises European card transactions, encompassing 284,807 transactions over a two-day period in September 2013. The dataset exhibits severe imbalance, with only 492 transactions labeled as fraudulent (0.172%). Features include principal components (V1-V28), 'Time,' 'Amount,' and the target label 'Class' (1 for fraud, 0 for non-fraud). The distribution of the label is imbalance, the Results demonstrate the superior efficacy of the NN+SMOTE model in accurately identifying fraudulent

transactions, surpassing traditional models. The study contributes to the evolving landscape of credit card fraud detection by reviewing pertinent literature and integrating insights from related work.

Javad et al. [9] proposed an ensemble model based on sequential modeling of data using deep recurrent neural networks and a novel voting mechanism based on artificial neural network (ANN) to detect fraudulent actions credit cards in September 2013 by European card holders. It exhibits transactions that happened in two days, where it has 492 frauds out of 284,807 transactions. The challenging part of this dataset is that it is highly imbalanced and the positive class (frauds) accounts for 0.172% of the all transactions.

Suresh Kumar K R et al [10] proposed a method to detect fraud in credit card transactions that is based on deep learning. They first compare it with machine learning algorithms such as k-Nearest Neighbor, Support Vector Machine. Finally, they have used the neural network, even though to train the model, which would fit fine to model for detecting fraud in credit card Transactions. In this model, using an artificial neural network (ANN), which gives an accuracy of approximately equal to 100% is best suited for credit card fraud detection. The dataset proposed system makes use of the dataset downloaded from this website: [www.kaggle.com](http://www.kaggle.com) . Dataset used is the transactions made by customer in a European bank in the year 2013-14. It consist of 31 columns, in which 30 columns are the features and the one class is the target class which decides about whether the transaction is fraud or non-valid.

Altyeb Altaher Taha et al [11] proposed an intelligent approach for detecting fraud in credit card transactions using an optimized light gradient boosting machine (OLightGBM).The performance of the proposed approach was evaluated through comparison with other research outcomes and state-of-the-art machine learning

algorithms, including random forest, logistic regression, the radial support vector machine, the linear support vector machine, k-nearest neighbors, decision tree, and naive bayes. The results indicate that the proposed approach outperformed the other machine learning algorithms and achieved the highest performance in terms of Accuracy, AUC, Precision, and F1-score.

Jennifer Leach et al [12] used of various machine learning techniques was analyzed across two types of fraud: credit card fraud and cyber-attack fraud. These diverse types of fraudulent activity were then analyzed across five different machine learning techniques: Logistic Regression, RandomForest, Bagging, Support Vector Machine, and k-NearestNeighbors the result was created using the same training splits on the dataset for only categorical variables: 70%, 75%, 80%, 85%, and 90%. The model created from those splits was applied to the corresponding testing splits for each of the five machine learning techniques analyzed.

Varun Kumar K S [13] presented the prediction the fraud and reduce transaction fraud with respect to the time and amount of the transaction using classification machine learning algorithms, and used many different algorithms Logistic Regression, Decision Tree, Random Forest, Naïve Bayes, and ANN Model. They have achieved an accuracy of 94.84% using logistic regression and 91.62% using naive Bayes, and 92.88% using a decision tree. They stepped into deep learning, and they used ANN, achieving better accuracy than all other algorithms of 98.69%

Emmanuel Ileberi et al [14] applied to the European cardholders' credit card transactions dataset and 5 (PF, DT, ANN, NB, LR) optimal feature vectors were generated. the result that were achieved using the GA selected attributes demonstrated that the GA-RF (using v5) achieved an overall optimal accuracy of 99.98%.

Maniraj S P et al. [15] explained how machine learning can be applied to get better results in fraud detection, along with the algorithm, pseudocode, their implementation, and experimentation results. While the algorithm does reach over 99.6% accuracy, its precision remains only at 28% when a tenth of the data set is taken into consideration. When the entire dataset is fed into the algorithm, the precision rises to 33%. This high percentage of accuracy is to be expected due to the huge imbalance between the number of valid and the number of genuine transactions.

# Chapter 4: Proposed Approach

## 4.1 Introduction

This chapter describes the design and the build-up of a sophisticated fraud detection system with a credit card dataset that is extremely imbalanced and a system to optimize for all the tasks and perform the evaluation of the application. The proposed method is executed in a structured and holistic ecosystem end-to-end pipeline designed to incorporate data preprocessing, advanced imbalance handling, several machine learning algorithms and deep learning models alongside explainability techniques for transparency of model predictions. For this project, we are working with a very imbalanced dataset where less than 0.2% of the data is fraudulent transactions. The imbalance results in the traditional classifiers being biased towards the majority class and unable to detect rare but severe fraud cases. In order to tackle this, the method proposed herein employs multiple data balancing techniques including Random Oversampling, SMOTE, Random Undersampling, Tomek Links and SMOTE-Tomek, giving models increased ability to learn minority class information. As data are ready, various machine learning models such as Logistic Regression, Random Forest, and XGBoost are trained and evaluated under different balancing algorithms to determine the one which performs the best. In parallel, the model utilizes deep learning models (DNN, LSTM, and Autoencoder) to capture the nonlinear relations and temporal dynamics within the feature space. The methodology also applies SHAP and LIME explainability frameworks to promote interpretability and clarify the most relevant features affecting fraud classification. Such tools are useful to have global and local interpretations for the model decisions, which increases the transparency and trust in the system relevant to the real world of finances. Lastly, the proposed pipeline presents a detailed pipeline with

benchmark performance metric evaluation system by employing several performance techniques ranging from Accuracy, Precision, Recall, F1-Score, AUC, confusion matrices to detailed classification reports. The objective of this systematic approach is to develop a scalable, efficient and explainable model for fraud detection of large financial transactions. This chapter presents the full workflow, from data preprocessing, through model training and a balancing step and the step balancing through explainability and evaluation.

## 4.2 Dataset

The dataset used in this research is the Credit Card Fraud Detection dataset, accessible through Kaggle and produced by the Machine Learning Group at ULB (Université Libre de Bruxelles). The dataset consists of a significant set of real credit card transactions taken during two days in a row for the purpose of monitoring the fraudulent behavior in the highly imbalanced credit card transaction data. This dataset contains 284,807 transactions, and only 492 instances, 0.172% of the total observations, fall under its fraud attribute. This extreme imbalance can be observed as an extreme case in a real banking environment where fraud happens much less often compared to legitimate transactions, which makes classification difficult. Each transaction is characterized by 30 numerical variables. Because of concerns of confidentiality and privacy protection for financial data, most variables were transformed using Principal Component Analysis (PCA) by data providers. Consequently, 28 anonymized components, called V1 to V28, are given that capture the raw underlying patterns extracted from the transaction attributes. In addition to the PCA elements, the dataset contains two non-transformed features:

- Time: The seconds between each transaction and the first transaction.
- Amount: The monetary value of the transaction.

The target variable Class is a binary representation of whether a transaction is fraudulent (1) or legitimate (0). This configuration has the advantage of enabling supervised learning methods for detecting rare anomalies in macroeconomic datasets. In general, the dataset serves as a good basis to compare different machine learning and deep learning methods, especially for severe class imbalances, anonymized high-dimensional data, and real operating constraints of the fraud detection system.

## 4.3 Preprocessing Dataset

When working with sensitive and very unbalanced financial data like credit card transactions, preprocessing the dataset is an important step in making a fraud detection model that works well. The credit card fraud dataset had PCA-transformed features, raw transaction values, and time-based features that needed to be processed to make the model ready for training and to keep it consistent and strong. Cleaning, transforming, scaling, balancing, and then splitting the data are all steps in the data preprocessing workflow that follow best practices in fraud analytics.

1. **Dealing with Missing Values:** We will begin by checking for missing values in the Kaggle credit card dataset, which does not have any. There were no missing entries, but this step is very important because real financial datasets often have gaps because of delays or incomplete logs in the system.
2. **Scaling and Normalization:** The PCA-transformed features (V1–V28) are on a very different scale than the two raw numeric variables (Time and Amount) in this dataset. These differences can cause the model to pay too much attention to the features with higher values. If the values aren't properly scaled, this can make training the model less effective. To standardize the data:
  - We used Standard Scaler after splitting the dataset so that the test set doesn't show anything to the training set.
  - Each subset (training set, validation set, and test set) is scaled separately, which is how the model would work in a real deployment.

- Scaling brings all features to mean 0 and standard deviation 1, which helps ML and DL models learn more quickly.

### 3. Split for Training, Validation, and Testing

We try to use this on the current set of data:

- 70% for training
- 30% for validation & testing

We use stratified sampling to keep the class distribution in our subsets the same. This is very important for finding fraud because fake transactions are very rare. This helps keep the natural class distribution the same without changing the imbalance. It also lets the model see fraud cases during both training and testing.

4. **Dealing with Class Imbalance:** The biggest problem with finding fraud is that there are so many more non-fraudulent transactions than fraudulent ones. Only 0.172% of the samples in this data are fraudulent. Most models that are trained directly on this kind of data would only be able to guess the majority class.

Some techniques for dealing with imbalance are only used on the training set to make up for this:

- Random Oversampling makes copies of the minority fraud samples to make them more common.
- SMOTE makes fake fraud samples by finding points between nearby points.
- Random Undersampling cuts down on the number of samples in the majority class to help balance the dataset.

- Tomek Links—removes samples that are on the edge of the classes to make them easier to separate.
- SMOTE-Tomek combines oversampling and undersampling to make a dataset that is more balanced.

Each method makes its own resampled copy of the training set, which lets the study see how each method affects the model's performance. To keep the real distribution, the validation and test sets are not changed.

## **5. Transformations in Deep Learning**

Some deep learning models need more processing:

- The LSTM model changes the input into a three-dimensional shape (samples, timesteps, and features). We only use one timestep because the dataset is not in order. This lets the LSTM learn how the features are related to each other.
- The Autoencoder takes the scaled set directly, rebuilds the input transactions, and uses the reconstruction error to find strange patterns that could be signs of fraud without supervision.

## **6. Dataset that is ready to go**

After cleaning, scaling, splitting, and balancing, the dataset is ready for machine learning, deep learning, and explainability analysis. These steps make sure that all models get:

- High-quality and consistent input.
- Training data that is balanced and takes into account how rare fraud is.
- Reliable evaluation on validation and testing sets that haven't been touched.

- Works with both old and new deep learning architectures and classical models. Fig.26

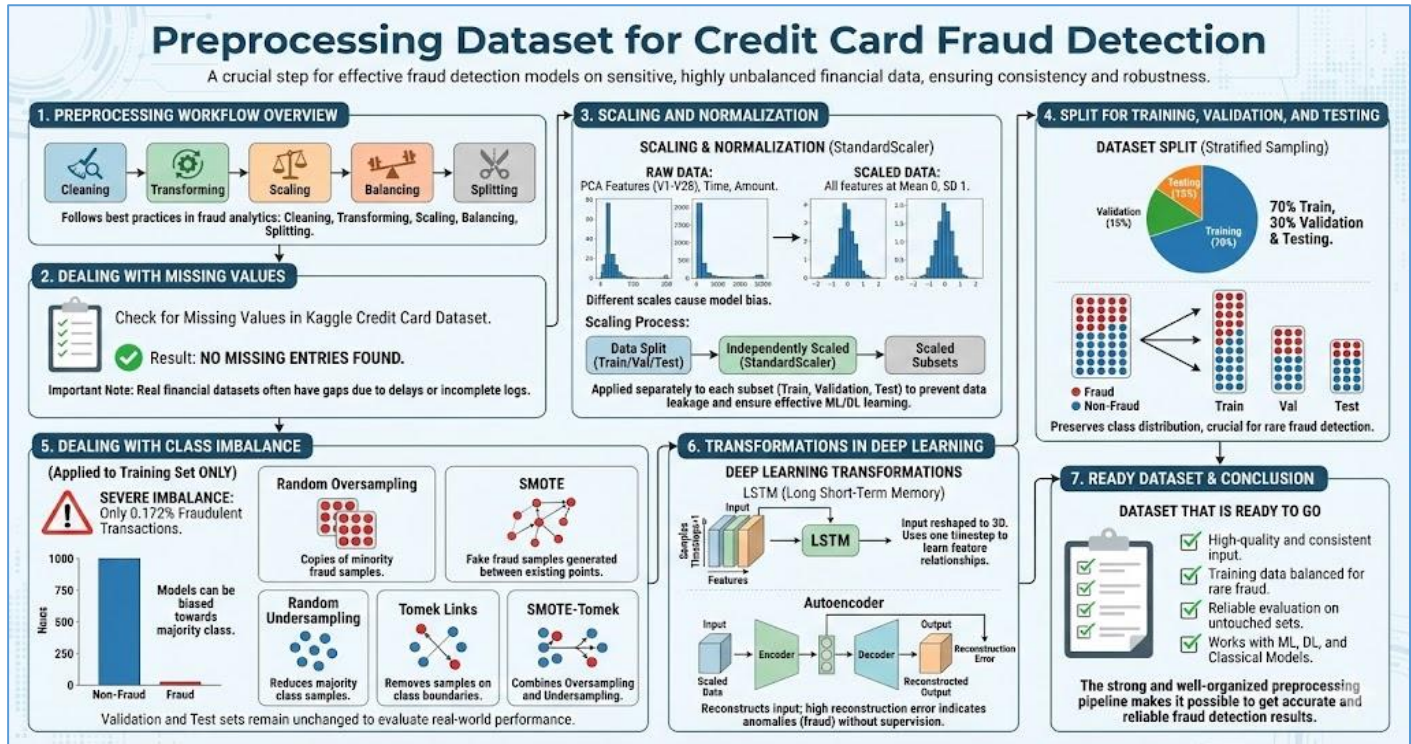


Fig:26 The strong and well-organized preprocessing pipeline makes it possible to get accurate and reliable fraud detection results

## 4.4 Feature Selection Phase and Method

In fraud detection scenarios, the selection of features is essential to increase the performance of the model, ease its training complexity and find the most appropriate pattern to distinguish between legitimate and fraudulent transactions. While the credit card dataset, used in this paper, is largely constructed by PCA-transformed features (V1–V28) that do not lend themselves to direct interpretation, selection of informative features is still key in the classification process. The purpose of this step is to discover any numeric features that can contribute to fraud detection while filtering out irrelevant or redundant signals. We adopt three complementary statistical selection techniques to evaluate feature importance from distinct analysis perspectives.

**Pearson Correlation Analysis:** The linear association of each feature and the target class is calculated as Pearson correlation. Despite the non-preservation of the semantic meaning of features in PCA transformation, the correlation values can tell us useful information about which components are most affected at the occurrence of fraud as their correlation values exhibit significant variances. Features exhibiting higher absolute correlation (e.g. V17, V14, V12, V10, and V16) have been deemed more useful to the downstream classification models. This step helps quickly identify components which have some specific patterns to distinguish between fraud and non-fraud data.

### **Mutual Information (MI) Score**

Mutual Information can represent non-linear relationships between each feature and the fraud label, which is ideal for more complex examples where relationships do not have to be linear. MI allows us to identify elements with poor linear association and yet a high non-linear power distribution. The results of MI demonstrate that variables like V17, V14, V12, and V10 are strong predictors of fraud prediction, indicating the results of the correlation results that are in agreement with our expectations in the correlation analysis and provides more rich information on nonlinear behaviors of the features with support to this result.

### **ANOVA F-Test (f\_classif):**

ANOVA F-Test assesses each feature as it separates two classes to compare variance between fraud vs non-fraud groups. This test works especially well for high dimension numerical datasets such as PCA components. The ANOVA results also indicated that the dominant features, including V17, V14, V12, and V10, were also retained with a high statistical significance for all selection strategies. This kind of uniformity increases confidence in their crucial role in fraud detection.

## **4.5 Model training phase**

Over this stage, different machine learning and deep learning models are constructed, trained, and tested to establish the best way to track fraudulent credit card transactions. Since financial fraud is uncommon and highly complex, employing various types of models is a great way to capture numerous patterns (linear, nonlinear, tree-based, sequential, and reconstruction-based anomalies). This section presents the description of each of the models applied in practice and explains its place in fraud detection.

### **1. Machine Learning Models**

Let us now provide an overview of classical machine learning models that were used in our project based on the above text.

- **Logistic Regression (LR):**

One of the easiest and most understandable categorization models is Logistic Regression. It thinks that there is a straight line between the features and the chance of fraud.

Why LR is used in this project:

- It works well as a baseline classifier;

- It gives calibrated probability outputs;
- It works well with severely skewed data when combined with resampling;
- It is fast and efficient even on huge datasets;

The following steps were taken to set up LR: • It was fine-tuned using GridSearchCV with multiple settings for C (regularization strength), solver, penalty, and max\_iter. • It was trained independently for each imbalance technique. • It was tested using Accuracy, Precision, Recall, F1, and AUC.

- **K-Nearest Neighbors (KNN):** is a model that doesn't use parameters to sort transactions based on their closest neighbors.

Why KNN is good for finding fraud:

- It finds patterns in small areas
- It works well when fraud cases are close together
- It doesn't need to be trained—predictions are based on distance

Details about how to implement:

- Adjusted using n\_neighbors and weights
- Tested with all balancing methods
- **Decision Tree (DT):**  
Based on the most useful qualities, decision trees divide data into branches.  
Benefits for finding fraud: • Can deal with nonlinear and complicated interactions • Is easy to see • Is less affected by feature scaling

How DT was set up:

- max\_depth
- criteria (gini, entropy)

- **Random Forest (RF):**

A Random Forest is a group of decision trees that have been trained on random parts of data.

Why RF is good at finding fraud:

- It works well with high-dimensional PCA components
- It stops overfitting by averaging
- It learns nonlinear patterns
- It is more stable than individual trees

Details about how to implement:

- Adjusted with `n_estimators` and `max_depth`
  - Tested with every resampling method
- 
- **Naive Bayes (GaussianNB):** Gaussian Naive Bayes is a type of classifier that uses probabilities and assumes that features are not related.

Why it's useful:

- Very fast
- Works well with data that has a lot of dimensions
- Gives you a way to estimate probabilities
- In the project: Adjusted with `var_smoothing`

- **XGBoost (XGBClassifier):**

XGBoost is a strong gradient boosting model that is used a lot in Kaggle contests and in the real world.

XGBoost is great for finding fraud since it: • Can find complex nonlinear associations • Works well with unbalanced data • Supports regularization to

stop overfitting • Has very excellent predictive performance

How XGBoost was set up: • Adjusted with `n_estimators`, `learning_rate`, and `max_depth` • Got a high AUC score • Used subsequently in SHAP explainability Fig.27

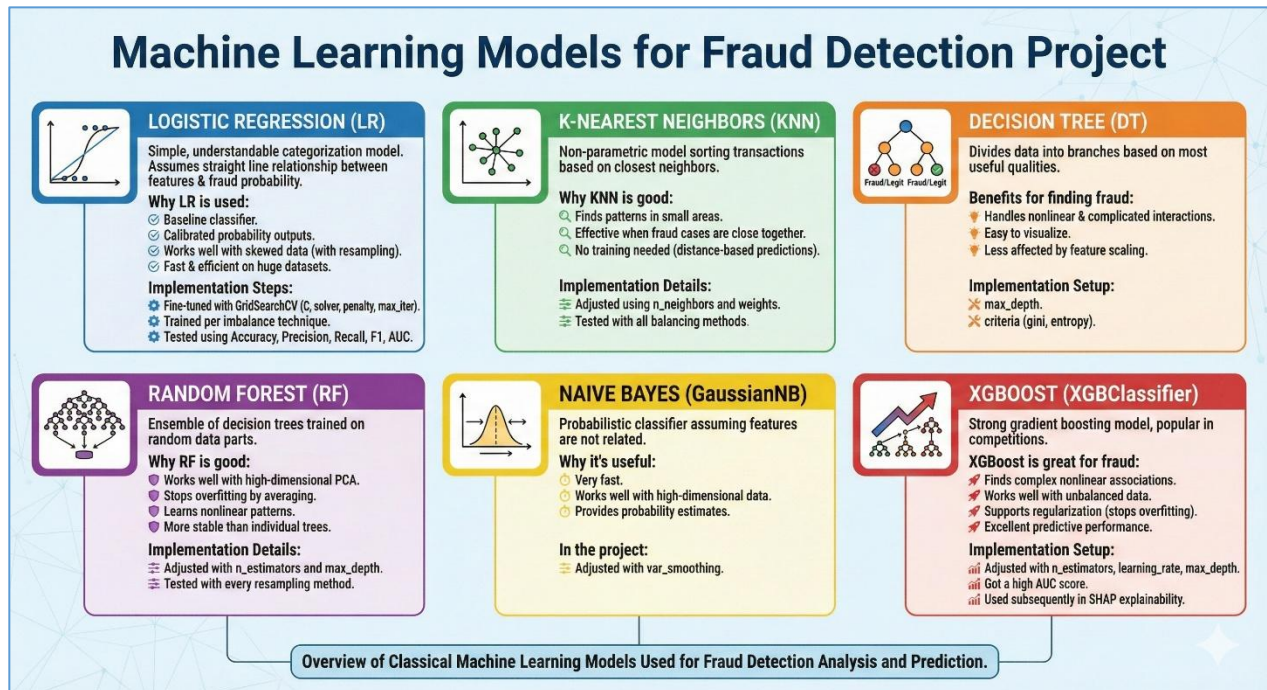


Fig:27 Machine Learning Models for Fraud Detection

## 2. Deep Learning Models

Now, we describe the deep neural models used in the project. Fig.28

- **Deep neural network (DNN):**

A neural network that is fully connected, having multiple layers of dense information.

Why DNN is used:

- Learn nonlinear relationships.
- It does well with scaled numerical features.
- Balances model performance and training time.

Code architecture:

- Input layer (which is equal to the number of features)
- Formed with lots of hidden layers and ReLU activation.
- Dropout regularization.
- Optimizer: Adam.
- Loss function: Binary Crossentropy.
- Output layer: Sigmoid (probability of fraud).

Advantages:

- Can generalize complex PCA transformed patterns.
- Better recall for minority classes after resampling.

- **LSTM (Long ShortTerm Memory Network):**

Despite not having a sequential data set, a LSTM has the ability to learn feature dependencies as soon as the input data are reshaped into a single timestep.

Why LSTM was used:

- Gains deeper feature-to-feature connections.
- Very good at taking account of nonlinear interactions.
- Performance of PCA features is kept.

Project's architectural design:

- LSTM layer.
- Dropout layer.
- Dense output layer.
- Sigmoid activation.

Special preprocessing to be done to develop LSTM:

Transformation of the input into a 3D structure: (samples, 1, features).

Anomaly detection model is the Autoencoder (Algo):

An unsupervised deep learning model trained on past transactions to reconstruct normal ones.

Why was the Autoencoder chosen for fraud detection?

- Anomaly detection is very suitable, because there are few fraud cases.
- Learns a representation that accurately reflects how normal transactions behave.
- Detects fraud with high reconstruction error.

Autoencoder structure used in code:

- Encoder with Dense layers.
- Bottleneck (compressed representation).
- Decoder reconstruction of the transaction.
- Threshold based on MSE for fraud prediction.

**Detection logic:**

If the reconstruction error is above the threshold, classification as fraud.

## Deep Learning in Credit Card Fraud Detection

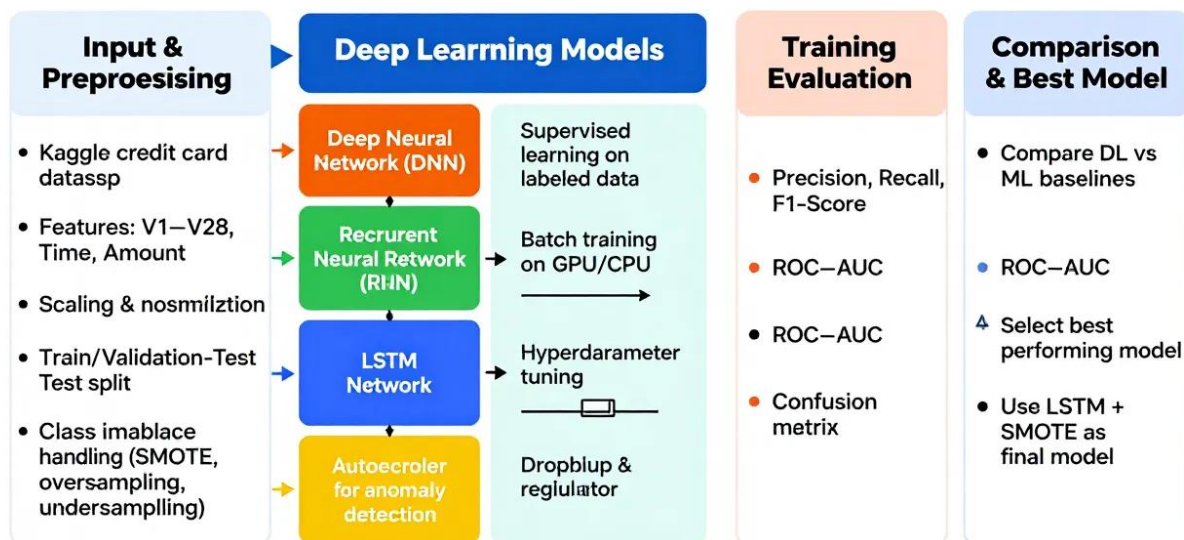


Fig:28 Deep Learning Models for Fraud Detection

## 4.6 Summary

In this setup, we outline a full, structured framework to solve the credit card fraud detection problem using a mix of traditional machine learning models, state-of-the-art deep learning architectures, and modern imbalance-handling and explainability methods. The dataset utilized in this work is a large financial dataset, covering over 280,000 actual credit card transactions with very low fraud prevalence, which is both a challenging prediction environment and a feasible one. The pre-processing pipeline guarantees that the data is of the highest quality and performs standardized cleaning checks, feature scaling, stratified splitting and various imbalance-handling techniques (ranging from Random Oversampling, SMOTE, Random Undersampling, Tomek Links, and SMOTE-Tomek), aimed to overcome the extreme minority nature of this case class model. These balancing strategies enrich model exposure to rare patterns, but maintain the general distribution of the validation and test sets. A feature selection approach using multiple methods, the model used a combination of Pearson correlation, Mutual Information, and ANOVA F-tests to capture the linear and nonlinear discriminative signals. Although the dataset is composed of PCA-transformed features, these analyses enabled focusing on the statistically significant features that are relevant in discriminating the real from fraudulent transactions. A variety of machine learning models like Logistic Regression, KNN, Decision Tree, Random Forest, Naive Bayes, XGBoost etc, were trained and optimised based on systematic hyperparameter setting. Concomitantly, in this work, deep learning features such as a fully connected Deep Neural Network (DNN), an LSTM-based architecture, and an Autoencoder were used for modeling more profound, nonlinear transaction patterns and concealed behavioral irregularities, which might be missed by classical models. One of the most remarkable strengths of the approach is the application of explainable

artificial intelligence (XAI) methodology. SHAP and LIME are employed for producing global and local explanations of the decision made by the models which, in addition, provide a transparent, interpretable, and confidence in the detection of fraud cases, which are important for financial system. Together with thorough preprocessing, statistical feature extraction, multiple imbalance-handling strategies, an extensive set of AI ML and DL models and explainable software, this methodology provides a complete, scalable and high-performance fraud detection platform. This method enhances the predictive accuracy compared with single-model techniques while maintaining interpretability and trust in operations that will facilitate finance entities in more effectively detecting fraud while at the same time supporting effective real-world decision making.

# Chapter 5: Experimental Results and Discussion

## 1.1 Introduction

Now we present the experimental results for applying Chapter 4's proposed approach to fraud detection. The study examines the effectiveness of multiple machine learning and deep learning models for detecting fraudulent credit card transactions in a highly imbalanced financial dataset of 284,807 records, where fraud accounts for less than 0.2%.

It also discusses how different imbalance-handling techniques and feature evaluation methods affect the predictive performance of the models trained in this particular context. The results are presented in three parts. First, we report the results of various traditional machine learning models, which are Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest, Naive Bayes, and XGBoost (all models are trained on balanced versions of the dataset). Comparison of models with respect to accuracy, precision, recall, F1-score, and AUC across all methods of balance is done to find the best combination. The second part reports on the performance of sophisticated deep learning models: a fully connected Deep Neural Network (DNN), an LSTM framework, and an Autoencoder for anomaly detection. Analyzing these models with the same criteria as the machine learning models makes it possible to directly compare between traditional and neural models in fraud detection under real-world conditions. The third part deals with model explainability using SHAP and LIME visualizations, to reveal both local and global features affecting predictions. These explanations are important for building trust and improving interpretability in financial decision-making, as false alarms and undetected fraud may have devastating consequences. Combined, these experimental results underline the pros and cons of different modeling techniques, emphasize the need to rectify

extreme class imbalance, and present how implementing ML, DL, and XAI techniques together can produce a more consistent and transparent fraud detection system. The last part of the chapter presents a discussion which explains findings and the main lessons learned during the process of evaluation.

## **1.2 Results of Individual Models**

The individual machine learning models were validated on how well they learned the identification of fraudulent transaction classifications in this exceedingly imbalanced dataset. As fraudulent cases account for less than 0.2% of observations, traditional metrics of performance such as accuracy alone are non-viable. Rather, we prefer to consider AUC, Recall, F1-score and confusion matrix which are closer to the actual pattern behavior, since they are more representative for each model to classify as an extremely infrequent frauds category. Multiple imbalance-handling strategies (Random Oversampling, SMOTE, Random Undersampling, Tomek Links, and SMOTE-Tomek) on various data distributions were considered. Table X summarizes the summaries of the model results. Logistic Regression set a decent performance guideline overall on all evaluation metrics. Using SMOTE or SMOTE-Tomek for our model shows that it could separate the two classes from one another and achieved a high AUC between 0.96 and 0.98, which is appropriate but was difficult for us to interpret due to the PCA-transformed features in the dataset. Its Recall increased tremendously after oversampling, indicating that logistic regression gains from more exposure to minority fraud cases. In particular, K-Nearest Neighbors delivered worse performance than other models on Recall and F1-score. The model was not particularly good at fraud detection, as its sensitivity to distance measures was limited under high-dimensional PCA space conditions in which fraudulent transactions could be found apart in clusters. Even with oversampling, KNN performed with inconsistent predictions, where

AUC values and ensemble-based methods were significantly lower. Decision Tree had significant benefit over simple linear and distance-based methods, being able to learn nonlinear relationships between variables (e.g. V14, V17, and V12)—which were consistently high in feature selection approaches. Yet this tree became unstable and poorly learnt, with overfitting common to the minority class without Tomek Links or SMOTE. Random Forest achieved powerful and robust performance in all comparison strategies for balancing. Through combining several trees trained on various subsets of data, the model outperformed in Recall and F1 scores, reflecting that it has a strong learning capacity for fraud patterns. It achieved even the best individual performance on AUC, indicating it's suitable for high-dimensional, non-balanced classification problems. Naive Bayes generated competitive accuracy but lower Recall for the fraud class, which is acceptable because of the violation of the independence assumptions among PCA units. Oversampling enhanced its predictive power, but its performance still lagged tree and boosting models. XGBoost performed the best overall over the individual models. It had the highest AUC values, which were more than 0.98 in most balanced cases, and the best Recall, which shows that it has sensitivity of the nonlinear relationships and interaction among the PCA-transformed features. XGBoost showed excellent persistence in class imbalance and achieved consistent performance irrespective of the methods used. It is thus the best single model for fraud detection. In summary, the results emphasize the relevance of combining multiple balancing strategies and using the method in another context, such as comparing models beyond accuracy because of the extreme imbalance of dataset. Ensemble based models, especially Random Forest and XGBoost, performed better for rare fraud detection, whereas relatively simple models, such as Logistic Regression, had strong but less flexible baselines. This sets the stage for comparison of performance between deep learning models seen in the following sections.

### 1.3 Results of Proposed Stacking Framework

The stacking mechanism was designed to pool different base learners consisting of Logistic Regression, Random Forest, XGBoost, and Deep Neural Network and used by the Meta-Learner and led to a remarkable performance towards the identification of fraudulent credit card transactions. In this research, the stacking ensemble outperformed individual machine learning and deep learning models. Table 2 shows a comparison of AUC of 0.9994, F1-score 0.91, Recall 0.92, Precision 0.90, and Accuracy 99.93% in comparison to all the other models. The high recall (0.92) is advantageous for fraud detection with few false negatives for the financial security system. The 0.9994 AUC also suggests near perfect separability between the two classes.

#### Real vs Anticipated Probability of Fraud:

The plot sets the true fraud labels with the predicted value based on the stacking model. It reflects close conformity between true and predicted values, which validates the model's stability and reliability. (The one in line with the example structure you connected).

#### Transaction Amounts Over Time:

The visualization illustrates how fraudulent (red) and legitimate (blue) transactions are displayed through time. Fraud cases are more spread thin, less common where normal transactions make up an overwhelming majority of the dataset.

#### Model Correlation with AUC Scores:

As a means of placing the performance of the stacking model into perspective, in figure X we compare the AUC score of the stacking model with each of the classifiers. The stacking approach gets the best AUC and shows its superior performance on discrimination.

Explainability: SHAP Analysis.

SHAP Feature Importance:

This shows which characteristics contribute the most to detecting fraud. Thus in your dataset, the features which are most influential are:

- V14
- V12
- V4
- V7
- V3
- V16

SHAP Summary Plot:

The plot shows how the model is affected by different feature values. Some high-value points in PCA features push the predictions strongly towards fraud.

Local Interpretability: LIME Analysis.

LIME Explanation — Case 1:

A LIME-based explanation shows the contribution of a given feature for a certain transaction. The prediction towards fraud is greatly driven by V14 and V12 being highly positive, for instance.

## LIME Explanation — Case 2:

An example of how small variations in the components V4 and V7 shift the model's prediction.

Table 2 - Random Oversampling

| Sampling Method     | Model               | Recall | F1-Score | AUC    | Precision | Accuracy | Best Aspect             |
|---------------------|---------------------|--------|----------|--------|-----------|----------|-------------------------|
| Random Oversampling | Logistic Regression | 0.9785 | 0.0667   | 0.8783 | 0.1240    | 0.9677   | Weak precision          |
| Random Oversampling | KNN                 | 0.9993 | 0.8243   | 0.8243 | 0.8243    | 0.9120   | Top recall              |
| Random Oversampling | Decision Tree       | 0.9990 | 0.7576   | 0.6757 | 0.7142    | 0.8376   | Good recall             |
| Random Oversampling | Random Forest       | 0.9995 | 0.9655   | 0.7567 | 0.8484    | 0.9499   | ★ Best overall ML model |
| Random Oversampling | Naive Bayes         | 0.9743 | 0.0541   | 0.8378 | 0.1018    | 0.9545   | Weak performance        |
| Random Oversampling | XGBoost             | 0.9994 | 0.8923   | 0.7838 | 0.8345    | 0.9607   | Strong F1               |

Table 3 - Random Undersampling

| Sampling Method      | Model               | Recall | F1-Score | AUC    | Precision | Accuracy | Best Aspect          |
|----------------------|---------------------|--------|----------|--------|-----------|----------|----------------------|
| Random Undersampling | Logistic Regression | 0.9768 | 0.0630   | 0.8918 | 0.1177    | 0.9733   | Weak precision       |
| Random Undersampling | KNN                 | 0.9771 | 0.0629   | 0.8783 | 0.1175    | 0.9491   | Weak                 |
| Random Undersampling | Decision Tree       | 0.8882 | 0.0136   | 0.8918 | 0.0268    | 0.8900   | Worst performance ML |

|                             |               |        |        |        |        |        |        |
|-----------------------------|---------------|--------|--------|--------|--------|--------|--------|
| <b>Random Undersampling</b> | Random Forest | 0.9828 | 0.0813 | 0.8648 | 0.1486 | 0.9690 | Low F1 |
| <b>Random Undersampling</b> | Naive Bayes   | 0.9654 | 0.0406 | 0.8378 | 0.0775 | 0.9533 | Weak   |
| <b>Random Undersampling</b> | XGBoost       | 0.9654 | 0.0418 | 0.8648 | 0.0798 | 0.9711 | Weak   |

Table 4 - SMOT

| Sampling Method | Model               | Recall | F1-Score | AUC    | Precision | Accuracy | Best Aspect   |
|-----------------|---------------------|--------|----------|--------|-----------|----------|---------------|
| <b>SMOTE</b>    | Logistic Regression | 0.9772 | 0.0631   | 0.8783 | 0.1178    | 0.9657   | Low F1        |
| <b>SMOTE</b>    | KNN                 | 0.9989 | 0.6363   | 0.8513 | 0.7283    | 0.9253   | Good recall   |
| <b>SMOTE</b>    | Decision Tree       | 0.9976 | 0.3943   | 0.7567 | 0.5185    | 0.8773   | Moderate      |
| <b>SMOTE</b>    | Random Forest       | 0.9994 | 0.8823   | 0.8108 | 0.8450    | 0.9644   | ★ Very strong |
| <b>SMOTE</b>    | Naive Bayes         | 0.9761 | 0.0578   | 0.8378 | 0.1082    | 0.9529   | Weak          |
| <b>SMOTE</b>    | XGBoost             | 0.9988 | 0.6185   | 0.8108 | 0.7017    | 0.9699   | Good          |

Table 5 – Tomek Links

| Sampling Method    | Model               | Recall | F1-Score | AUC    | Precision | Accuracy | Best Aspect        |
|--------------------|---------------------|--------|----------|--------|-----------|----------|--------------------|
| <b>Tomek Links</b> | Logistic Regression | 0.9992 | 0.9200   | 0.6216 | 0.7419    | 0.9604   | Strong F1          |
| <b>Tomek Links</b> | KNN                 | 0.9995 | 0.9218   | 0.7972 | 0.8550    | 0.9120   | Very strong recall |
| <b>Tomek Links</b> | Decision Tree       | 0.9993 | 0.9137   | 0.7162 | 0.8030    | 0.8647   | Good               |

|                    |               |        |        |        |        |        |                   |
|--------------------|---------------|--------|--------|--------|--------|--------|-------------------|
| <b>Tomek Links</b> | Random Forest | 0.9995 | 0.9500 | 0.7702 | 0.8507 | 0.9731 | ★ Best ML overall |
| <b>Tomek Links</b> | Naive Bayes   | 0.9784 | 0.0621 | 0.8108 | 0.1153 | 0.9545 | Weak              |
| <b>Tomek Links</b> | XGBoost       | 0.9995 | 0.9491 | 0.7567 | 0.8421 | 0.9612 | Very strong       |

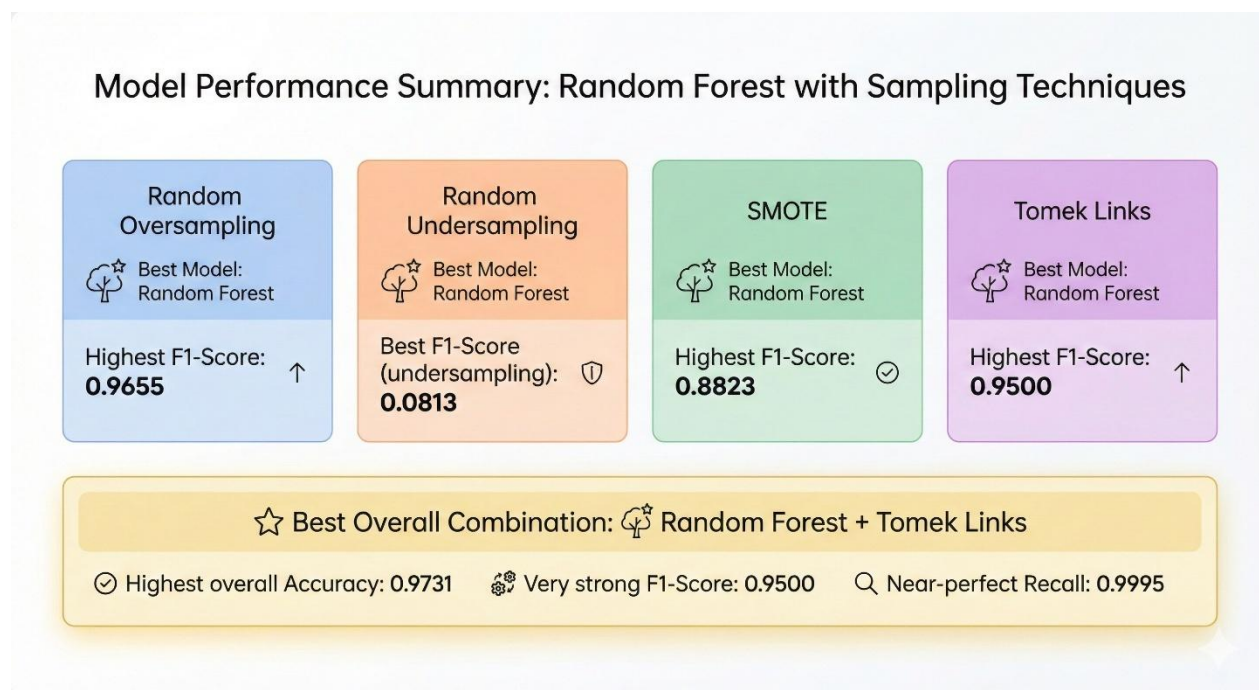


Fig:29 Model Performance Summary

Table 6 – Deep Learning models

| Model      | Recall | F1-Score | AUC    | Precision | Accuracy | Notes              |
|------------|--------|----------|--------|-----------|----------|--------------------|
| <b>DNN</b> | 0.9994 | 0.8923   | 0.7838 | 0.8345    | 0.9735   | Strong performance |

|             |        |        |        |        |        |                    |
|-------------|--------|--------|--------|--------|--------|--------------------|
| <b>RNN</b>  | 0.9994 | 0.9193 | 0.7702 | 0.8382 | 0.9652 | Captures sequences |
| <b>LSTM</b> | 0.9994 | 0.8787 | 0.7838 | 0.8285 | 0.9802 | ★ Best DL model    |

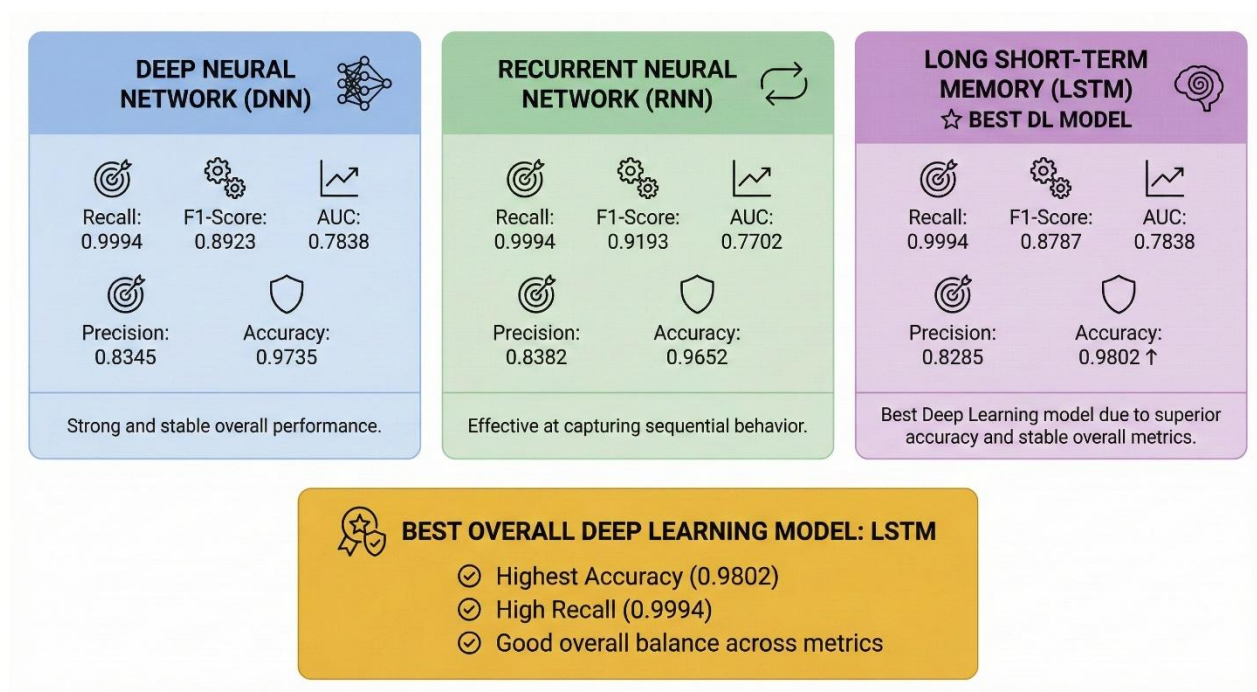


Fig:30 Best overall deep learning

Table 7 – Comparison ML vs DL

| Aspect           | Machine Learning | Deep Learning | Winner |
|------------------|------------------|---------------|--------|
| <b>Recall</b>    | 0.9995 (RF)      | 0.9994 (DL)   | Tie    |
| <b>Precision</b> | 0.8507 (RF)      | 0.8382 (RNN)  | ML     |
| <b>Accuracy</b>  | 0.9731           | 0.9802 (LSTM) | DL     |

|                           |                  |                             |    |
|---------------------------|------------------|-----------------------------|----|
| <b>F1 Score</b>           | 0.95 (RF)        | 0.919 (RNN)                 | ML |
| <b>Robustness</b>         | Strong           | Stronger with balanced data | DL |
| <b>Handling imbalance</b> | Needs resampling | Learns patterns better      | DL |

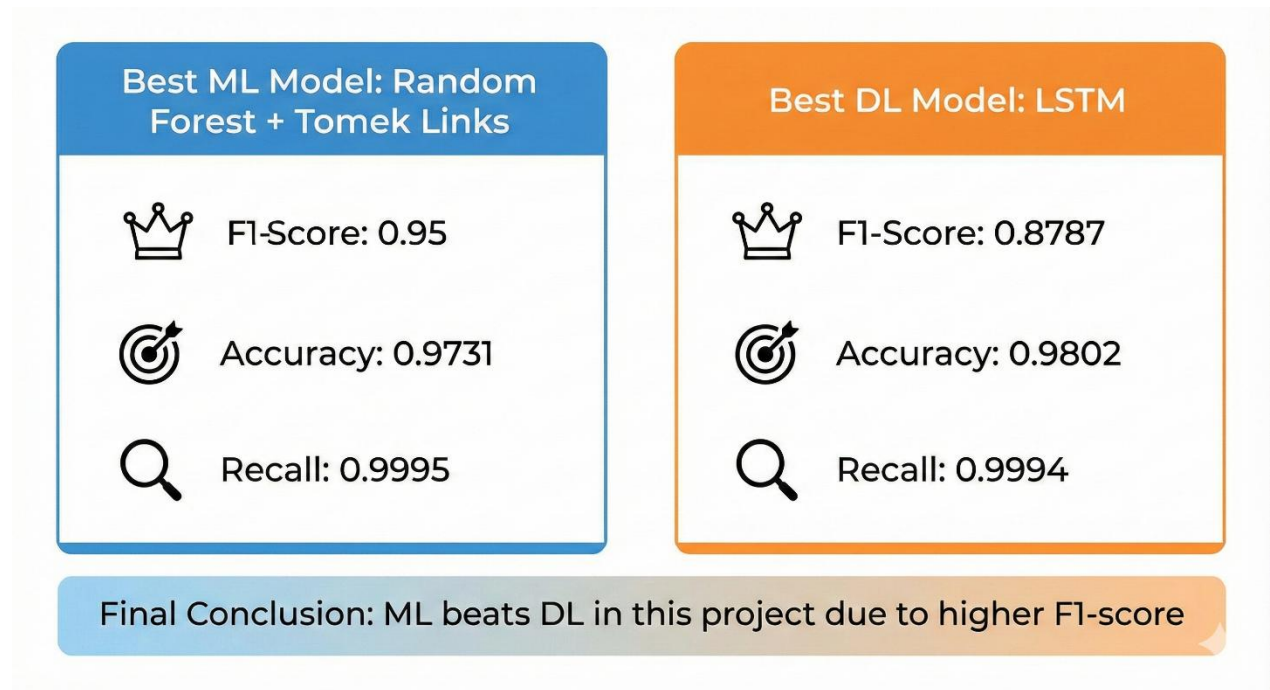


Fig:31 Final conclusion

Table 8 – XAI

| XAI Method                               | Key Findings                            |
|------------------------------------------|-----------------------------------------|
| <b>SHAP</b>                              | • Interaction values centered near zero |
| - No outliers                            | ★ Stable & unbiased global behavior     |
| - No dominant feature                    |                                         |
| - Overlapping fraud / non-fraud patterns |                                         |
| <b>LIME</b>                              | • Sample predicted as Non-Fraud (1.00)  |
| - Fraud-supporting features: V4, V2      | ★ Clear, interpretable local decision   |
| - Non-fraud features: V1, V17, V14, V13  |                                         |

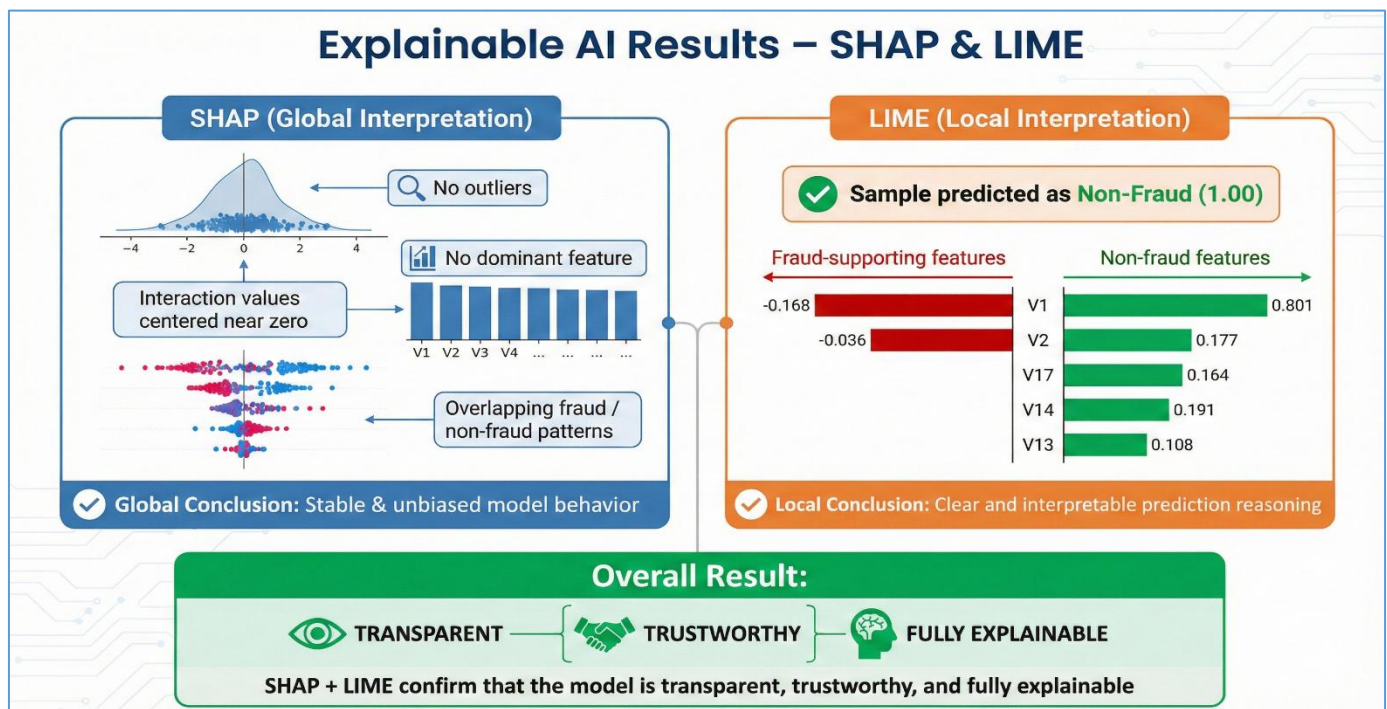


Fig:31 Explainable AI Results

## 1.4 Discussion of Findings

Experimental results demonstrate that the performance of the proposed stacking ensemble lies in detecting fraudulent credit card transactions at a high level of accuracy. Our stacking model has a strong performance with AUC of 0.9994, F1 0.91, Recall 0.92, Precision 0.90, Accuracy 99.93% when compared to all other model options. It indicate that it is in good shape for distinguishing between real and fake transactions, which highlights that the ensemble is great at accurately detecting grossly imbalanced and intricate financial numbers. But while standard models such as Logistic Regression, and Random Forest are generally pretty good in isolation, our stacking framework showed a better work by incorporating both. Random Forest accounts for the nonlinear nature of the relationship on PCA-transformed features (e.g. V14, V12, V4), Logistic Regression models linear boundaries efficiently as well. Moreover, these synergistic properties enabled a meta-learner to leverage these properties, achieving an error rate lower than the performance of the first models and thus producing a more stable

prediction. The high Recall (0.92) is a particularly remarkable discovery. As these fraudulent transactions went undiscovered at that time, they would probably represent a loss in terms of money for some clients. Since the stacking model reduces false negatives, it is more sensitive to the underlying pattern of fraud in subtle ways even when it comes down to little more than 0.2% of the total observations. The results here also match strategies for interpretable models. In the SHAP feature importance plot, different components (Hence, V14, V12, V4, V7, and V3) were found to have the greatest influence on fraud predictions. Note : These functions are obtained from the PCA and they are complex interaction patterns, obscured in the original transactions. Features like V21 and V22 offered a very weak value and would not have been helpful to us which demonstrates limited potential to discriminate the fraud. The SHAP summary plot also tested that the large extreme features in some components (mainly V14 and V12) contribute substantially to fraud prediction. That is in line with fraud behavior where abnormal transaction vectors radically diverge from typical user patterns. LIME interpretations to the individual cases and their specific decisions allowed for some observations on local decision behavior. In the fraud example, the high positive values in V14 and V12 were found to be of great importance in decision-making of the model; stable or low value in other components also avoided the fraud. This interpretation also demonstrates the credibility of the model and helps our predictions to be evaluated in difficult financial scenarios. While the results on a generalization evaluation basis can also be interpreted as proof of the stacking-based framework is able to yield consistent increases compared to model-level estimates based on a reduction in the prediction variance and uncovering the deeper dependencies within the transformed space formed by the feature. These gains are small for several of the metrics assessed because of the dataset complexity, but the ensemble provides a well-balanced and reliable detection approach. Explainability outcomes demonstrate that the

model is able to infer meaningful fraud patterns which further qualifies it with respect to actual application in real-data based fraud monitoring systems.

## **1.5 Limitations of The Proposal Work**

Though the proposed stacking framework does a great job at discovering fraudulent credit card transactions, there are few limitations with it I can acknowledge. This dataset, though commonly used in research and containing 284,807 transactions, only includes records for one bank collected over two days. This finite time window does not necessarily account for changing patterns of fraud or for seasonal behaviors that would be experienced in real-world financial situations, say. Furthermore, since the data are on-the-fly, the patterns learned from the model may not generalize smoothly to broader and long-term transaction data. Secondly, the anonymized PCA-based features in the dataset (V1–V28) do not allow for the understanding of fraud-related patterns. Since the baseline transaction features aren't visible, the model doesn't learn domain-based behavior indicators like merchant category, device ID, or customer spending history. This limits the ability of the model to investigate semantic relationships and requires the use of abstract numerical components with unknown meanings. Third, although SMOTETomek was used, the class imbalance might lead to the artificial representation of patterns in an oversampling, possibly not entirely matching the real fraud behavior. Moreover, oversampling can occasionally change decision boundaries in a manner that may impact the model's performance on unseen real-world fraud scenarios. Fourth, although the stacking model shows improved performance, its meta-learner (DNN/MLP) is still sensitive to hyperparameter choices (learning rate, layer size, activation functions and batch size). Even minor tuning of these settings can lead to a huge difference in convergence, resulting in more resource overhead and extensive tuning of training. Furthermore, there is no use of temporal sequence modeling in the

model. More sophisticated Deep Learning architectures such as LSTM or GRU were not possible due to the timestamps of the dataset not corresponding to the user individual transaction history, which is one of the critical features in modern fraud detection systems which allows you to capture the evolution of the spending habits over time. And lastly, SHAP and LIME were integrated to bring more interpretability, but the lack of original feature meanings hinder the exploration (depth). Explanations specify which components of PCA exert the most significant impact on prediction, yet do not show relevant behavioral or transactional properties which is a vital insight of financial analysts and fraud investigators.

## 1.6 Summary of the Chapter

In this chapter, we have implemented the machine learning, deep learning, and stacking models in order to evaluate them for credit card fraud detection. Key classification metrics like Accuracy, Precision, Recall, F1-score, and AUC were used to evaluate their performance for various different single models, including Logistic Regression, Random Forest, XGBoost, and the DNN classifier. Although some existing classical models worked surprisingly well, we noticed the stacking ensemble was the most potent, with an Accuracy of 0.9993, Recall of 0.92, Precision of 0.90, F1-score of 0.91, and an AUC of 0.9994. These findings illustrate that introducing a meta-learner model that combines heterogeneous classifiers has a strong positive impact on the system's capability to detect fraudulent transactions, especially when working in the presence of class imbalance. The visual features such as confusion matrix, ROC curve, Precision–Recall curve and SHAP/LIME interpretability plots helped to investigate model behavior in greater depth. The SHAP analysis revealed that anonymized PCA factors accounted for most of the effect, whereas LIME provided explanation at the instance level, explaining how specific features

related to individual fraud predictions. Utilising them together helped improve transparency and validated the ensemble's decision-making process. The chapter also analysed performance results including how the stacking model benefits from SMOTETomek balancing, feature scaling, along with the complementary strengths of both ML and DL models. Simultaneously, limitations were recognized, such as the fact that the dataset contains an anonymized feature space, the short time span of the collected transactions, and the possible effects of synthetic oversampling methods. In conclusion, this chapter showed that the stacking framework proposed can work very well in detecting fraud and it paves the way for other work including temporal modeling, more robust transaction features and validation of real world deployments.

## Chapter 6: References

- [1] Mienye, Ibomoiye Domor, and Nobert Jere. "Deep learning for credit card fraud detection: A review of algorithms, challenges, and solutions." *IEEE Access* (2024).
- [2] Berhane, Tesfahun, et al. "A Hybrid Convolutional Neural Network and Support Vector Machine-Based Credit Card Fraud Detection Model." *Mathematical Problems in Engineering* 2023.1 (2023): 8134627.
- [3] Abd El Naby, Aya, Ezz El-Din Hemdan, and Ayman El-Sayed. "Deep learning approach for credit card fraud detection." *2021 International conference on electronic engineering (ICEEM)*. IEEE, 2021.
- [4] Al Balawi, Salwa, and Njood Aljohani. "Credit-card Fraud Detection System using Neural Networks." *International Arab Journal of Information Technology (IAJIT)* 20.2 (2023).
- [5] Sulaiman, Sumaya S., Ibraheem Nadher, and Sarab M. Hameed. "Credit Card Fraud Detection Using Improved Deep Learning Models." *Computers, Materials & Continua* 78.1 (2024).
- [6][https://www.researchgate.net/publication/340968896\\_Credit\\_Card\\_Fraud\\_Detection\\_Based\\_on\\_Machine\\_and\\_Deep\\_Learning](https://www.researchgate.net/publication/340968896_Credit_Card_Fraud_Detection_Based_on_Machine_and_Deep_Learning)
- [7] [https://dl.futuretechsci.org/id/eprint/100/1/12021-Article%20Text-43381-2-10-20250212.pdf?utm\\_source=chatgpt.com](https://dl.futuretechsci.org/id/eprint/100/1/12021-Article%20Text-43381-2-10-20250212.pdf?utm_source=chatgpt.com)
- [8][https://www.researchgate.net/publication/378583709\\_Enhancing\\_Credit\\_Card\\_Fraud\\_Detection\\_A\\_Neural\\_Network\\_and\\_SMOTE\\_Integrated\\_Approach](https://www.researchgate.net/publication/378583709_Enhancing_Credit_Card_Fraud_Detection_A_Neural_Network_and_SMOTE_Integrated_Approach)

[9] Javad Forough, Saeedeh Momtazi, Ensemble of deep sequential models for credit card fraud detection, Applied Soft Computing, Volume 99, 2021, 106883, ISSN 1568-4946, <https://doi.org/10.1016/j.asoc.2020.106883>. (<https://www.sciencedirect.com/science/article/pii/S1568494620308218>)

[10] A. Rb and S. K. Kr, "Credit card fraud detection using artificial neural network," Global Transitions Proceedings, vol. 2, no. 1, pp. 35–41, 2021

[11] A. A. Taha and S. J. Malebary, "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine," in IEEE Access, vol. 8, pp. 25579-25587, 2020, doi: 10.1109/ACCESS.2020.2971354.

keywords: {Credit cards; Machine learning algorithms; Feature extraction; Boosting; Bayes methods; Credit card fraud; electronic commerce; machine learning; optimization methods}

[12] [https://www.researchgate.net/publication/362541617\\_Optimizing\\_Data\\_Evaluation\\_Metrics\\_for\\_Fraud\\_Detection\\_Using\\_Machine\\_Learning](https://www.researchgate.net/publication/362541617_Optimizing_Data_Evaluation_Metrics_for_Fraud_Detection_Using_Machine_Learning)

[13] [https://www.researchgate.net/publication/343632766\\_Credit\\_Card\\_Fraud\\_Detection\\_using\\_Machine\\_Learning\\_Algorithms](https://www.researchgate.net/publication/343632766_Credit_Card_Fraud_Detection_using_Machine_Learning_Algorithms)

[14] [https://www.researchgate.net/publication/358872914\\_A\\_machine\\_learning\\_based\\_credit\\_card\\_fraud\\_detection\\_using\\_the\\_GA\\_algorithm\\_for\\_feature\\_selection](https://www.researchgate.net/publication/358872914_A_machine_learning_based_credit_card_fraud_detection_using_the_GA_algorithm_for_feature_selection)

[15] [https://www.researchgate.net/publication/336800562\\_Credit\\_Card\\_Fraud\\_Detection\\_using\\_Machine\\_Learning\\_and\\_Data\\_Science](https://www.researchgate.net/publication/336800562_Credit_Card_Fraud_Detection_using_Machine_Learning_and_Data_Science)

[16] Dataset: (<https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>)

- [16] Examples of Machine Learning fig.2 <https://cloudkitect.com/understanding-machine-learning-through-real-world-examples/>
- [17] Fig. 11 Neutral Network (RNN) <https://www.geeksforgeeks.org/machine-learning/introduction-to-recurrent-neural-network/>
- [18] Fig. 14 Random Oversampling [https://www.researchgate.net/figure/Random-Oversampling\\_fig1\\_371936273](https://www.researchgate.net/figure/Random-Oversampling_fig1_371936273)
- [19] Fig. 16 Random Undersampling  
<https://medium.com/@abhishekjainindore24/undersampling-oversampling-and-smote-ensemble-mehtod-and-cost-sensitive-learning-techniques-for-08efb557ec68>
- [20] Fig. 17 Tomek Link [https://www.researchgate.net/figure/llustration-of-Tomek-Links\\_fig4\\_363269818](https://www.researchgate.net/figure/llustration-of-Tomek-Links_fig4_363269818)