

**National & Kapodistrian University of Athens**

**Department of Economics**

Master of Science in

Business Administration, Analytics and Information Systems

**Master Thesis**

**Fraud Detection On Transactional Data with Machine Learning**

**Koutsompinas Konstantinos, Computer Scientist**

**Κωνσταντίνος Κουτσομπίνας**

Supervisory teacher,

Argyriou Athanasios

Athens 2025-2026

Table of Contents

[2. Literature Review: Machine Learning for Fraud Detection in Online Financial Transactions 1](#_Toc212489008)

[2.1. Introduction and Overview of the Domain 1](#_Toc212489009)

[2.2. Machine Learning Approaches to Fraud Detection 3](#_Toc212489010)

[2.2.1. Supervised Learning (Classification) 3](#_Toc212489011)

[2.2.2. Unsupervised and Semi-Supervised Learning (Anomaly Detection) 3](#_Toc212489012)

[2.2.3. Ensemble Methods 4](#_Toc212489013)

[2.2.4. Deep Learning 5](#_Toc212489014)

[2.2.5. Reinforcement Learning 6](#_Toc212489015)

[2.2.6. Comparison of Approaches 6](#_Toc212489016)

[2.3. Datasets and Benchmarks 8](#_Toc212489017)

[2.4. Feature Engineering for Fraud Detection 10](#_Toc212489018)

[2.5. Model Evaluation Metrics 12](#_Toc212489019)

[2.6. Challenges and Limitations in Machine Learning-Based Fraud Detection 14](#_Toc212489020)

[2.7. Tools, Libraries, and Frameworks 17](#_Toc212489021)

[2.8. Research Gaps and Future Directions 17](#_Toc212489022)

[2.9. Conclusion 19](#_Toc212489023)

[3. Data 21](#_Toc212489024)

[3.1. Overall Description 21](#_Toc212489025)

[3.1.1. Overview 21](#_Toc212489026)

[3.1.2. Transaction Table 21](#_Toc212489027)

[3.1.3. Identity Table 22](#_Toc212489028)

[3.2. Target Variable Analysis 23](#_Toc212489029)

[3.3. Missing Values 24](#_Toc212489030)

[3.4. Transaction DT (Time) 25](#_Toc212489031)

[3.5. Transaction Amount 27](#_Toc212489032)

[3.6. Transaction Categorical Features 29](#_Toc212489033)

[3.7. Count and Deltatime features 34](#_Toc212489034)

[3.8. Vesta Features 36](#_Toc212489035)

[6. References 38](#_Toc212489036)

# 2. Literature Review: Machine Learning for Fraud Detection in Online Financial Transactions

2.1. Introduction and Overview of the Domain

Financial fraud in online transactions is a critical problem with far-reaching economic and societal impacts. Organizations worldwide face significant losses and erosion of trust due to fraudulent activities. For instance, global payment fraud losses climbed from about $9.8 billion in 2011 to $32.4 billion in 2020 and are projected to exceed $40 billion by 2027 (Ramachandran et al., 2023). The Association of Certified Fraud Examiners (ACFE) estimates that fraud costs roughly 5% of annual revenues globally (around $4 trillion) (Barnes, 2020). Surveys indicate that over half of companies worldwide have experienced some form of fraud in recent years (56% in a 2022 PwC survey), with even higher exposure in regions such as the Americas (KPMG, 2022). Likewise, cyber-fraud threats are escalating – in one KPMG survey, 83% of executives reported cyber-attacks in 12 months, and 71% encountered internal or external fraud attempts (KPMG, 2022). These trends underscore the urgency of effective fraud detection in financial services.

Fraudulent activities have been a critical problem in recent years, especially with the trend of digitalization. The rise of electronic financial transactions has resulted in a corresponding rise in fraudulent electronic activities. For instance, global payment fraud losses climbed from about $9.8 billion in 2011 to $32.4 billion in 2020 and are projected to exceed $40 billion by 2027 (Ramachandran et al., 2023). This indicates the urgency of effective fraud detection mechanisms.

Traditionally, fraud detection relied on manual audits and **rule-based systems** (e.g., static if-then rules), but these approaches have become inadequate against the **volume and complexity** of modern online transactions (Aljunaid *et al.*, 2025). Fraudsters continually adapt their tactics, often camouflaging illicit transactions among normal activity (Aljunaid et al., 2025). Rule-based systems with fixed thresholds struggle to keep pace with emerging fraud patterns and the scale of big data, resulting in high false-positive rates and missed fraud (Aljunaid et al., 2025). In this context, machine learning (ML) has emerged as a powerful tool to detect fraud more **effectively**. ML techniques can automatically learn complex patterns from large transaction datasets and detect anomalies or suspicious behaviors that would evade manual rules (Khetani *et al.*, 2023). Consequently, there has been intense academic and industry interest in applying ML to financial fraud detection, aiming to improve the accuracy, speed, and adaptability of fraud detection systems.

Multiple categories of financial fraud fall under the umbrella of online transaction fraud, including credit card fraud, online banking fraud, mobile payment fraud, insurance claim fraud, securities and investment fraud, money laundering, and other related offenses (Ali et al., 2022). Each domain has its nuances – for example, credit card fraud involves unauthorized card use either offline (stolen cards) or online (card-not-present transactions) (Ali et al., 2022; Nicholls et al., 2021), whereas insurance fraud might involve false claims in healthcare or auto insurance contexts (Ali et al., 2022; Hernandez Aros et al., 2024). Despite this variety, the core challenge is identifying abnormal transaction patterns among vast volumes of legitimate data. Machine learning is well-suited to this anomaly detection problem, which explains its growing adoption in fraud detection systems. ML models can be trained to recognize fraudulent patterns that are subtle or not explicitly defined by human analysts, thus complementing or surpassing rule-based methods in both coverage and agility (Ikemefuna et al., 2024; Nicholls, Kuppa and Le-Khac, 2021).

A diagram of financial fraud

AI-generated content may be incorrect.

Figure 1. Financial fraud classification framework across financial sectors (Zhu et al., 2021).

In summary, detecting fraudulent online financial transactions is an increasingly important and challenging task. The scale of the problem (billions of dollars in losses) and the limitations of traditional methods have driven the shift toward data-driven, learning-based techniques. In the following sections, we provide a comprehensive review of how machine learning has been applied to this domain, covering the main ML approaches, commonly used datasets, feature engineering practices, evaluation metrics, key challenges, tools, and open research directions. This review synthesizes findings from over a decade of academic work (with a focus on recent years) to serve as a foundation for further research in fraud detection using machine learning.

2.2. Machine Learning Approaches to Fraud Detection

A wide range of machine learning approaches has been explored for fraud detection in online financial transactions. These include **supervised learning methods, unsupervised and semi-supervised techniques for anomaly detection, ensemble methods, deep learning models,** and even emerging paradigms such as **reinforcement learning**. Each approach has its strengths and weaknesses in the fraud detection context. Below, we summarize the main techniques and compare their utility:

2.2.1. Supervised Learning (Classification)

The majority of fraud detection research has focused on supervised learning, where models are trained on labeled transaction data (fraud vs. non-fraud) (Waleed Hilal, S. Andrew Gadsden, and John Yawney, 2021; Hernandez Aros et al., 2024). Popular algorithms include **logistic regression, decision trees, random forests, support vector machines (SVMs), and neural networks**. Supervised classifiers tend to achieve high accuracy when ample labeled examples of fraud are available, and they can directly optimize metrics like classification accuracy or F1-score (see metrics section for more details). For example, decision tree ensembles (like Random Forests and Gradient Boosted Trees) have been widely used and often perform well due to their ability to handle nonlinear relationships and feature interactions. In fact, a recent survey found that Random Forest was among the most frequently used models in supervised fraud detection studies (Hernandez Aros et al., 2024). Likewise, SVMs have been applied successfully; research has shown SVM models outperforming earlier methods in credit card and insurance fraud detection cases (Ali et al., 2022).

**Strengths:** Supervised models can leverage well-understood algorithms and yield high detection performance if trained on representative data. They produce a clear decision output (fraud or not) for each transaction and can incorporate cost-sensitive learning to penalize misclassification of fraud heavily.

**Weaknesses:** They require large quantities of labeled fraud data, which is often scarce. By nature, supervised models struggle with new fraud patterns not present in the training data; they tend to detect “known” fraud patterns but can miss novel schemes. Moreover, in highly imbalanced settings (fraud cases << legitimate cases), supervised learners may be biased toward the majority class without special handling. Many studies address this via resampling or algorithmic techniques (discussed later), but it remains a challenge.

2.2.2. Unsupervised and Semi-Supervised Learning (Anomaly Detection)

Because fraudulent transactions are rare and evolving, unsupervised learning techniques are crucial to detect anomalies without relying on labeled examples (Ali et al., 2022; Waleed Hilal, S. Andrew Gadsden, and John Yawney, 2021). Unsupervised methods attempt to model the distribution of normal transactions and flag outliers as potential fraud. Common approaches include **clustering algorithms** (k-means, DBSCAN), **density estimation**, and **one-class classification** methods (One-Class SVM, Isolation Forest). For instance, clustering has been used to group similar transactions and identify outlier clusters that correspond to fraud rings (Malini and Pushpa, 2017; Ahmed, Mahmood, and Islam, 2016). One-Class SVM and Isolation Forest have been applied to credit card data to detect suspicious events without needing any fraud labels (Ali et al., 2022). **Autoencoders** (a type of unsupervised deep neural network) have also gained popularity for fraud detection – they learn to reconstruct “normal” transactions and instances with high reconstruction error are flagged as anomalies (Hernandez Aros *et al.*, 2024).

**Strengths:** Unsupervised anomaly detection can potentially identify previously unseen fraud patterns, making it adaptive to emerging threats. It does not require annotated fraud data, which is useful when labels are expensive or delayed (a common situation in financial fraud, where only some frauds are ever confirmed).

**Weaknesses:** These methods can suffer from higher false positive rates, since not every statistical outlier is fraudulent. Tuning them can be difficult, and evaluating their performance is tricky without ground-truth labels (often requiring manual review). Indeed, surveys report that unsupervised techniques, while studied, have been used **less frequently than supervised ones in the literature** (Ngai *et al.*, 2011). There is a recognized research gap here: recent reviews suggest more attention should be given to unsupervised and semi-supervised approaches, as they can uncover new insights and address the label scarcity problem (Ali et al., 2022; Waleed Hilal, S. Andrew Gadsden, and John Yawney, 2021). Semi-supervised approaches (e.g., training on a large set of unlabeled data with a small labeled subset) also show promise in combining the advantages of both worlds (Waleed Hilal, S. Andrew Gadsden, and John Yawney, 2021).

2.2.3. Ensemble Methods

Ensemble learning is a special sub-category of supervised learning, which combines multiple models to improve predictive performance. Ensembles are widely used in fraud detection to boost accuracy and robustness. Techniques include bagging (e.g., Random Forest, which aggregates many decision trees), boosting (e.g., XGBoost, LightGBM), and stacking (blending different model types). Ensembles **often win data science competitions for fraud detection** due to their ability to capture diverse patterns. For example, boosting algorithms have achieved top results on credit card fraud datasets by effectively handling imbalance and nonlinear interactions (Hernandez Aros et al., 2024). Also, top scores on one of the most recent fraud detection Kaggle competitions ([IEEE-CIS Fraud Detection | Kaggle](https://www.kaggle.com/competitions/ieee-fraud-detection/overview)) have been achieved with such methods. Ensembles can also incorporate different algorithms for different sub-tasks: one study used a hybrid of SVM, logistic regression, and linear regression as a composite model to classify transactions (Ali et al., 2022).

**Strengths:** Ensembles usually outperform individual models by reducing variance and bias – this is valuable in fraud detection, where catching as many fraud cases as possible (high recall) while keeping false alarms low (high precision) requires a delicate balance.

**Weaknesses:** The main downsides are increased complexity and reduced interpretability. A large ensemble (hundreds of trees or mixed models) can be a **“black box”** that is hard to interpret, which is problematic in regulated financial environments. Training and deploying ensembles may also be computationally heavier, though modern computing power often mitigates this.

2.2.4. Deep Learning

In recent years, deep learning techniques have been increasingly applied to fraud detection, thanks to their ability to model complex data patterns. Artificial Neural Networks (ANNs) in various forms (**multilayer perceptrons, convolutional networks, recurrent networks, graph neural networks**) have been explored. Early works used basic ANNs to classify fraudulent transactions (Sahin and Duman, 2011; Bouchti *et al.*, 2017; Dang *et al.*, 2021), while newer studies leverage specialized architectures. For example, recurrent neural networks (RNNs) and long short-term memory (LSTM) networks can capture sequential patterns in transaction streams (useful for detecting sequential fraud behavior or repeated offenses). Graph Neural Networks (GNNs) have emerged to detect fraud by modeling transactions as graphs (e.g., linking customers, merchants, IP addresses) – graph-based approaches can uncover organized fraud rings or money laundering patterns by relational analysis (Nicholls et al., 2021; Deng et al., 2021). Deep learning has also been combined with other methods: autoencoder networks for anomaly detection (unsupervised) or hybrid deep models with feature extraction followed by classification (Jurgovsky et al., 2018).

**Strengths:** Deep learning models can capture highly nonlinear and intricate patterns that shallow models might miss. They are adept at processing large-scale data and can learn feature representations automatically (reducing the need for manual feature engineering). Some deep models (like GNNs) naturally incorporate network information that is crucial for certain fraud types (e.g., collusive networks).

**Weaknesses:** Deep models typically require even larger training datasets and computational resources, while it hasn’t been proven that their results are notably better than other methods. They also tend to be **black-box** in nature, raising explainability concerns. Also deep learning algorithms have very high capacity and are more likely to overfit, if not treated with caution.

2.2.5. Reinforcement Learning

Reinforcement learning (RL) is an emerging approach that views fraud detection as a sequential decision-making problem. Instead of static classification, an RL agent learns policies for actions such as blocking a transaction, requesting additional verification, or allowing it, with the goal of minimizing fraud loss and false alarms over time. Although not as widely studied as other categories, some recent works demonstrate RL’s potential. For instance, Sofia Patel *et al.* (2025) formulate real-time fraud detection as an RL task where the agent receives rewards for correctly identifying fraud while avoiding customer insult (blocking legitimate transactions). The RL agent learns to adapt to evolving fraud strategies by continuously updating its policy based on feedback, which could address the non-stationary nature of fraud (concept drift).

**Strengths:** RL can optimize long-term objectives and handle interactive scenarios (e.g., an agent that scans transactions and decides on interventions). It is well-suited for real-time fraud prevention where decisions must consider downstream effects (like blocking one transaction might prevent future fraud or incur a customer service cost).

**Weaknesses:** RL models typically require careful reward design and substantial training experience (which in the fraud domain may equate to simulated data or long-term historical data). They can be complex to implement and validate. Moreover, the rarity of fraud makes it tricky to define rewards – a naive reward might not get enough fraud signals for the agent to learn effectively. Thus, RL in fraud detection is still largely experimental, but it represents a promising research direction for continuous learning systems that keep up with adversaries.

2.2.6. Comparison of Approaches

In practice, the choice of ML approach often depends on data availability and operational requirements. The table below (conceptually) summarizes strengths/weaknesses:

* **Supervised:** High precision/recall if trained well, but needs labels and may miss novel fraud.
* **Unsupervised:** Can catch new patterns and needs no labels, but higher false positives, harder to evaluate.
* **Ensembles:** High accuracy and robustness, but complex and black-box.
* **Deep Learning:** Powerful pattern extraction, handles big data, but data-hungry, black-box, potentially overfits.
* **Reinforcement:** Adapts to changing patterns, optimizes long-term reward, but is complex to train and not widely proven yet.

Overall, hybrid approaches are common, combining multiple techniques to capitalize on their complementary strengths (Ali et al., 2022). The literature demonstrates that **no single method is a silver bullet – effective fraud detection often requires an ensemble of strategies, careful calibration to the domain, and continuous updating as fraudsters evolve their methods.**

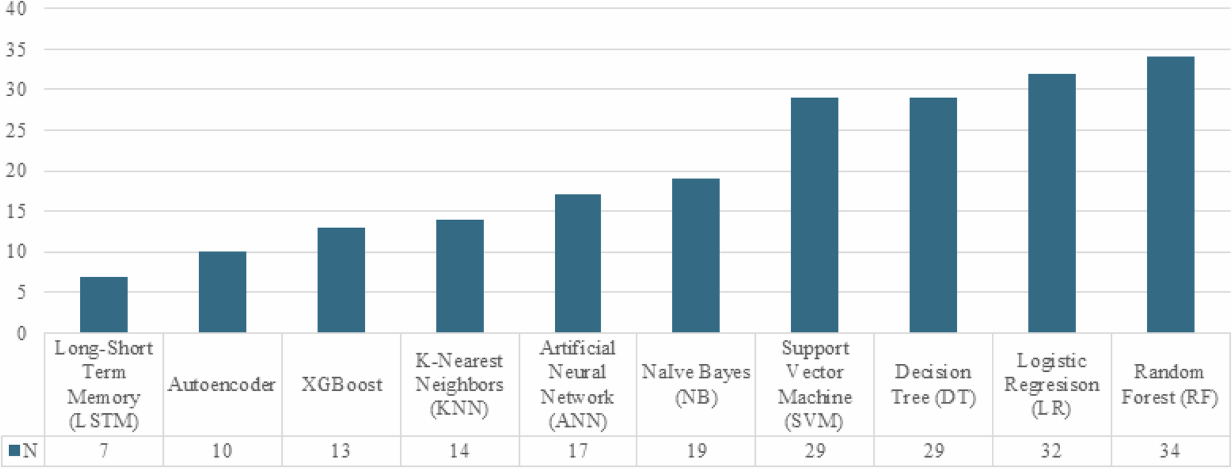


Figure 2:Main machine learning models used for financial fraud detection, in papers included in literature review by Hernadez Aros et al. (2024)

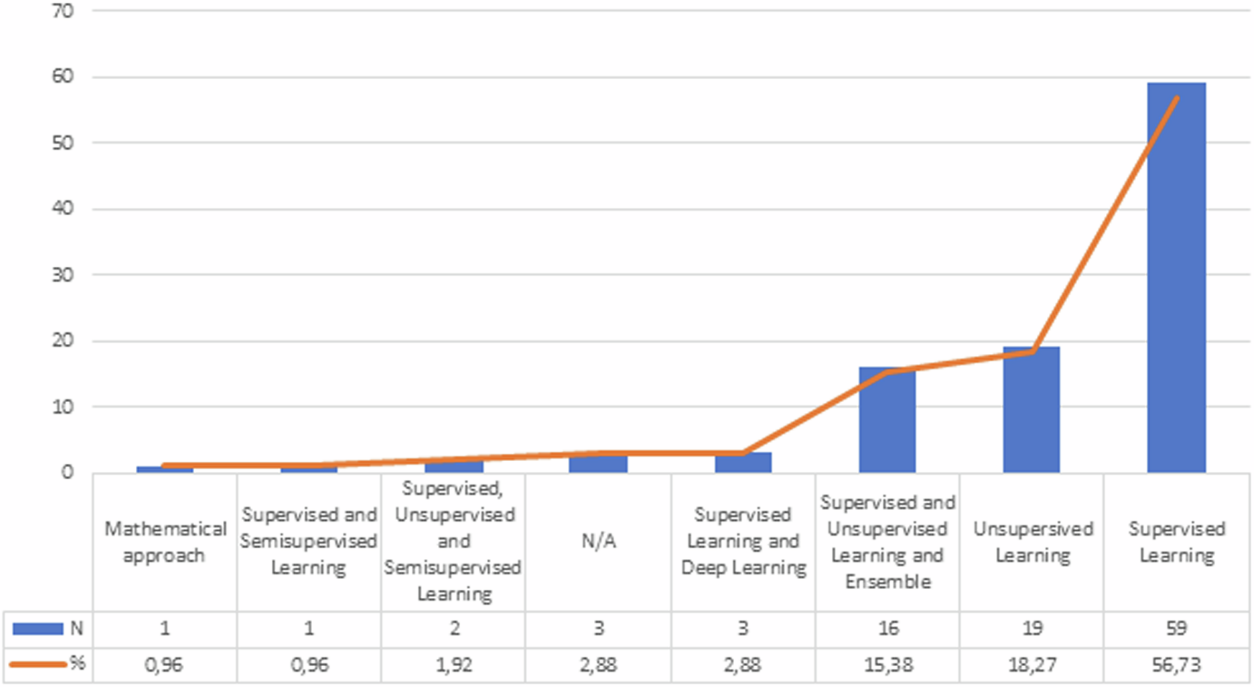


Fig. 8: Approaches used in the experiments included in the literature review of Hernadez Aros et al (2024).

2.3. Datasets and Benchmarks

Access to quality data is crucial for developing and benchmarking fraud detection models. Over the years, researchers have utilized a variety of datasets, both real-world datasets (often proprietary or released for research) and synthetic datasets, to evaluate ML techniques. Here, we outline some of the most used datasets and benchmarks in the field:

**Credit Card Transaction Datasets:** (dataset link: [Credit Card Fraud Detection](https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud)) The most referenced benchmark is the European credit card fraud dataset made public by researchers from Université Libre de Bruxelles (ULB) in 2013. This dataset contains 284,807 credit card transactions (aggregated over two days), of which 492 are labeled as fraudulent (Dal Pozzolo et al., 2018). The data include numerical features (which are results of PCA transformations for confidentiality) and the transaction amount and time. Despite its anonymization and relatively low fraud ratio (~0.17%), this dataset has become a standard benchmark – it was used by at least 15 studies in recent years for evaluating various ML models (Hernandez Aros et al., 2024). Its popularity stems from being one of the few publicly available real fraud datasets. Models achieving high AUC (>0.95) on this data are considered state-of-the-art, though issues like its static nature and limited feature information are noted.

**E-commerce Transactions and Mixed Online Payment Data: (our choice)** (Dataset link: [IEEE-CIS Fraud Detection | Kaggle](https://www.kaggle.com/competitions/ieee-fraud-detection/data)) In 2019, a large industry dataset was released through a Kaggle competition (the IEEE-CIS Fraud Detection challenge). This dataset, although proprietary in origin, was made public for the contest and contains over 1 million online transactions with a rich set of features (device information, IP address, product codes, etc.) and a binary fraud label. Researchers sometimes use this dataset to evaluate performance on e-commerce fraud detection, as it reflects modern online payment fraud scenarios. It is highly unbalanced and includes many categorical features, testing an algorithm’s ability to handle feature engineering and big data. (While direct academic papers on it are few, it has indirectly influenced research by revealing which techniques perform well in practice — for example, winning solutions used extensive feature engineering and ensemble models.)

**PaySim Mobile Money Dataset:** (Dataset link: [Synthetic Financial Datasets For Fraud Detection](https://www.kaggle.com/datasets/ealaxi/paysim1)) PaySim is a synthetic dataset generated to simulate mobile payment transactions, based on patterns from real financial data. It was created using the PaySim simulator (by Edgar López-Rojas and colleagues) and released on Kaggle. One commonly used version contains about 6.3 million transactions (with a few thousand labeled frauds) across different transaction types (payments, transfers, cash-out, etc.) (Lopez-Rojas et al., 2016). PaySim is valuable for research because it provides large-scale data with realistic behavior for fraud (though the fraud instances are simulated). Several studies have used PaySim to test algorithms that need big data and to validate the scalability of methods (Hernandez Aros et al., 2024). The synthetic nature means results might not fully translate to a real deployment, but it remains a widely cited benchmark for comparing unsupervised techniques and testing under high class imbalance.

**BankSim and Other Simulated Bank Transaction Data: (**Dataset Link: [Synthetic data from a financial payment system](https://www.kaggle.com/datasets/ealaxi/banksim1)**)** BankSim is another simulator-based dataset (simulated from a Spanish bank’s data sample) that provides transactions with attributes like customer ID, merchant, location, etc., along with fraud labels (Lopez-Rojas et al., 2014). It has been used in studies focusing on fraud detection in banking transactions and often for testing pattern recognition algorithms in a controlled environment (Hernandez Aros et al., 2024). Similar to PaySim, it provides a flexible testbed for new algorithms without privacy concerns.

It is worth noting that data imbalance is a pervasive issue across these datasets. This reflects real-world base rates and makes these benchmarks valuable for testing how algorithms handle imbalanced data. Many studies report performance in terms of area under the ROC curve (AUC) or precision/recall rather than raw accuracy due to this imbalance (discussed in the next section on metrics).

**Real vs. Synthetic Data:** Recent literature reviews have found that a majority of studies now use real datasets in experiments, with relatively few relying solely on synthetic data (Hernandez Aros et al., 2024). This suggests a push towards evaluating models on realistic data distributions. The availability of open datasets like the ULB credit card data and various Kaggle datasets has facilitated this. However, obtaining real fraud data beyond these public sets remains a challenge – many researchers collaborate with financial institutions to test their models on private datasets (e.g., a bank’s internal transaction logs), but such results may not be fully public. Synthetic datasets (like PaySim) help fill this gap by allowing algorithm development and preliminary testing in a reproducible way. In fact, there are ongoing efforts to generate better simulation datasets. For example, Ramachandran et al. (2023) introduced FraudAmmo, a large-scale synthetic transactional dataset for payment fraud research. Such resources aim to provide realistic scenarios for training and evaluating fraud detectors while avoiding privacy issues.

2.4. Feature Engineering for Fraud Detection

Effective feature engineering is often cited as one of the most crucial steps in building a successful fraud detection model (Ti et al., 2022). Raw transaction logs are usually high-dimensional, heterogeneous, and not immediately suitable for feeding into ML algorithms. Therefore, researchers and practitioners devote significant effort to creating informative features. In this section, we discuss typical feature types and engineering techniques used in fraud detection.

**Transactional Behavior Features (RFM):** A common framework for thinking about customer transaction behavior is the **Recency, Frequency, Monetary (RFM) model**. Prior studies often derive features based on how recently transactions have occurred, how frequently they appear in a given period, and the monetary values involved (Ti et al., 2022). For example, features can include the time since last transaction, number of transactions in the past 24 hours (or past week, month, etc.), average transaction amount, and maximum transaction amount. These RFM features help characterize if an account’s recent behavior deviates from its usual pattern.

**Anomaly and Derived Features:** In addition to straightforward stats, researchers use anomaly detection features. For instance, an “amount deviation score” (how far a transaction’s amount is from the customer’s usual amount distribution) could be a feature. Or the output of an unsupervised model (like an autoencoder’s reconstruction error for a transaction) can be used as a feature input to a supervised model – essentially a meta-feature indicating how anomalous a transaction appears (Ti et al., 2022). Ti et al. (2022) found that anomaly features (like ones based on outlier scores) performed the worst among feature categories in their experiments, possibly because they are redundant with other signals or too noisy. However, they remain part of the feature engineering toolbox as they can highlight novel patterns.

**Time and Sequence Features:** Timing is critical in fraud. Features capturing temporal patterns include time-of-day or day-of-week of transactions (fraud might happen at unusual times for a given user), velocity features (e.g., number of transactions in the last hour), and sequence patterns (like rapid successive transactions at different merchants). ML models can learn such thresholds if provided with features like “transactions\_per\_hour”. Geographic and IP features also fall in this category, e.g., distance between successive transaction locations, or change in IP address.

**Identity and Device Features:** In online transactions, information about the user’s device, browser, IP address, etc., can be critical. Fraudsters often use different devices or anonymization. Thus, features like device ID, IP address, email domain, etc., are used to connect transactions. If historical data is available, one might compute “number of distinct devices used by this account in the last N days” or “first time this device is seen for any transaction”. These features aim to detect identity inconsistencies.

**Feature Selection and Importance:** Given a large pool of candidate features, it’s important to evaluate which are most useful. Techniques like mutual information, correlation analysis, or tree-based feature importance can be used to select a meaningful subset. Some studies report that a handful of carefully engineered features can outperform hundreds of raw attributes. In fact, Butaru et al. (2017) showed that features related to customer behavior stability were top predictors for credit card fraud in a large-scale analysis.

Feature generation is proven to be really important for fraud detection algorithms. Studies consistently show that without good features, even the most sophisticated algorithms underperform (Cheng *et al.*, 2020; Baesens, Höppner, and Verdonck, 2021). Conversely, a simple model with well-crafted features can achieve strong results. The ongoing challenge is to develop features that are general enough to catch new fraud schemes yet specific enough to avoid false positives – an area where domain **expertise and creativity are as important as technical skills**.

2.5. Model Evaluation Metrics

Choosing the right evaluation metrics is crucial for fraud detection because of the **class imbalance and the high cost asymmetry** between false positives and false negatives. A variety of metrics are used in the literature, each providing different insights.

**Confusion Matrix Basics:** At its core, fraud detection can be evaluated by the confusion matrix counts: True Positives (TP = fraudulent transactions correctly identified), True Negatives (TN = legitimate transactions correctly passed as non-fraud), False Positives (FP = legitimate transactions incorrectly flagged as fraud), and False Negatives (FN = fraudulent transactions missed by the model). From these, many metrics are derived (Hernandez Aros et al., 2024). A key point is that in fraud detection, **false negatives (missed frauds) directly translate to financial loss, while false positives (false alarms) cause customer friction and operational cost**. Thus, organizations often prioritize high recall (catch as much fraud as possible), subject to keeping false positives within manageable limits.

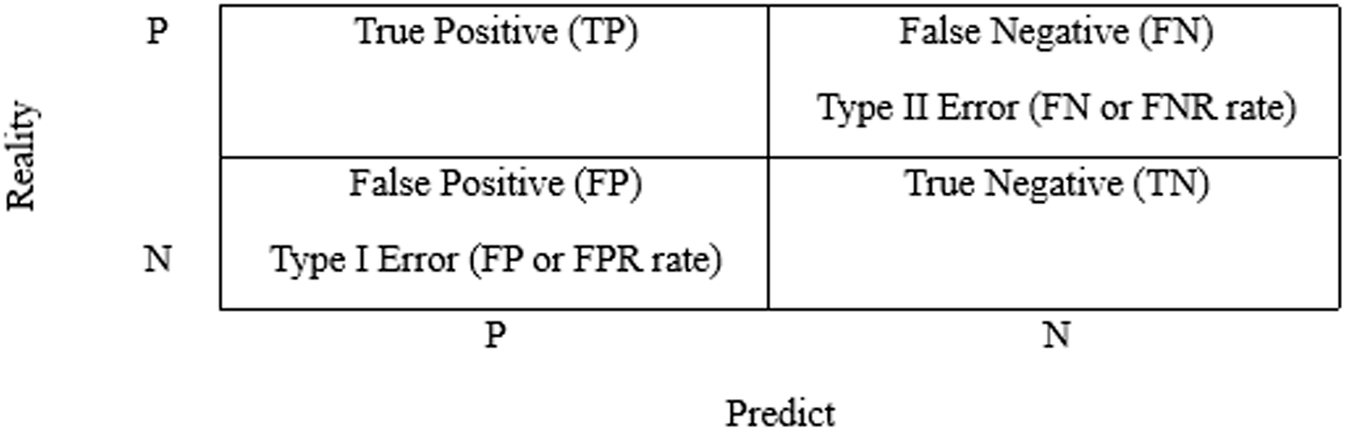


Figure 3: Confusion Matrix (Hernandez Aros et al., 2024)

* **Accuracy:** While accuracy is a standard metric in classification, it is misleading for fraud detection because of extreme class imbalance. If frauds are 0.1% of transactions, a trivial classifier that labels everything as “legitimate” will be 99.9% accurate but completely useless.
* **Precision (Positive Predictive Value):** This is the proportion of flagged transactions that are actually fraudulent. Precision answers “if the model says fraud, how often is it correct?” High precision means few false alarms. This is important for operational efficiency – **low precision means investigators waste time on many false leads.**
* **Recall (Sensitivity or True Positive Rate):** This is the fraction of actual frauds that the model catches. Recall addresses “how much fraud is caught by the system?” **High recall is crucial to minimize fraud losses.**
* **F1-Score:** The harmonic mean of precision and recall, F1 = 2 \* (Precision \* Recall) / (Precision + Recall). F1 gives a single measure that balances both, useful when we want a trade-off assessment. In fraud detection, one might optimize models for maximum F1 or set a recall target and then maximize precision.

**ROC-AUC (Area Under the ROC Curve):** AUC has been widely used in fraud detection research as a threshold-independent metric (Chen & Wu, 2022). On heavily imbalanced data, AUC can be high even if the model isn’t great at finding the minority class, because it gives equal weight to TPR and TNR. However, it’s still a standard metric. Many papers report AUC values; for example, an AUC in the 0.95+ range on the credit card dataset is considered state-of-the-art (Chen & Wu, 2022). One advantage of ROC-AUC is that its prevalence in the literature allows easy comparison, and it’s not tied to a particular threshold. But a criticism is that ROC can be overly optimistic under imbalance, since it includes regions of false positive rate that are not operationally relevant (e.g., a very low FPR region might be more important to focus on).

**Precision-Recall Curve and PR-AUC:** The Precision-Recall curve focuses on the minority class performance by plotting Precision vs. Recall. PR curves can be more informative than ROC when fraud is rare, as they zoom in on how precision drops as you try to increase recall. The area under the PR curve (average precision) is another scalar metric; a high PR-AUC means the model maintains good precision across a range of recall levels. Some researchers advocate for PR-AUC in addition to ROC-AUC for imbalanced problems. In practice, it’s less commonly reported than ROC-AUC, but it is very useful. For example, an algorithm might have an ROC-AUC of 0.98, but if the fraud is 0.1%, its PR-AUC might be much lower (due to class imbalance) – improving PR-AUC directly relates to better fraud catch with fewer false alerts.

**Cost-Based and Composite Metrics:** In some studies, especially those with an industry angle, metrics incorporate **cost**. For example, a cost matrix can be applied by assigning **a dollar cost to each FP and FN, then compute total cost of errors**. A notable approach is to use the **Expected Value framework** – several papers calculate the monetary value of model decisions (Dal Pozzolo *et al.*, 2018; Correa Bahnsen, Aouada, and Ottersten, 2015; Whitrow *et al.*, 2009). While academically many works stick to F1/AUC, cost-based evaluation is recommended for real-life deployment.

A systematic review by Hernandez Aros et al. (2024) found that the **most prevalent metrics in fraud detection studies are precision, recall (sensitivity), F1-score, and ROC-AUC**, reflecting the community’s focus on both detection coverage and correctness (Hernandez Aros et al., 2024). Many papers list multiple metrics for completeness. However, simply maximizing accuracy is recognized as inadequate – hence, authors tend to discuss precision/recall trade-offs explicitly.

One interesting point is evaluating **unsupervised methods**. Without labels, unsupervised model performance can be measured using internal clustering metrics (like silhouette coefficient, Davies–Bouldin index) if treating fraud detection as clustering (Amrutha et al., 2023; García-Ordás et al., 2023; Palacio, 2019). But these are not directly indicative of detection performance. In practice, unsupervised methods are eventually evaluated on labeled test sets too (e.g., assign outlier scores and evaluate how well they rank true frauds). Hernandez Aros et al. (2024) note that for unsupervised techniques, studies sometimes report metrics like silhouette score to show clustering quality, but the ultimate measure is **whether those clusters align with fraud**. Some works use a proxy label or assume a certain percentage of outliers.

2.6. Challenges and Limitations in Machine Learning-Based Fraud Detection

Despite substantial progress, deploying machine learning for fraud detection in online financial systems faces numerous challenges and limitations. These arise from the inherent nature of fraud (adaptive adversaries, rarity of events) and practical constraints (data privacy, system scalability, etc.). We outline the key challenges identified in the literature and their implications:

**Class Imbalance:** As noted multiple times, fraud datasets are extremely imbalanced – genuine transactions vastly outnumber fraudulent ones (Hernandez Aros et al., 2024). ML models tend to be biased towards predicting the majority class (“not fraud”), potentially missing many frauds unless carefully trained. Also evaluation of models becomes tricky – high overall accuracy is easy to achieve while missing all fraud. Techniques like resampling (oversampling frauds or undersampling non-frauds), cost-sensitive learning, and specialized algorithms (e.g., one-class classifiers) are used to address imbalance, but none is a panacea. Excessive oversampling can lead to overfitting or “sampling artifacts,” whereas undersampling throws away useful data. Researchers often consider evaluation metrics that focus on the minority class (precision/recall, AUC) to mitigate the skewed feedback of accuracy (Nicholls et al., 2021).

**Evolving Fraud Tactics (Concept Drift):** Fraud patterns continuously evolve as fraudsters adapt to bypass known defenses. Many academic studies assume stationarity (training and test from the same distribution), but in real deployments, models must be updated frequently. Ali et al. (2022) observe that addressing concept drift by retraining periodically incurs high overhead, and model performance can suffer between retraining cycles. Online learning algorithms or incremental updates are potential solutions, but are complex to engineer. Reinforcement learning approaches are one idea to handle non-stationarity by continuous adaptation (Patel et al., 2025), but this is still nascent. In practice, a combination of human analyst feedback and model retraining on recent data is used to tackle drift, but this lag means new fraud types can still cause damage before models catch up. The challenge is essentially a cat-and-mouse game with adversaries, a unique aspect of fraud detection compared to many ML tasks.

**False Positives vs. False Negatives Trade-off:** In fraud detection, false positives result in time cost of investigating a legitimate transaction that has been marked as fraudulent, whereas false negatives result in cost of missed fraud. The above trade-off is constantly being examined as it remains a challenge to create a model that has high recall and precision at the same time.

**Explainability and Interpretability:** Financial institutions operate in regulated environments and need to explain decisions. ML models, particularly complex ones (deep neural nets, ensembles), are often **black boxes** that do not provide clear reasoning for why a transaction was flagged (Aljunaid *et al.*, 2025). This lack of transparency can erode trust from fraud analysts and compliance officers. Explainability is crucial for analyst acceptance, as an investigator presented with hundreds of alerts needs clues as to why the model marks them as fraud. Recent research has started addressing this by applying explainable AI (XAI) techniques to fraud detection. For example, safer ML models like decision trees or rule-based systems are inherently interpretable but might be less accurate. Alternatively, post-hoc explanation tools like **SHAP or LIME** can be used on complex models to highlight influential features (Aljunaid et al., 2025). While progress is being made, achieving a balance between accuracy and interpretability is a significant challenge; many of the best-performing models are ensemble or deep models that inherently lack simple explanations.

**Data Privacy and Sharing:** Financial data is sensitive. Sharing data between institutions or even within departments of a bank is often restricted by privacy regulations and business secrecy. This limits the size and diversity of fraud datasets that any one entity can use for training. A fraud pattern seen at one bank might help others, but data cannot be directly shared. Federated Learning has been proposed as a solution, where multiple institutions collaboratively train a model without sharing raw data (Aljunaid et al., 2025). While promising, federated learning for fraud detection is still in early stages, with challenges like aligning feature spaces and ensuring no leakage of sensitive info through model updates. Another privacy issue is personal data usage – features involving personal identifiers might be subject to regulations. Even storing certain data for feature engineering (like device IDs or location) must be handled carefully (anonymization, encryption at rest, etc.). These constraints can limit the available features or require heavy compliance oversight. In research, this is a limitation because the best academic solutions may use data that in practice would violate privacy policies. Thus, a gap often exists between academic prototypes and deployable solutions due to privacy considerations.

**Label Quality and Ground Truth:** Not every fraud is known. The training labels for fraud detection are typically based on confirmed fraud cases (chargebacks, confirmed investigations). There is a zone of uncertainty: some transactions might be fraud that went undetected (false negatives that remain unlabeled), and some might be labeled fraud that are later found to be legitimate (if errors occur in the investigation). This label noise can affect model training. Semi-supervised learning and human-in-the-loop systems help mitigate this by iteratively improving labels. But fundamentally, if the training set is missing entire classes of fraud (e.g., a new scam not seen before), the model cannot learn it. Many researchers mention this limitation: ML can only detect what it’s been taught (or what deviates significantly as an anomaly). As fraudsters innovate, completely new fraud patterns initially have no labels and can slip through. This has led to suggestions for continual learning and anomaly detection to catch those and then quickly incorporate them as new labeled examples – essentially shortening the feedback loop.

2.7. Tools, Libraries, and Frameworks

Implementing machine learning for fraud detection leverages a broad ecosystem of tools and software frameworks. Given the variety of techniques (from training complex deep networks to deploying real-time decision systems), different tools come into play for different stages. Here we outline some of the most commonly used tools, libraries, and frameworks in this domain, as reported in the literature and industry case studies:

**General Programming Languages:** **Python** is overwhelmingly popular in the ML community and in fraud detection prototyping. Many academic papers use Python for experiments, thanks to its rich set of ML libraries. **R** is also used in some academic works, particularly those focusing on statistical techniques or when analysts (with a statistics background) drive the effort.

**ML Libraries and Frameworks:** For traditional machine learning algorithms, scikit-learn (Python) is a go-to library. It provides implementations of logistic regression, decision trees, random forests, SVM, clustering, etc., which are frequently used in fraud research. Many papers cite using scikit-learn for baseline models because of its simplicity and reliability. For deep learning, TensorFlow and Keras (TensorFlow’s high-level API), as well as PyTorch, are widely used to build and train neural networks for fraud detection experiments.

**Imbalanced Data Tools:** Since class imbalance is a key issue, libraries like Imbalanced-learn (which provides implementations of SMOTE, ADASYN, etc.) are used alongside scikit-learn to handle resampling.

**Visualization and Analysis:** Tools like Jupyter Notebooks are commonly used for exploratory data analysis and initial model prototyping in research. Fraud data often requires visualization to see patterns (like plotting transaction amounts or network graphs of transactions). Libraries such as matplotlib, seaborn for charts, and NetworkX for graph visualization are handy.

## 2.8. Research Gaps and Future Directions

While machine learning has greatly advanced fraud detection, there are still notable gaps in literature and many open problems. Researchers have identified several areas where further work is needed to enhance effectiveness, efficiency, and scope. Based on recent surveys and emerging trends, here are some of the possible future directions that could be subject to improvement:

**Improving Unsupervised and Semi-Supervised Methods:** As discussed, most current systems rely on supervised learning and known fraud patterns. A gap exists in detecting previously unseen (zero-day) fraud schemes. Future research is expected to focus on more powerful unsupervised and semi-supervised techniques, allowing models to flag anomalies that don’t resemble any known fraud (Ali et al., 2022; Hernandez Aros et al., 2024). This includes advanced clustering methods, deep anomaly detectors, and hybrid models that can learn from few labeled examples (one-shot or few-shot learning for fraud).

**Handling Concept Drift and Adaptive Learning:** Dealing with concept drift remains a crucial research area. Future systems need to learn and update in real-time or near-real-time as fraudsters change tactics. One direction is online learning algorithms that update model parameters continuously with each new batch of data, rather than periodic retraining (Ali et al., 2022). Another direction is reinforcement learning and other adaptive frameworks, which treat fraud detection as a dynamic environment – initial studies in this space have shown promise in models that self-tune to evolving patterns (Patel et al., 2025; Aljunaid et al., 2025).

**Explainability and Transparency (XAI for Fraud):** As noted in the challenges section, the lack of interpretability in complex models is problematic. A clear research direction is developing explainable AI techniques tailored for fraud detection. This might involve creating models that naturally output human-understandable patterns (e.g., rule-based ML that extracts logical rules for fraud) or applying post-hoc explainers to black-box models. Recent work like Aljunaid et al. (2025), combining SHAP and LIME explanations with a federated model, is one example (Aljunaid et al., 2025).

**Privacy-Preserving and Federated Learning:** Collaboration across institutions is a powerful weapon against fraud (since fraudsters often hit multiple targets), but sharing data is limited by privacy. Federated learning (FL) offers a way for institutions to jointly train models on their combined data without exposing sensitive information. Future research is likely to refine FL techniques for fraud: improving how models aggregate without sharing vulnerabilities, handling heterogeneity in data across institutions, and ensuring even federated models are interpretable (the XFL – explainable federated learning – concept) (Aljunaid et al., 2025). There are also related approaches like secure multi-party computation and homomorphic encryption that could allow computing fraud models on encrypted data. These are computationally heavy now, but an open area is making them feasible for real-world fraud detection. If successful, this could lead to “consortium” models that benefit from vastly more data. A concrete example: banks could collaboratively train a model to detect fraudulent credit applications across the industry without ever sharing their customer data directly – research prototypes of such systems are under exploration.

**Integration of Text and Unstructured Data:** A lot of fraud detection research focuses on structured transaction records. However, there is rich unstructured data (emails, claim documents, customer phone call transcripts, dark web intelligence) that could aid fraud detection. Text mining and NLP for fraud is a relatively under-explored area noted by reviewers (Shahana et al., 2023).

**Graph-Based Fraud Detection and Network Analysis:** While graph techniques have emerged, there is still a lot of room to grow. Graph Neural Networks (GNNs) and advanced network algorithms are at early stages of application to fraud.

**Evaluation and Benchmarking Improvements:** A meta-level gap is the inconsistency in evaluation across studies. Future work could establish standardized benchmarks and evaluation protocols specific to fraud detection. This might involve creating shared (perhaps simulated but agreed-upon) datasets for different fraud domains, and defining evaluation metrics that encapsulate cost better.

In conclusion, the arms race nature of fraud means research is never “done.” Each advancement by defenders (researchers, practitioners) is eventually met with new strategies by attackers, which in turn raises new research questions. Current literature is rich in algorithms and case studies, but gaps remain in adaptability, generalization, and deployability of these systems. The future will likely see more emphasis on robust, self-learning fraud detection systems that can explain their reasoning and respect privacy constraints while operating at scale. By addressing the gaps above – especially unsupervised detection, concept drift, interpretability, and collaborative modeling – the next generation of fraud detection tools will be more resilient and effective. Researchers are actively pursuing these directions, making fraud detection a dynamic and continually evolving field.

## 2.9. Conclusion

Machine learning has become an indispensable component of modern fraud detection in online financial transactions. This literature review has highlighted how and why ML is applied to this domain, summarizing key approaches (from supervised classifiers to deep networks and beyond), datasets and features commonly used, evaluation practices, and the challenges that persist. In summary, ML offers substantial improvements in accuracy and adaptability for fraud detection, but it also introduces new challenges like model interpretability and the need to handle adaptive adversaries.

The problem of fraud is adversarial, costly, and evolving – which means that research and development in this area must also evolve. We see a clear trajectory in the literature: starting from basic anomaly detection and supervised models in early years, moving towards more sophisticated ensemble and deep learning methods, and now pushing into frontiers like graph analysis, federated learning, and real-time adaptive systems. As financial transactions increasingly digitize and fraudsters become more tech-savvy, the importance of machine learning in fraud detection will only grow.

Crucially, success in this field requires more than just algorithms. It demands understanding the fraud domain (patterns of fraud, fraudster behavior), ensuring robust data pipelines and feature engineering, selecting appropriate metrics that reflect business priorities, and maintaining a human-ML partnership where each complements the other. The literature provides a foundational knowledge base – for instance, reviews by Ali et al. (2022) and Hernandez et al. (2024) offer broad mappings of techniques to fraud types, and Nicholls et al. (2021) delves into deep learning strategies for cyber-fraud – and these can guide new researchers in identifying what has been done and where contributions can be made.

The future directions discussed show that there is ample room for innovation. Making fraud detection models more adaptive, collaborative, and transparent stands out as a unifying theme for future work. By addressing current limitations – such as improving detection of novel frauds, reducing false positives through better explanations, and leveraging cross-organization data safely – the next advancements will aim to outpace fraudsters and protect financial systems with even greater efficacy.

For a master’s thesis focused on this area, grounding your work in the insights and gaps identified in prior studies is essential. We recommend building on the strengths of existing approaches while explicitly tackling one or more of the identified gaps. Whether it’s experimenting with a new unsupervised algorithm on a challenging fraud dataset or devising an interpretable model that an analyst can trust, contributions along these lines will be valuable to both the academic community and industry practitioners. Fraud detection is not a solved problem, but with each research iteration – informed by thorough literature surveys like this – we move closer to more secure and fraud-resistant financial transaction ecosystems.

# 3. Data

For this thesis, we picked the IEEE fraud detection dataset ([IEEE-CIS Fraud Detection | Kaggle](https://www.kaggle.com/competitions/ieee-fraud-detection/data)). This dataset contains real-world anonymized data from card transactions. It consists of a training dataset, with labeled fraudulent data, and a test set that is not labeled. Train dataset has 590540 rows and 434 columns and test dataset has 506691 rows and 433 columns (1 less column, which is the fraud label). Before training our machine learning models, we have to analyze the nature of the data. In the appendix, we have conducted a detailed explanatory data analysis. On the following section we present the most important results.

3.1. Exploratory Data Analysis

3.1.1. Overview

* **Two main tables:**
  + Transaction Table (core features about transactions)
  + Identity Table (network, device, and digital fingerprint features)
* **Target variable: isFraud**
  + 1 = reported fraud (chargeback + linked transactions)
  + 0 = legitimate transaction
* **Transaction Table Core Features**
  + TransactionDT:
    - Timedelta in seconds from a reference point (not actual timestamp).
    - First value = 86,400 (one day).
    - Maximum ~15,811,131 → ~183 days (~6 months).
  + TransactionAMT:
    - Payment amount in USD.
    - Sometimes 3 decimal places → may indicate currency conversion.
  + ProductCD:
    - (not necessarily physical product).
  + card1–card6:
    - card information (type, category, issuer, country, etc.).
  + addr1, addr2:
    - addr1 = billing region,
    - addr2 = billing country.
  + dist1, dist2: Distance measures (between billing, mailing, IP, phone area, etc.).
  + P\_emaildomain / R\_emaildomain: Purchaser and recipient email domains.
* **Engineered / Aggregated Features**
  + C1–C14: Counting features. (e.g., number of devices, IPs, phone numbers, billing addresses linked to a card).
  + D1–D15: Timedeltas. (e.g., days since last transaction, days between events).
  + M1–M9: Match flags.  (e.g., whether name on card matches billing address).
  + Vxxx (V1–V339): Rich engineered features by Vesta: ranking, counting, entity relationships (e.g., frequency of card + IP + email within 24h).
* **Identity Table**
  + Description: Contains identity and device-level features (network, device, browser, OS fingerprints). Fields are masked; no dictionary provided (for privacy)
  + Key Fields:
    - DeviceType  (e.g., mobile/desktop).
    - DeviceInfo (specific device string).
    - id\_01–id\_11: numerical signals
    - id\_12–id\_38: categorical/numerical mix

3.1.2. Target Variable Analysis

A white square with blue squares

AI-generated content may be incorrect.

Detailed Statistics:

* Total Transactions: 590,540
* Fraud Transactions: 20,663
* Non-Fraud Transactions: 569,877
* Fraud Rate: 3.50%

We can see that the class imbalance that is encountered on many fraud detection datasets and is also a phenomenon that is observed in the real world, is shown here too. Only 3.5% of the train data is labeled as fraud.

3.1.3. Missing Values

A blue and white striped background

AI-generated content may be incorrect.

* Features with more than 50% missing values: 214
* Features with 20-50% missing values: 38

We can see that we have a lot of missing values. However, this is expected in a real-world dataset. Often in the real world, the available data is not complete, some info might be missing or corrupted. Especially in the fraud detection case, data missingness might additionally give information about whether a transaction is legit or fraudulent. For example, missingness of user identification data, might mean that the adversary couldn’t provide all the user information.

3.1.4. Transaction DT (Time)

A graph of a graph of a graph

AI-generated content may be incorrect.

Time-based Statistics:

* Total time span: 182.0 days
* Average daily transactions: 3244.7

As we can see, fraud rate and transaction volume over the timespan of the data looks relatively random at first sight. There is no visible pattern. So here we have created **2 new features:**

1. day\_of\_week: Day of week that transaction was made
2. hour\_of\_day: Hour of day that transaction was made

These features have a conceptual meaning, as we can later compare each transaction with the relative transaction amount statistics, of the corresponding day of week, or hour of day. So on these features we observe the following:

A graph with blue bars

AI-generated content may be incorrect.

A graph showing a number of blue bars

AI-generated content may be incorrect.

3.1.5. Transaction Amount

A close-up of a graph

AI-generated content may be incorrect.

**Transaction Amount Statistics:**

* Minimum amount: $0.25
* Maximum amount: $31937.39
* Mean amount: $135.03
* Median amount: $68.77
* Q1: $43.32
* Q2: $68.77
* Q3: $125.00
* Q4: $31937.39

**Fraud Rate by Transaction Amount Quartile:**

* Q1 0.042428
* Q2 0.025090
* Q3 0.028252
* Q4 0.044388

Based on the above Transaction Amount statistics, we can see that the data is **extremely skewed to the left,** and that there is a higher concentration of fraudulent transactions on the 1st and 4th quartile, (i.e. on the transactions with amount < $43.32 or amount >$125.00)

A graph of a number of decimal places

AI-generated content may be incorrect.

|  |  |  |  |
| --- | --- | --- | --- |
| Decimal Places  Is Fraud | 1 | 2 | 3 |
| False | 316638 | 198565 | 54674 |
| True | 11312 (3.45%) | 2092 (1.04%) | 7259 (11.77%) |

On the above statistics we can see that transactions that have amount feature logged with **3 decimal places, have a much higher fraud percentage**. Note that as we mentioned earlier, 3 decimal places might indicate that the transaction **included a currency exchange .**

3.1.6. Transaction Categorical Features

The transaction table contains a lot of categorical features. For readability, we do not present all the features here, since a lot of them are anonymized and their results had no significant interpretability. Below follow some interesting results.

* Product Category

A graph of different sizes and shapes

AI-generated content may be incorrect.

These categories are encoded, but we can see that C category has the largest fraud percentage

* Card related features

A graph of a bar and a bar of a graph

AI-generated content may be incorrect.

A graph of a bar and a bar of a graph

AI-generated content may be incorrect.

We can see that some card categories are more likely to be used in fraudulent transactions

* Purchaser and Receiver email domains

A graph and chart of a graph

AI-generated content may be incorrect.

A graph of a bar and a bar of a graph

AI-generated content may be incorrect.

We can see that some email domains have much higher fraud percentage than others. However, those email domains are pretty rare.

3.1.7. Count and Time features

A screenshot of a graph

AI-generated content may be incorrect.

As we can see, none of those count features have a high difference in the fraud and non-fraud distribution, or have a notably high correlation with fraud.

A collage of blue squares

AI-generated content may be incorrect.

3.1.8. Vesta Features

These features are anonymized aggregated features, they must consist of various aggregations and/or transformations (e.g. sums, averages, or even PCA components). They are numerical and might consist of NaN values. Based on a study by the winners of the Kaggle competition ([EDA for Columns V and ID | Kaggle](https://www.kaggle.com/code/cdeotte/eda-for-columns-v-and-id)), some features can be dropped, because they were deemed insignificant. So we used the rest of the features.

A triangle with red and blue squares

AI-generated content may be incorrect.

The correlation matrix shows us that a lot of features are very highly correlated. For that reason, we conducted a PCA analysis with a decreasing number of components. The following plots show the results on the explained variance of the data, relative to the number of components.

A graph with a line

AI-generated content may be incorrect.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Number of components | 100 | 50 | 20 | 15 | 13 | 12 | 10 | 5 |
| Explained variance ratio | 100% | 100% | 100% | 100% | 100% | 99.9% | 99.9% | 99.9% |

We can see that the explained variance rate is extremely high, even with minimal number of components. We will use 13 components, which is the smallest number that achieves 100% explained variance ratio.

3.2. Feature Preprocessing and Engineering

3.2.1. Encoding

For categorical feature encoding, we used a **label encoder**, since we focus only on tree-based classifiers, which are insensitive to numerical order assumptions.

3.2.2. Feature Engineering

For feature engineering, we used the following main concepts:

* **Concept 1:** For various feature pairs where **feature A: numeric and feature B: categorical**, we grouped feature A based on feature B’s categories, and then calculated mean (μ), standard deviation (σ) and finally z-value .
* **Concept 2:** For feature pairs where **feature A and B were both categorical**, we grouped feature A based on feature B’s categories, and then calculated value count and value frequency.
* **The user ID concept:** In transactional data, in the real world, there is often a user identification available (account ID, or card number, etc.). In our case, for privacy reasons, the data are anonymized. However, in many Kaggle notebooks, there are proposed feature joins/concatenations that can be constructed, so that we can simulate such identification. From the various propositions that we found, we selected the combination of:
  + **card1:**  some card identifier (possibly issuer identification) with a lot of unique values and 0% missingness
  + **addr1:** billing region
  + **D1-TransactionDay** (derived from transaction DT)**:** D1 in many Kaggle notebooks, is considered something like “time passed from registration”. So “D1-Transaction day” is a constant value for a given account.

So given the mentioned concepts, we used **Concept 1** for :

* Transaction amount based on Card Type (Credit or Debit)
* Transaction amount based on Product Type
* Transaction amount based on user ID and hour of day
* Transaction amount based on user ID and day of week
* All numerical features based on user ID

Also, we used **Concept 2 for all categorical features based on user ID.**

Finally, we encoded all categorical features using a label encoder and scaled the transaction amount using a standard scaler.

# 6. References

1. Abbas, S. *et al.* (2024) “Smart Vision Transparency: Efficient Ocular Disease Prediction Model Using Explainable Artificial Intelligence,” *Sensors*, 24(20), p. 6618. Available at: <https://doi.org/10.3390/s24206618>.
2. Abolarin, J. (2025) Banking Law and Financial Regulations: The Imperatives for Managing Stability in the Banking Sector. Austin Macauley Publishers.
3. Ahmed, M., Mahmood, A.N. and Islam, Md.R. (2016) “A survey of anomaly detection techniques in financial domain,” *Future Generation Computer Systems*, 55, pp. 278–288. Available at: <https://doi.org/10.1016/j.future.2015.01.001>.
4. Albashrawi, M. (2016) “Detecting Financial Fraud Using Data Mining Techniques: A Decade Review from 2004 to 2015,” *Journal of Data Science*, 14(3), pp. 553–569. Available at: <https://doi.org/10.6339/JDS.201607_14(3).0010>.
5. Al-Hashedi, K.G. and Magalingam, P. (2021) “Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019,” *Computer Science Review*, 40, p. 100402. Available at: <https://doi.org/10.1016/j.cosrev.2021.100402>.
6. Ali, A. *et al.* (2022) “Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review,” *Applied Sciences*, 12(19), p. 9637. Available at: <https://doi.org/10.3390/app12199637>.
7. Aljunaid, S.K. *et al.* (2025) “Secure and Transparent Banking: Explainable AI-Driven Federated Learning Model for Financial Fraud Detection,” *Journal of Risk and Financial Management*, 18(4), p. 179. Available at: <https://doi.org/10.3390/jrfm18040179>.
8. Amrutha, E., Arivazhagan, S. and Jebarani, W.S.L. (2023) “Deep Clustering Network for Steganographer Detection Using Latent Features Extracted from a Novel Convolutional Autoencoder,” *Neural Processing Letters*, 55(3), pp. 2953–2964. Available at: <https://doi.org/10.1007/s11063-022-10992-6>.
9. Ashtiani, M.N. and Raahemi, B. (2022) “Intelligent Fraud Detection in Financial Statements Using Machine Learning and Data Mining: A Systematic Literature Review,” *IEEE Access*, 10, pp. 72504–72525. Available at: <https://doi.org/10.1109/ACCESS.2021.3096799>.
10. Awoyemi, J.O., Adetunmbi, A.O. and Oluwadare, S.A. (2017) “Credit card fraud detection using machine learning techniques: A comparative analysis,” in *2017 International Conference on Computing Networking and Informatics (ICCNI)*. *2017 International Conference on Computing Networking and Informatics (ICCNI)*, pp. 1–9. Available at: <https://doi.org/10.1109/ICCNI.2017.8123782>.
11. Baesens, B., Höppner, S. and Verdonck, T. (2021) “Data engineering for fraud detection,” *Decision Support Systems*, 150, p. 113492. Available at: <https://doi.org/10.1016/j.dss.2021.113492>.
12. Bhattacharyya, S. *et al.* (2011) “Data mining for credit card fraud: A comparative study,” *Decision Support Systems*, 50(3), pp. 602–613. Available at: <https://doi.org/10.1016/j.dss.2010.08.008>.
13. Bouchti, A.E. et al. (2017) “Fraud detection in banking using deep reinforcement learning,” in 2017 Seventh International Conference on Innovative Computing Technology (INTECH). 2017 Seventh International Conference on Innovative Computing Technology (INTECH), pp. 58–63. Available at: <https://doi.org/10.1109/INTECH.2017.8102446>.
14. Chen, Z. and Liu, G. (2019) “DenseNet+Inception and Its Application for Electronic Transaction Fraud Detection,” in 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 2551–2558. Available at: <https://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00357>.
15. Cheng, D. *et al.* (2020) “Spatio-Temporal Attention-Based Neural Network for Credit Card Fraud Detection,” *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01), pp. 362–369. Available at: <https://doi.org/10.1609/aaai.v34i01.5371>.
16. Choi, D. and Lee, K. (2018) “An Artificial Intelligence Approach to Financial Fraud Detection under IoT Environment: A Survey and Implementation,” *Security and Communication Networks*, 2018(1), p. 5483472. Available at: <https://doi.org/10.1155/2018/5483472>.
17. Correa Bahnsen, A., Aouada, D. and Ottersten, B. (2015) “Example-dependent cost-sensitive decision trees,” *Expert Systems with Applications*, 42(19), pp. 6609–6619. Available at: <https://doi.org/10.1016/j.eswa.2015.04.042>.
18. Dal Pozzolo, A. *et al.* (2018) “Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy,” *IEEE Transactions on Neural Networks and Learning Systems*, 29(8), pp. 3784–3797. Available at: <https://doi.org/10.1109/TNNLS.2017.2736643>.
19. Damian Chukwujekwu Ikemefuna *et al.* (2024) *Adaptive Fraud Detection Systems:Using Machine Learning To Identify and Respond To Evolving Financial Threat*, *ResearchGate*. Available at: <https://doi.org/10.56726/IRJMETS61738>.
20. Dang, T.K. *et al.* (2021) “Machine Learning Based on Resampling Approaches and Deep Reinforcement Learning for Credit Card Fraud Detection Systems,” *Applied Sciences*, 11(21), p. 10004. Available at: <https://doi.org/10.3390/app112110004>.
21. Dornadula, V.N. and Geetha, S. (2019) “Credit Card Fraud Detection using Machine Learning Algorithms,” *Procedia Computer Science*, 165, pp. 631–641. Available at: <https://doi.org/10.1016/j.procs.2020.01.057>.
22. García-Ordás, M.T. *et al.* (2023) “Clustering Techniques Selection for a Hybrid Regression Model: A Case Study Based on a Solar Thermal System,” *Cybernetics and Systems*, 54(3), pp. 286–305. Available at: <https://doi.org/10.1080/01969722.2022.2030006>.
23. Hajek, P. and Henriques, R. (2017) “Mining corporate annual reports for intelligent detection of financial statement fraud – A comparative study of machine learning methods,” *Knowledge-Based Systems*, 128, pp. 139–152. Available at: <https://doi.org/10.1016/j.knosys.2017.05.001>.
24. Hernandez Aros, L. *et al.* (2024) “Financial fraud detection through the application of machine learning techniques: a literature review,” *Humanities and Social Sciences Communications*, 11(1), p. 1130. Available at: <https://doi.org/10.1057/s41599-024-03606-0>.
25. Hsin, Y.-Y. *et al.* (2022) “Feature Engineering and Resampling Strategies for Fund Transfer Fraud With Limited Transaction Data and a Time-Inhomogeneous Modi Operandi,” *IEEE Access*, 10, pp. 86101–86116. Available at: <https://doi.org/10.1109/ACCESS.2022.3199425>.
26. Javadpour, A. *et al.* (2024) “A comprehensive survey on cyber deception techniques to improve honeypot performance,” *Computers & Security*, 140, p. 103792. Available at: <https://doi.org/10.1016/j.cose.2024.103792>.
27. Joy Nnenna Okolo *et al.* (2025) “Analyzing the use of machine learning techniques in detecting fraudulent activities,” *World Journal of Advanced Research and Reviews*, 26(1), pp. 1198–1209. Available at: <https://doi.org/10.30574/wjarr.2025.26.1.1097>.
28. Khetani, V. *et al.* (2023) “Cross-Domain Analysis of ML and DL: Evaluating their Impact in Diverse Domains,” *International Journal of Intelligent Systems and Applications in Engineering*, 11(7s), pp. 253–262.
29. Malini, N. and Pushpa, M. (2017) “Analysis on credit card fraud identification techniques based on KNN and outlier detection,” in *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*. *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, pp. 255–258. Available at: <https://doi.org/10.1109/AEEICB.2017.7972424>.
30. Nayan Uchhana *et al.* (2025) “Literature Review of Different Machine Learning Algorithms for Credit Card Fraud Detection,” *ResearchGate* [Preprint]. Available at: <https://doi.org/10.35940/ijitee.C8400.0410621>.
31. Ngai, E.W.T. *et al.* (2011) “The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature,” *Decision Support Systems*, 50(3), pp. 559–569. Available at: <https://doi.org/10.1016/j.dss.2010.08.006>.
32. Nicholls, J., Kuppa, A. and Le-Khac, N.-A. (2021) “Financial Cybercrime: A Comprehensive Survey of Deep Learning Approaches to Tackle the Evolving Financial Crime Landscape,” *IEEE Access*, 9, pp. 163965–163986. Available at: <https://doi.org/10.1109/ACCESS.2021.3134076>.
33. Palacio, S.M. (2019) “Abnormal Pattern Prediction: Detecting Fraudulent Insurance Property Claims with Semi-Supervised Machine-Learning | Data Science Journal.” Available at: <https://doi.org/10.5334/dsj-2019-035>.
34. Raghavan, P. and Gayar, N.E. (2019) “Fraud Detection using Machine Learning and Deep Learning,” in 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE). 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), pp. 334–339. Available at: <https://doi.org/10.1109/ICCIKE47802.2019.9004231>.
35. Ramachandran, K. et al. (2023) “FraudAmmo: Large Scale Synthetic Transactional Dataset for Payment Fraud Detection,” in 2023 International Joint Conference on Neural Networks (IJCNN). 2023 International Joint Conference on Neural Networks (IJCNN), pp. 1–7. Available at: <https://doi.org/10.1109/IJCNN54540.2023.10191990>.
36. Sahin, Y. and Duman, E. (2011) “Detecting credit card fraud by ANN and logistic regression,” in 2011 International Symposium on Innovations in Intelligent Systems and Applications. 2011 International Symposium on Innovations in Intelligent Systems and Applications, pp. 315–319. Available at: <https://doi.org/10.1109/INISTA.2011.5946108>.
37. Shahana, T., Lavanya, V. and Bhat, A.R. (2023) “State of the art in financial statement fraud detection: A systematic review,” *Technological Forecasting and Social Change*, 192, p. 122527. Available at: <https://doi.org/10.1016/j.techfore.2023.122527>.
38. Sofia Patel *et al.* (2025) *Leveraging Reinforcement Learning for Real-Time Fraud Detection in Financial Transactions*, *ResearchGate*. Available at: <https://doi.org/10.13140/RG.2.2.12519.15525>.
39. Song, R. *et al.* (2020) “Fraud Detection of Bulk Cargo Theft in Port Using Bayesian Network Models,” *Applied Sciences*, 10(3), p. 1056. Available at: <https://doi.org/10.3390/app10031056>.
40. Thennakoon, A. et al. (2019) “Real-time Credit Card Fraud Detection Using Machine Learning,” in 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence). 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 488–493. Available at: <https://doi.org/10.1109/CONFLUENCE.2019.8776942>.
41. Ti, Y.-W. *et al.* (2022) “Feature generation and contribution comparison for electronic fraud detection,” *Scientific Reports*, 12(1), p. 18042. Available at: <https://doi.org/10.1038/s41598-022-22130-2>.
42. Tiwari, P. *et al.* (2021) “Credit Card Fraud Detection using Machine Learning: A Study.” arXiv. Available at: <https://doi.org/10.48550/arXiv.2108.10005>.
43. Varmedja, D. *et al.* (2019) “Credit Card Fraud Detection - Machine Learning methods,” in *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*. *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pp. 1–5. Available at: <https://doi.org/10.1109/INFOTEH.2019.8717766>.
44. Waleed Hilal, S. Andrew Gadsden, and John Yawney (2021) “Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances | Request PDF,” *ResearchGate* [Preprint]. Available at: <https://doi.org/10.1016/j.eswa.2021.116429>.
45. Whitrow, C. *et al.* (2009) “Transaction aggregation as a strategy for credit card fraud detection,” *Data Mining and Knowledge Discovery*, 18(1), pp. 30–55. Available at: <https://doi.org/10.1007/s10618-008-0116-z>.
46. Xie, Y. et al. (2019) “A Feature Extraction Method for Credit Card Fraud Detection,” in 2019 2nd International Conference on Intelligent Autonomous Systems (ICoIAS). 2019 2nd International Conference on Intelligent Autonomous Systems (ICoIAS), pp. 70–75. Available at: <https://doi.org/10.1109/ICoIAS.2019.00019>.
47. Zhu, X. *et al.* (2021) “Intelligent financial fraud detection practices in post-pandemic era,” *The Innovation*, 2(4). Available at: <https://doi.org/10.1016/j.xinn.2021.100176>.