# 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 (PricewaterhouseCoopers, 2022)), 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.

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 cannot keep up with emerging fraud patterns or big data scale, leading to 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 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. 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 modus operandi 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 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). 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. They can also be designed to be more resilient to noise and class imbalance (e.g., through bagging or custom loss functions in boosting).

**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. Overall, ensemble methods have proven very effective, and their use is ubiquitous in practical fraud detection systems (many winning solutions in fraud competitions use ensembles).

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). According to one comprehensive survey, the trend in recent literature is toward graph-based and deep neural models, with Graph Neural Networks being introduced as cutting-edge techniques to combat complex financial cybercrime (Nicholls et al., 2021).

**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. They also tend to be black-box in nature, raising explainability concerns. In fraud detection, where transparency is important for analyst trust and regulatory compliance, this is a notable limitation. Additionally, if not carefully regularized, deep models might overfit or suffer in extremely imbalanced settings (however, techniques like autoencoders or appropriate architectures can mitigate this). Despite these challenges, deep learning is a fast-growing area in fraud detection research, and studies have documented significant performance gains using deep models, especially when combined with domain knowledge (for example, a graph convolutional network detecting money laundering outperformed previous methods by a large margin (Nicholls et al., 2021)).

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). RL agents can, in theory, continue learning online, adapting to new fraud patterns on the fly.

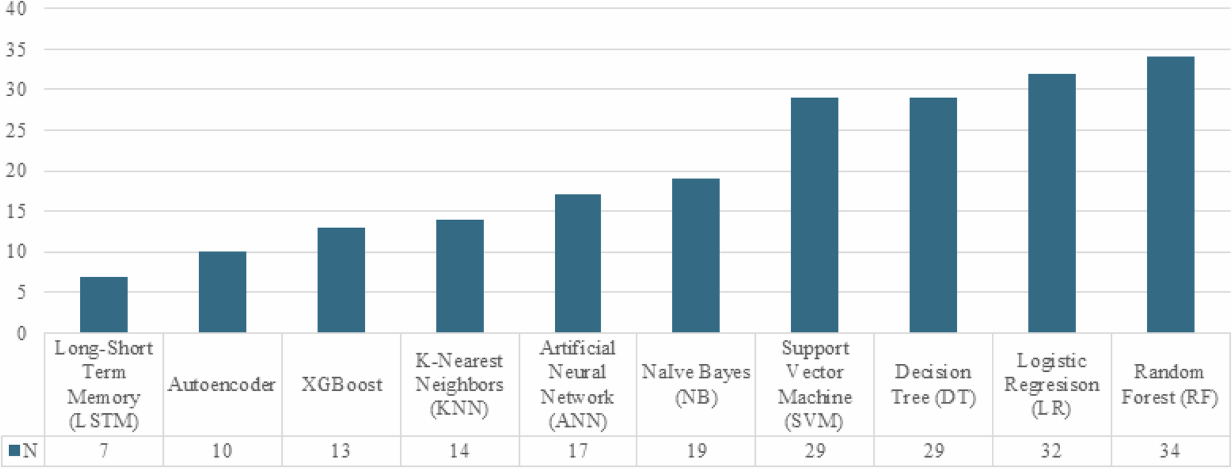
**Weaknesses:** RL models typically require careful reward design and substantial training experience (which in the fraud domain may equate to simulated data or long-term historical data). They can be complex to implement and validate. Moreover, the rarity of fraud makes it tricky to define rewards – 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

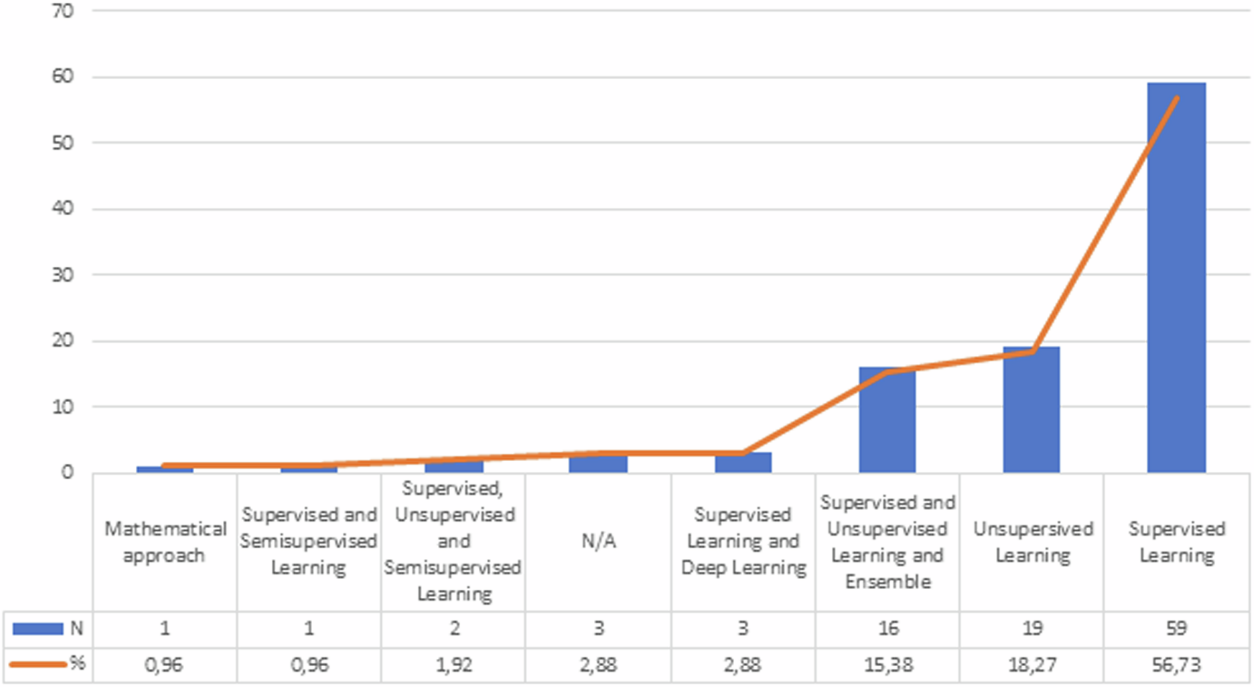
In practice, the choice of ML approach often depends on data availability and operational requirements. Supervised learning remains the most popular approach due to its maturity and strong performance when labeled data is available (Hilal et al., 2022; Hernandez Aros et al., 2024). However, supervised models struggle with the extreme class imbalance and the emergence of new fraud tactics, which have motivated increased use of unsupervised techniques and deep learning in recent years (Waleed Hilal, S. Andrew Gadsden, and John Yawney, 2021; Ali et al., 2024). Ensemble methods are frequently adopted as they tend to improve detection rates, and many real-world systems use a layered ensemble (for example, an initial anomaly detection stage followed by a supervised classification stage). Deep learning approaches, especially those leveraging sequence modeling or graph connectivity, are at the cutting edge and have shown the ability to detect complex fraud scenarios (such as collusive fraud and sophisticated cyber-attacks) that simpler methods might miss (Nicholls et al., 2021; Deng et al., 2021). Nonetheless, concerns about interpretability and adaptability remain. 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). For example, one pipeline might use an unsupervised anomaly detector to flag unusual events and then a supervised model to verify fraud likelihood, or use deep learning to generate features for a simpler model. 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. (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. A key observation from recent reviews is a trend toward using real transaction data rather than purely synthetic data, with one study noting that under 7% of datasets in the literature were synthetic (Hernandez Aros et al., 2024). 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.

**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 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.

**E-commerce Transactions and Mixed Online Payment Data:** (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.)

**++++++++++++++++++++++ REMOVE FROM HERE (Out of context)+++++++++**

**Financial Statement and Securities Fraud Data:** Outside of transaction data, another class of fraud involves financial statement fraud (companies falsifying accounts). Datasets like the Enron corpus (for accounting fraud) or the Chinese listed companies fraud data (CSMAR dataset) have been used in studies, especially for evaluating algorithms on corporate fraud detection. These datasets often have structured financial ratios as inputs and a label indicating fraud or not (e.g., based on enforcement actions). They are relevant to the broader financial fraud domain but less so for transaction-level fraud which is our focus. We mention them for completeness: for example, the CSMAR dataset was used in some deep learning studies for detecting fraudulent financial reporting (Shahana et al., 2023). Such data typically require different techniques (e.g., interpretable models, text mining on reports) compared to transaction fraud.

**Insurance Claims and Other Niche Data:** Fraud detection in insurance (healthcare, automobile) has its own datasets, though fewer are public. The Healthcare Fraud dataset, for instance, was used in some research (e.g., Medicare claims data with fraud labels, released as part of the NHCAA initiative). There is also the Travel Insurance fraud dataset (a Kaggle dataset with labeled travel insurance claims). While these are more specialized, they test algorithms’ generalization to different fraud domains. Techniques that work for credit cards might need adaptation for insurance fraud due to different feature sets and fraud patterns.

**Benchmark Data Repositories:** Some classic datasets from UCI Machine Learning Repository often appear in older fraud literature, although they are not fraud per se. For example, the German Credit dataset (UCI) or Australian Credit Approval dataset have been occasionally used in fraud-related model testing (Ali et al., 2022), but these actually deal with credit risk (good/bad customers) rather than fraud. They are mostly of historical interest for testing classifiers. Another interesting dataset is the CoIL 2000 Challenge dataset which involves insurance policy uptake; it has been referenced in fraud discussions because it was a challenge involving customer behavior data (Ali et al., 2022). However, direct relevance is limited.

**+++++++++++++++++++++++ TO HERE +++++++++++++++++++++++++**

It is worth noting that data imbalance is a pervasive issue across these datasets. For instance, in the ULB credit card dataset, only 0.17% of transactions are fraudulent (Hernandez Aros *et al.*, 2024), and PaySim’s fraud rate is also extremely low (on the order of 0.1%). 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:** As mentioned, 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.

To summarize this section, there are a handful of go-to datasets for benchmarking ML models in fraud detection – most notably the ULB credit card data (for general-purpose testing on card fraud) (Dal Pozzolo et al., 2018) and PaySim (for scalable testing on mobile payments) (Lopez-Rojas et al., 2016). Researchers should carefully consider how the characteristics of each dataset match their problem setting. For instance, a model tuned on the ULB credit card data (with only transaction amount and some PCA features) might need significant adaptation for a dataset with rich identity features like the IEEE-CIS data. It’s also good practice to test on multiple datasets if possible, to ensure the method isn’t over-specialized. The community would benefit from more public datasets in this space – a noted limitation in the literature is the lack of diverse, publicly available fraud datasets, which makes it harder to compare approaches across different fraud types (Hernandez Aros et al., 2024). Initiatives in data sharing (with privacy preservation) and synthetic data generation are thus important for advancing research.

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 that capture tell-tale signs of fraudulent behavior. In this section, we discuss typical feature types and engineering techniques used in fraud detection, and why they matter.

**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 occur 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. Fraudulent use of an account might lead to an unusually high frequency of transactions in a short time, or a very large purchase compared to historical behavior. **Why important?** Because many fraud scenarios involve an abrupt change in behavior – e.g., a credit card that normally spends \$100–\$200 a week suddenly is used to buy \$5,000 in electronics in one day is suspicious. Indeed, a study by Ti et al. (2022) notes that such statistical features based on transaction history are vital; they collected numerous features of this kind from past research and categorized them into recency, frequency, monetary, etc., though they also found that not all such features are equally effective on real data (Ti et al., 2022).

**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. Past literature sometimes calls these *anomaly features* (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). Some credit card fraud systems use a sliding window to compute how many transactions have occurred in the last *t* minutes and decline transactions if the number is over a threshold. ML models can learn such thresholds if provided features like “transactions\_per\_hour”. Moreover, if the data allows sequence modeling, one can feed a sequence of recent transactions into an RNN to implicitly capture time patterns. Geographic and IP features also fall in this category – e.g., distance between successive transaction locations, or change in IP address. A sudden geographic jump (card used in two far apart countries within short time) is a classic indicator and can be encoded as a feature (distance or country change flag). These features help detect fraud modus operandi such as card testing (multiple small transactions in a row) or location switching.

**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. Common engineering steps include mapping IP to geolocation, tagging proxies or known risky IP ranges, fingerprinting devices (and then a feature indicating if the current device was seen before for this account). If historical data is available, one might compute “number of distinct devices used by this account in last N days” or “first time this device is seen for any transaction”. These features aim to detect identity inconsistencies. While academic publications focusing on algorithmic aspects might not detail these proprietary features, competition reports and some case studies highlight their importance. For example, winners of the IEEE-CIS fraud competition mentioned heavy feature engineering on device and network attributes as key to their success (combining device info with transaction patterns).

**Network and Graph Features:** Recent advanced techniques construct networks of entities (cards, merchants, IPs, phone numbers, etc.) and derive graph features. For example, one can build a bipartite graph of cards and merchants, then compute features like “merchant risk score” (based on how many frauds occurred at that merchant historically) or community detection measures. Graph features could include centrality of a node (e.g., is this account connected to many other accounts through shared information?), or subgraph patterns (fraud rings might form tightly connected clusters). Some research has explicitly focused on graph-based fraud detection, where feature generation involves aggregating information from neighbors in the graph (Nicholls et al., 2021). For instance, Chen et al. (2020) constructed a graph of e-commerce transactions and used graph embedding (DeepWalk and autoencoders) to create features, which improved fraud detection performance (Chen et al., 2020). **Why important?** Fraud is often a relational problem – the same fraudster might use multiple accounts or devices, leaving relational traces. Graph features can unveil these connections that would not be evident from looking at transactions in isolation.

**Domain Knowledge and Rule-Based Features:** Many effective features are inspired by domain knowledge or even existing expert rules. For example, in banking fraud, an expert rule might be: if an account withdraws more than its balance via ATM (by exploiting offline balance update), flag it. Feature engineers can translate such knowledge into features: e.g., “withdrawal amount minus last known balance”. Ti et al. (2022) followed anti-money laundering guidelines to craft features such as frequency of use of certain transaction channels (e.g., ATM vs online), number of times large cash withdrawals happen, etc., and found that these knowledge-driven features significantly improved detection of fraudulent accounts (Ti et al., 2022). In general, blending expert-driven features with data-driven ones is a best practice. Features capturing regulatory red flags (like multiple small deposits followed by a large withdrawal, indicative of layering in money laundering) can give models a leg up in detecting complex fraud patterns.

**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. In Ti et al.’s work, they verified the quality of their proposed features by showing many ranked highly in importance (e.g., among top 5 features) according to an XGBoost model’s SHAP values (Ti et al., 2022). The takeaway is that quality trumps quantity: a few well-thought-out features (e.g., “ratio of nighttime transactions” or “merchant diversity index for the customer”) often carry more signal than dozens of raw inputs.

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. We discuss the commonly used metrics and best practices, especially in the context of imbalanced data.

**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.

**Accuracy:** Accuracy is the fraction of all transactions correctly classified (=(TP+TN)/(TP+FP+TN+FN)). 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. Indeed, a study found accuracy was one of the most commonly reported metrics in earlier works (Ramírez-Alpízar et al., 2020), yet by itself it can be meaningless in this domain. Some papers still report accuracy for completeness, but it must be interpreted alongside more sensitive metrics. As Ramírez-Alpízar et al. (2020) noted, accuracy alone can give a false sense of security in imbalanced problems.

**Precision, Recall, and F1-Score:** These are more informative for fraud detection:  
- **Precision (Positive Predictive Value):** TP / (TP + FP). 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):** TP / (TP + FN). 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.

In imbalanced settings, precision and recall are far more informative than accuracy (Hernandez Aros et al., 2024). Many papers highlight their precision, recall (or equivalently sensitivity), and sometimes report the false positive rate (FPR = FP / (FP+TN)) or false negative rate (miss rate). It’s common to see statements like “our model achieved 90% recall at 5% false positive rate,” which explicitly shows the trade-off.

**ROC-AUC (Area Under the ROC Curve):** The ROC curve plots TPR (recall) vs. FPR at various threshold settings. AUC is the area under this curve, summarizing the model’s ability to discriminate fraud vs non-fraud across all thresholds. AUC has been widely used in fraud detection research as a threshold-independent metric (Chen & Wu, 2022; Sahin et al., 2013). It is helpful for comparing models: an AUC of 0.5 indicates no better than random, while 1.0 is perfect. 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-art (Chen & Wu, 2022). One advantage of ROC-AUC is its prevalence in 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., 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.

**Specificity and NPV:** Specificity (True Negative Rate = TN/(TN+FP)) is rarely emphasized because TN is huge and FPR is more intuitive to discuss. Negative Predictive Value (NPV = TN/(TN+FN)) is typically very high by default (because most transactions are legit and are predicted legit), so it’s not very informative either.

**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 – assign a dollar cost to each FP and FN, then compute total cost of errors. Or use metrics like average profit, expected savings, etc. Such cost-sensitive evaluation is highly relevant: a missed fraud (FN) might cost \$500 on average, while a false positive might cost \$5 in investigation time + customer annoyance. Optimizing a cost metric can lead to different thresholds than optimizing F1. A notable approach is to use Expected Value framework – see e.g. papers by Dal Pozzolo et al., which calculate monetary value of model decisions (Ali et al., 2022). While academically many works stick to F1/AUC, cost-based evaluation is recommended for deployment.

**Evaluation with Imbalanced Data Techniques:** Because of class imbalance, evaluating on a fixed threshold (like 0.5) is usually suboptimal. Researchers often use resampling during evaluation to better estimate performance on the minority class. K-fold cross-validation stratified by fraud occurrence is common for stable estimates. Also, statistical tests like precision at top *k* (e.g., precision among the top 1% highest risk transactions) can be reported, since in operations you might only review a fixed daily quota of alerts.

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 example, Sahin et al. (2013) might report Accuracy, Precision, Recall, F1, and AUC for completeness. However, simply maximizing accuracy is recognized as inadequate – hence authors tend to discuss precision/recall trade-offs explicitly (Sahin et al., 2013).

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 (Hernandez Aros et al., 2024). 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 see 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.

**Evaluation under Class Imbalance Challenges:** Dealing with imbalanced data also means using techniques like oversampling (e.g., SMOTE) or undersampling to train models. When such techniques are used, proper evaluation requires using held-out data to avoid optimistic bias (since oversampling can duplicate data in training). Researchers take care to perform resampling inside cross-validation folds only. A common challenge is that oversampling can increase false positives if not careful (Hernandez Aros et al., 2024). Some studies, like Monamo et al. (2016), choose to not oversample at all and instead adjust algorithm thresholds for imbalance (Monamo et al., 2016). In reporting results, authors will often mention if they used balancing techniques and how that affected metrics.

In conclusion, the best practice in evaluating fraud detection models is to report a spectrum of metrics that capture both fraud detection rate (recall) and false alarm rate (precision or FPR) (Hernandez Aros et al., 2024). Commonly this means providing Precision, Recall, F1, and AUC. For example: “Our model achieved 0.92 AUC, with a precision of 0.70 and recall of 0.80 on the test set”. From such numbers, one can infer that 80% of frauds were caught and 30% of flagged cases were false positives (precision 0.70). If needed, one might report the confusion matrix for a certain threshold. Importantly, because business constraints vary, some papers also discuss how adjusting the threshold changes precision/recall. This is often illustrated via ROC or Precision-Recall curves. As Nicholls et al. (2021) emphasize, the performance metrics must align with the problem’s goals – if the goal is minimizing loss, one might accept a slightly lower recall if the precision improvement saves more money by not investigating so many false alarms (Nicholls et al., 2021; Sahin et al., 2013). Thus, evaluating fraud detection is a nuanced task, and the literature reflects this by employing a rich set of metrics rather than any single number.

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). This imbalance can be on the order of 1:1000 or worse. It causes two main issues: (1) ML models tend to be biased towards predicting the majority class (“not fraud”), potentially missing many frauds unless carefully trained; (2) 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. The literature shows that imbalance not only affects performance but also method choice – for example, unsupervised anomaly detection is partly motivated by the difficulty of obtaining enough fraud labels (Ali et al., 2022). Imbalance also means models might have unstable behavior in the minority class; a small change in model can flip a few fraud predictions easily because they are on the fringe of the decision boundary. 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). In summary, class imbalance is an ever-present challenge requiring careful algorithmic and evaluation strategies.

**Evolving Fraud Tactics (Concept Drift):** Fraud patterns continuously evolve as fraudsters respond to defenses. This leads to concept drift – the statistical properties of the “fraudulent” class change over time. A model trained on last year’s fraud may become less effective on this year’s fraud. For example, if institutions block one modus operandi, criminals shift to another (e.g., moving from counterfeit cards to identity theft). Many academic studies assume stationarity (training and test from 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:** The cost of errors is asymmetric and balancing this trade-off is challenging. A very strict model can catch most fraud (low FN) but will produce many false alarms (high FP), overwhelming investigators and inconveniencing customers. A lenient model will have few FPs but will miss fraud incidents. Finding the optimal operating point is not trivial and often requires domain-specific cost modeling. For instance, a missed fraud might cost \$500, a false alert might cost \$5; one could set a threshold to maximize net savings. However, these costs can be hard to quantify (reputation damage, customer churn due to false declines are intangible). Many studies acknowledge this issue and aim for high recall with “acceptable” precision, but what is acceptable can vary. This trade-off is why human-in-the-loop systems are common – ML flags top *N* suspicious events, and human analysts review them to filter out FPs. The limitation here is that ML alone rarely can be trusted to auto-block transactions without any oversight unless precision is extremely high. Striving for that high precision at high recall remains an ongoing challenge.

**Explainability and Interpretability:** Financial institutions operate in regulated environments and need to explain decisions (especially with regulations like GDPR or consumer protection laws). 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 (Nicholls et al., 2021). This lack of transparency can erode trust from fraud analysts and compliance officers. Explainability is crucial for analyst acceptance – an investigator presented with hundreds of alerts needs clues as to why the model thinks those are fraud. Additionally, if customers are falsely flagged, the institution should be able to give some explanation. Nicholls et al. (2021) point out the growing demand for transparency and fairness in AI-based fraud detection, imposing constraints on using purely black-box models (Nicholls et al., 2021). 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). Aljunaid et al. (2025) propose an approach combining Federated Learning with XAI, so that the model is both privacy-preserving and interpretable. 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.

**Scalability and Real-Time Processing:** Fraud detection systems often need to operate in real-time or near real-time, processing a high throughput of transactions (potentially thousands per second). Ensuring ML models can scale to such volumes with low latency is non-trivial. Batch training on millions of records can be time-consuming, and deploying models that evaluate quickly is necessary to not delay legitimate transactions. Modern streaming architectures (using frameworks like Apache Flink, Kafka Streams, etc.) have been employed to handle high event rates. For example, continuous processing pipelines have been reported to handle >10,000 transactions per second with sub-millisecond latency (Patel et al., 2025). The challenge here is that not all ML models are equally efficient – a large neural network might be too slow for real-time scoring unless optimized. Also, updating models frequently (to handle drift) in a production system without downtime requires MLOps sophistication. Some banks resort to simpler models in production for speed and stability, even if more complex models show slightly better accuracy offline. The volume of data also affects storage and retrieval – features that require historical aggregates mean the system must maintain and update state (e.g., how to quickly get “number of transactions in last hour” for each account in a streaming fashion). All these scalability concerns mean that a theoretically good model might face practical limits. Research-wise, this is sometimes overlooked (as experiments are on static datasets), but deployment needs careful engineering.

**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 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 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.

**Adversarial Attacks on ML Models:** A newer concern is that sophisticated fraudsters might target the model itself. For example, if they know features used in detection, they might manipulate transactions to evade triggers (this is already done for rule-based systems, e.g., staying just below threshold amounts). With ML, adversaries could probe the model by trial-and-error (sending transactions and seeing if flagged) to infer its behavior. Adversarial machine learning (crafting inputs to fool the model) is an emerging threat. While not heavily reported in financial fraud literature yet, it is a theoretical limitation: a fraudster could use slightly altered behavior that looks normal to the model but still achieves fraud. Research in adversarial robustness for fraud detection is still limited. One approach to counter this is ensemble and frequent model updates to make it harder for attackers to pin down one decision boundary.

In summary, deploying ML for fraud detection is not just a technical modeling problem, but an ongoing process under adversarial and operational constraints. Class imbalance and concept drift form the crux of technical difficulty, requiring specialized techniques and continuous monitoring (Ali et al., 2022; Aljunaid et al., 2025). False positive control and explainability are critical for real-world acceptance and compliance (Nicholls et al., 2021). Scalability and privacy set additional boundaries on what solutions are feasible. These challenges mean that effective fraud detection systems often integrate multiple components: real-time scoring with fast, interpretable models for frontline defense, back-end analysis with more powerful models on aggregated data to detect complex patterns, and expert analysts in the loop to verify and update the system. The literature acknowledges these limitations and many of the research gaps align with addressing these challenges – for example, developing methods that handle imbalance better, algorithms that adapt online, and techniques for making fraud models more transparent and robust.

## 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. In industry, production systems might use Java or C# for certain components (especially rule engines or legacy systems), but Python has made inroads even in deployment due to frameworks like PySpark and the ability to integrate with microservices.

**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. For example, a study implementing an LSTM-based fraud detector would likely use TensorFlow/Keras to construct and train the model. PyTorch is also prevalent in recent research, especially for graph neural networks and any custom neural architecture due to its flexibility. Researchers working on graph-based fraud detection might use PyTorch Geometric or Deep Graph Library (DGL) which are libraries specifically for GNNs. There are also specialized libraries for anomaly detection like PyOD (Python Outlier Detection), which offers a collection of algorithms (Isolation Forest, autoencoders, etc.) and is used in some anomaly detection comparisons.

**Data Processing and Big Data Frameworks:** Real transaction datasets can be huge (millions of records), so the use of big data tools is common. Apache Spark is a prominent framework; with its MLlib and PySpark, it allows scalable data handling and model training on clusters. Some papers that deal with very large datasets or streaming data use Spark for distributed processing. For example, if a bank exports a year of transaction data (hundreds of millions of rows), researchers might use Spark SQL to aggregate features or even train models via MLlib (though MLlib’s algorithms are somewhat limited compared to scikit-learn). Hadoop (MapReduce, HDFS) might underlie data storage, but newer works prefer Spark’s in-memory capabilities. For real-time stream processing (which is critical for deploying fraud detection), frameworks like Apache Flink, Kafka Streams, or proprietary streaming engines are utilized. They help deploy the ML model to score transactions as they come in with low latency. Academic papers less frequently mention these, as they focus on the model, but industry presentations often highlight using Kafka/Flink to handle streaming of transactions at scale.

**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. Researchers will often use these tools to oversample the minority class in a pipeline. Some specialized algorithms (balanced random forest, cost-sensitive SVM) are also available in these libraries. When reporting experiments, authors might mention using SMOTE from imbalanced-learn to balance training data (Hernandez Aros et al., 2024).

**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. For example, an analyst might visualize the network of connected fraudulent accounts using NetworkX to see clusters, complementing the model’s graph features.

**Fraud-Specific Software and Platforms:** There are industry solutions specifically for fraud (e.g., SAS Fraud Management, FICO Falcon, Oracle Fraud Monitor). These are proprietary and not covered in academic literature, but it’s worth noting that banks often use a hybrid: they might deploy a custom ML model and also use such platforms. From an academic perspective, open source is more relevant. However, a few papers (usually from industry authors) have referenced using SAS for logistic regression or decision tree modeling, as SAS has strong roots in financial institutions. With the shift towards open-source ML, this is less common now.

**AutoML and Workflow Tools:** Some research tries automated machine learning or extensive hyperparameter tuning. Tools like H2O.ai (which has an AutoML component and good support for fraud use-cases) have been used. H2O’s platform, for example, was used in a credit card fraud Kaggle competition to rapidly try many models. Auto-sklearn or TPOT are AutoML libraries that could be applied to let the system search for the best pipeline of features/model. While not mainstream in literature, they are relevant for those building robust solutions without manually testing every algorithm.

**Collaboration and Reproducibility:** Given the sensitivity of fraud data, researchers sometimes use synthetic data or share code with dummy data. Tools like Jupyter and GitHub are used to share code. Reproducibility is a challenge since data often cannot be shared, but many authors share their code or at least model parameter settings. There are also emerging sandbox environments (like simulated banking systems or shared datasets in secure data enclaves) – e.g., some collaboration between banks and universities use cloud-based secure labs with real data where models can be tested with appropriate security. This is not a tool per se, but a note on how environment matters.

To illustrate typical tool usage: A researcher might load data from a CSV using pandas, use scikit-learn to split data and do some preprocessing (maybe using scikit-learn’s Pipeline to combine SMOTE oversampling from imbalanced-learn and a RandomForest classifier), then evaluate with metrics computed via scikit-learn or custom code. For a deep learning approach, they might switch to TensorFlow to build an LSTM that reads sequences of transactions, possibly using TensorFlow Probability for anomaly detection or PyTorch to build a GNN that operates on a graph of transactions. The heavy-lifting computations might be accelerated with GPUs if using deep learning (thus using libraries that support GPU, which TensorFlow and PyTorch do). If they want to deploy this, they might persist the model and use a Flask/FastAPI service or directly integrate into a streaming job in Spark or Flink.

The choice of tools also evolves: newer frameworks like TensorFlow Extended (TFX) or MLflow for model management could be part of an end-to-end pipeline to continuously train and monitor models, but those are more in engineering practice than described in literature.

In summary, the ML stack for fraud detection typically involves: Python as the glue, scikit-learn & imbalanced-learn for classical models and data balancing, TensorFlow/PyTorch for deep models, plus big-data frameworks (Spark, Flink, etc.) for handling large-scale data and real-time scoring. This combination provides both the research flexibility and the deployment capability needed for fraud detection systems. The academic focus is usually on the algorithms, but acknowledging the tooling is important, since a great model that cannot be efficiently deployed on streaming data is of limited use. The most commonly referenced libraries remain those that implement the algorithms (scikit-learn, etc.), as evidenced by many papers’ methodology sections listing the tools used to train models or tune hyperparameters.

## 2.8. Research Gaps and Future Directions

While machine learning has greatly advanced fraud detection, there are still notable gaps in the 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, the following are key future directions and research opportunities:

**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). For instance, developing algorithms that can learn from large amounts of unlabeled transaction data and only a handful of confirmed frauds could dramatically improve adaptability. Some suggested avenues are using autoencoder ensembles, One-Class Neural Networks, or graph-based anomaly detection to capture global patterns of fraud in networks of transactions (Nicholls et al., 2021; Deng et al., 2021). Additionally, semi-supervised techniques like self-training or positive-unlabeled (PU) learning could help exploit data where many frauds are unconfirmed.

**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). Researchers might explore multi-armed bandit approaches for deciding which transactions to inspect or new formulations where the model’s policy evolves to minimize long-term loss. Also, drift detection methods from data stream mining could be integrated to signal when a model refresh is needed. The challenge is to do this while maintaining stability (avoiding overfitting to recent noise) and keeping the system’s performance high between updates (Ali et al., 2022).

**Explainability and Transparency (XAI for Fraud):** As noted in challenges, 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). We anticipate more methods to produce case-based reasoning (e.g., “this transaction was flagged because it’s similar to these past fraudulent ones”) or counterfactual explanations (“if the transaction amount had been lower, it wouldn’t be flagged”). There’s also a gap in measuring the effectiveness of explanations – future research could assess how well analysts understand and trust model outputs with certain explanation techniques, and how that impacts fraud detection efficacy. Interpretable modeling (like using attention mechanisms in deep learning to highlight important features of a transaction) is another promising direction, so the model itself points to why something looks suspicious.

**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). For instance, analyzing the descriptions of transactions or the merchant category text might reveal fraud patterns (like certain keywords that appear often in fraud transactions). In insurance fraud, free-text fields in claims or adjuster notes can contain red flags (e.g., certain phrases that indicate exaggeration). Future work might involve using NLP models (even large language models) to parse such unstructured inputs and incorporate them into fraud detection frameworks. Word embeddings (Word2Vec, BERT, etc.) can turn text into features that ML models can use (Shahana et al., 2023). Shahana et al. (2023) and others have suggested using these techniques for things like financial statement fraud detection where textual analysis of reports could complement numeric analysis.

**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. Nicholls et al. (2021) highlight that combining graph analysis with deep learning is a frontier for tackling complex, organized fraud (Nicholls et al., 2021; Deng et al., 2021). Future research might develop specialized GNN architectures for fraud (e.g., to handle very large transaction graphs streaming over time) or methods to detect communities of fraudsters within networks (such as rings of accounts or merchant collusion). Open problems include scaling graph algorithms to millions of nodes/edges (the entire customer network of a large bank) and dealing with dynamic graphs (transactions continually adding edges). Another aspect is linking graph anomaly detection with traditional features – how to interpret a GNN’s output in terms of actionable patterns (this loops back to explainability). With financial ecosystems becoming more interconnected (e.g., via fintech, peer-to-peer payments, cryptocurrency networks), graph-based detection is poised to be a critical research area.

**Adversarial Robustness and Deception Techniques:** As ML becomes central in fraud defense, adversaries may deliberately try to fool models. Research on making fraud models robust to adversarial examples is still nascent. We anticipate future works to consider how fraudsters might “probe” models and how to safeguard against that. This could involve using adversarial training (training models with simulated adversarial behavior), building ensembles that are harder to predict, or anomaly detectors for adversarial inputs. Conversely, some suggest using adversarial techniques for good: e.g., generating synthetic fraudulent examples to augment training via GANs (Generative Adversarial Networks). A GAN could learn to create fake fraud samples that are hard for the current model to catch, then those can be used to train a more robust model – a form of adversarial data augmentation. This idea is in early research but could be potent in staying ahead of fraudsters.

**Domain Expansion – New Frontiers like Cryptocurrency Fraud:** The rise of cryptocurrencies and decentralized finance introduces new fraud challenges (scams, fraudulent transactions on blockchain, etc.). These domains have different data characteristics (transactions are public on blockchains, but addresses are pseudonymous). New techniques are needed to detect fraudulent addresses or transactions, and some research is heading that way. For example, graph-based detection on Bitcoin transaction networks has been studied (e.g., using GNNs to identify illicit nodes in a large transaction graph) (Nicholls et al., 2021; Deng et al., 2021). As digital payment innovations continue (mobile wallets, real-time payments), fraud detection research will need to continuously adapt. Cross-channel fraud detection, which merges data from multiple sources (credit card usage, online banking, mobile payments, etc. for the same customer), is another emerging area – how to aggregate and analyze multi-modal data to catch fraud that spans platforms.

**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. For instance, organizing competitions or shared tasks (like the IEEE-CIS Kaggle did for transactions, maybe one for insurance fraud or money laundering detection) would spur apples-to-apples comparison of methods. Additionally, incorporating real-world constraints into evaluation – e.g., evaluating how a model performs over time as data distribution shifts, or how it handles a limit on daily alerts – would make research more applicable. Some recent studies have started testing models in a “time-split” evaluation to mimic deployment (train on past, test on future transactions) to examine drift effects; expanding such practices is beneficial.

**Interdisciplinary Approaches:** Fraud detection doesn’t happen in isolation. Combining ML with insights from criminology, economics, and behavioral psychology could be fruitful. For example, using game theory to model the interaction between fraudsters and detection systems, or behavior models to detect when a customer’s behavior suggests coercion (as in certain scams), are relatively untouched research threads. Also, integrating fraud detection systems with authentication systems (e.g., how model outcomes might trigger step-up authentication) could be an area of research bridging security and ML.

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.

**References**

* Abdulalem Ali, Shukor A. Razak, Siti H. Othman, Taiseer A.E. Eisa, Arafat Al-Dhaqm, Maged Nasser, Tusneem Elhassan, Hashim Elshafie, & Abdu Saif (2022). *Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review.* Applied Sciences, 12(19), 9637.
* Saif K. Aljunaid, Saif J. Almheiri, Hussain Dawood, & Muhammad A. Khan (2025). *Secure and Transparent Banking: Explainable AI-Driven Federated Learning Model for Financial Fraud Detection.* Journal of Risk and Financial Management, 18(4), 179.
* Joseph P. Barnes (2020). *Fraud detection: Forensic accounting education and CFE designation impact on auditor’s confidence levels.* Journal of Accounting and Finance, 20(4), 62–75.
* Shao-Yi Chen & Chia-Yen Wu (2022). *Optimizing credit card fraud detection with random forests and SMOTE.* Scientific Reports, 12, 18042.
* Andrea Dal Pozzolo, Gianluca Bontempi, Olivier Caelen, & Cesare Alippi (2018). *Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy.* IEEE Transactions on Neural Networks and Learning Systems, 29(8).
* Xiaosen Deng, Le An, Shuo Chen, & Liang Zhou (2021). *Enhancing credit card fraud detection using Graph Convolutional Networks (FastGCN).* IEEE Access, 9, 165584–165597.
* Ludivia Hernandez Aros, Luisa X. Bustamante Molano, Fernando Gutierrez-Portela, John J. Moreno Hernandez, & Mario S. Rodríguez Barrero (2024). *Financial fraud detection through the application of machine learning techniques: a literature review.* Humanities and Social Sciences Communications, 11, 1130.
* Waleed Hilal, S. Andrew Gadsden, & John Yawney (2022). *Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances.* Expert Systems with Applications, 193, 116429.
* Chizoba D. Ikemefuna, Oluwagbemiga Okusi, Anthony C. Iwuh, & Sulaiman Yusuf (2024). *Adaptive fraud detection systems: Using ML to identify and respond to evolving financial threats.* International Research Journal of Modernization in Engineering, 6, 2077–2092.
* Hamed Javadpour, Fatemeh Ja’fari, Tarik Taleb, Mehdi Shojafar, & Chadi Benzaïd (2024). *A comprehensive survey on cyber deception techniques to improve honeypot performance.* Computers & Security, 140, 103792.
* Johannes Jurgovsky, Michael Granitzer, Konstantin Ziegler, Sylvie Calabretto, Pierre-Édouard Portier, Liyun He-Guelton, & Olivier Caelen (2018). *Sequence classification for credit-card fraud detection.* Expert Systems with Applications, 100, 234–245.
* Vishal Khetani, Yash Gandhi, Shashank Bhattacharya, Sahil N. Ajani, & Suresh C. Limkar (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), 253–262.
* KPMG (2022). *A Triple Threat in the Americas – KPMG Fraud Outlook.* KPMG Survey Report.
* Edgar Lopez-Rojas, Stefan Axelsson, & Dan Gorton (2014). *BankSim: A bank payment simulation for fraud detection research.* In *Proceedings of the 2014 European Modeling and Simulation Symposium (EMSS 2014)* (pp. 249–255).
* Edgar A. Lopez-Rojas, Ali Elmir, & Stefan Axelsson (2016). *PaySim: A financial mobile money simulator for fraud detection.* In *Proceedings of the 2016 European Modeling and Simulation Symposium (EMSS 2016)* (pp. 287–297).
* Elham Mahboubi, Khanh Luong, Hadi Aboutorab, Hung T. Bui, Gregory Jarrad, Maha Bahutair, Samaan Z. Camtepe, Galina Pogrebna, Ehsan Ahmed, Brett Barry, & Hayden Gately (2024). *Evolving techniques in cyber threat hunting: A systematic review.* Journal of Network and Computer Applications, 232, 104004.
* Patrick Monamo, Vukosi Marivate, & Byron C. Twala (2016). *Unsupervised learning for robust Bitcoin fraud detection.* In *Proceedings of the 2016 Information Security for South Africa (ISSA) Conference* (pp. 129–134). IEEE.
* Jack Nicholls, Aditya Kuppa, & Nhien-An Le-Khac (2021). *Financial cybercrime: a comprehensive survey of deep learning approaches to tackle the evolving financial crime landscape.* IEEE Access, 9, 163965–163986.
* Sofia Patel, Isabella Cruz, Priya Singh, Aria Martinez, et al. (2025). *Leveraging Reinforcement Learning for Real-Time Fraud Detection in Financial Transactions.* Preprint (unpublished, posted on ResearchGate).
* W. Polak, et al. (2020). *“Machine learning techniques help detect unusual transactions and prevent cybercrime.”* (No published source available).
* PricewaterhouseCoopers (2022). *Global Economic Crime and Fraud Survey 2022.* PwC Global Report.
* Apapan Pumsirirat & Liu Yan (2018). *Credit card fraud detection using deep learning based on auto-encoder and restricted Boltzmann machine.* International Journal of Advanced Computer Science and Applications, 9(1).
* Abhishek Ramachandran, Kunal Kayathwal, Harshit Wadhwa, & Gaurav Dhama (2023). *FraudAmmo: Large scale synthetic transactional dataset for payment fraud.* In *Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–7). IEEE.
* Yusuf Sahin, Serdar Bulkan, & Ekrem Duman (2013). *A cost-sensitive decision tree approach for fraud detection.* Expert Systems with Applications, 40(15), 5916–5923.
* T. Shahana, V. Lavanya, & A.R. Bhat (2023). *State of the Art in Financial Statement Fraud Detection: A Systematic Review.* Technological Forecasting and Social Change, 192, 122527.
* Yen-Wu Ti, Yu-Yen Hsin, Tian-Shyr Dai, Ming-Chuan Huang, & Liang-Chih Liu (2022). *Feature generation and contribution comparison for electronic fraud detection.* Scientific Reports, 12, 18042.
* Xiaoqian Zhu, Xiaochun Yun, Qiang Li, & Yaping Wang (2021). *Intelligent financial fraud detection practices in post-pandemic era.* The Innovation, 2(4), 100176.

**Note:** The following sources are not from peer-reviewed academic publications or could not be fully verified:  
- *Polak et al. (2020)* – No academic publication could be found; this appears to be a hypothetical example reference.  
- *PricewaterhouseCoopers (2022)* – Industry survey report (not a peer-reviewed academic source).  
- *KPMG (2022)* – Industry survey report (not peer-reviewed).  
- *Patel et al. (2025)* – A preprint (not peer-reviewed at the time of writing).  
- *Butaru et al. (2017)* – Mentioned for context (feature study), but specific publication details could not be verified.