A work project, presented as part of the requirements for the Award of a Master's degree in Business Analytics from the Nova School of Business and Economics.

Account-based Fraud Detection on the XRP Ledger: A Supervised Learning Approach

Leonardo Heinemann

Work project carried out under the supervision of:

Prof. Leid Zejnilovic

17/12/2024

Abstract

XRP is among the top 5 most prevalent cryptocurrencies as of 2024. However, the pseudonymous nature of XRP Ledger (XRPL) transactions poses challenges for understanding economic activities and enforcing regulatory oversight, particularly in detecting fraud. This study trains various supervised machine learning algorithms on an off-chain dataset containing accounts involved in spam, token theft, and Ponzi schemes within the XRPL. The best-performing model, LightGBM, achieved a fraud detection accuracy of over 90%. Additionally, an in-depth analysis of the model's predictions reveals behavioral patterns of fraudulent actors in the network. Based on these insights, the study explores potential applications for a fraud detection model to enhance security and regulatory compliance within the XRPL.

Keywords

Blockchain, Fraud Detection, Ledgerlytics, Machine Learning, Ripple, Supervised, XRP Ledger

This work used infrastructure and resources funded by Fundação para a Ciência e a Tecnologia (UID/ECO/00124/2013, UID/ECO/00124/2019 and Social Sciences DataLab, Project 22209), POR Lisboa (LISBOA-01-0145-FEDER-007722 and Social Sciences DataLab, Project 22209) and POR Norte (Social Sciences DataLab, Project 22209).

1 Introduction

Since its introduction in 2009, Bitcoin has significantly influenced the global monetary system, leading to the emergence of almost ten thousand competing cryptocurrencies on the market (Statista, n.d.; Alahmad et al. 2023). The XRP token, as one of these currencies, was introduced by Ripple Inc. in 2012 and is widely recognized as one of the most prominent alternatives to Bitcoin. XRP ranks in the top 5 digital currencies worldwide with a market capitalization of over 100 billion USD as of November 2024 (Ahmadova and Erek 2022; CoinMarketCap 2024).

The XRP network mainly aims to serve banks and financial institutions by providing a decentralized alternative to traditional cross-border payment systems such as the Society for Worldwide Interbank Financial Telecommunication (SWIFT), offering superior transaction speed and reduced costs for currency exchanges (Sergeenkov 2024). Additionally, the network supports key applications, including tokenization, central bank digital currencies, and stablecoins, as well as Non-Fungible Tokens (Ripple, n.d.-b).

While the XRP Ledger, as the decentralized ledger technology behind its token, operates as a publicly accessible database, the information recorded on-chain alone is insufficient to trace users back to their real-world counterparts. In blockchain, participants transact assets using unique alphanumeric addresses. These addresses lack a direct connection to real-world identities or individual agents, a trait known as pseudonymity (Hellwig, Karlic, and Huchzermeier 2020).

This characteristic inherently results in a lack of transparency and uncertainty regarding the types of activities in blockchain networks such as the XRP Ledger. The limited traceability of transactions raises concerns about the potential misuse of the technology for illicit purposes. Illegal crypto activities such as scams, stolen funds, and illicit trade accounted for a total estimate of 39.6 billion USD in 2022 (Chainalysis 2024). As a result, stakeholders – including governments, industry leaders, and financial institutions – require a shared understanding of the

participants within blockchain ecosystems to foster sustainable growth and address the vulnerabilities inherent to the networks (Chainalysis, n.d.).

Addressing these challenges, this study seeks to analyze agents on the XRP Ledger to detect accounts involved in selected fraud schemes. The main objective of the study is:

1. To assess the ability of supervised machine learning to identify accounts that are involved in fraudulent activity on the XRP Ledger.

To the best of our knowledge, no similar research has been conducted on XRP, leading to uncertainties about the applicability of existing methods to gain insights into the network. While modeling techniques, such as machine learning models have been applied with similar aims to Bitcoin and Ethereum, their relevance and effectiveness on other ecosystems remains unclear, as ledger structure and use cases differ between different ledgers. By addressing this research gap, the primary goal of this work is to test the applicability of the selected research methods on XRP by adapting them to the specific properties of the networks' structure as well as the available off-chain data. The secondary goal is to provide insights into the implications of our results for the XRP ecosystem and potential implementation scenarios for our classifiers.

The relevance to apply this research on the XRP Ledger arises from its increasing importance in financial markets. The lack of studies in this area creates ambiguity regarding the network's usage and techniques to mitigate fraudulent behavior. This presents a challenge for regulators, compliance bodies, investors, financial institutions, and the broader XRP ecosystem in assessing the network's reliability and its participation in economically significant activities within markets. The detection of fraud in the network has the potential to implement effective ecosystem monitoring and demonstrate XRP's utility and diverse applications to stakeholders, including investors, financial institutions, and researchers. The potential of the utilized methods also lies in monitoring regulatory compliance. Ripple can leverage the findings of our work to manage funded partnerships through data-driven decisions. Eventually, our final model is expected to benefit regulatory and prosecutorial bodies by facilitating automated real-time

detection of illicit agents, thereby strengthening compliance and enforcement mechanisms. The trust of users, particularly institutional stakeholders, is expected to depend on XRP being a secure payment environment where fraudulent activities are effectively identified and mitigated. Ultimately, we expect this work to contribute to greater transparency in a financially significant blockchain network, fostering trust, supporting informed decision-making, and enhancing confidence in its integrity.

2 The XRP Ledger

This section introduces the XRP ecosystem as the primary domain of this study. The XRP Ledger (XRPL) is a blockchain solution and distributed ledger technology (DLT) developed and first launched in 2012 by Ripple Inc (Ripple, n.d.-c). A blockchain is a decentralized public ledger that records transactions within a peer-to-peer network. It consists of linked data blocks, each referencing the previous one. The XRP network relies on trusted validator nodes to reach consensus on data validity, enabling secure and integer transactions without a central authority (Ripple 2024b; European Union Agency for Cybersecurity (ENISA), n.d.).

The XRPL leverages its native cryptocurrency, XRP. Unlike Bitcoin (BTC), which requires mining to create new coins, all 100 billion XRP tokens were created at its inception, ensuring a fixed supply from the beginning (Alahmad et al. 2023; Li and Whinston 2020; Ripple 2024b). As of November 2024, XRP's market capitalization exceeded $100 billion, with each token valued at around 1.80 USD (CoinMarketCap 2024). Ripple Inc., is the company and issuer behind the network and token, holding about 48 billion XRP in escrow (2024), with a monthly release of up to 1 billion tokens to stabilize XRP supply and prices (Sergeenkov 2024). The company has partnered with over 200 financial institutions, including major and central banks, to integrate the XRPL network into their systems (Rella 2020; Ahmadova and Erek 2022).

Designed to enable seamless global value transfers, the XRPL primarily functions as a distributed payment system aimed at improving cross-border payment efficiency, offering an alternative to traditional solutions like SWIFT (Chase and MacBrough 2018; Sergeenkov 2024). While widely adopted, SWIFT transactions can be slow and costly due to intermediary fees and the need for pre-funded foreign accounts. In contrast, the XRPL supports rapid, cost-effective transactions across multiple currencies through features like trust lines and an on-chain decentralized exchange (Rella 2020; Peduzzi, James, and Xu 2021). By eliminating intermediary fees and the need for pre-funded foreign accounts, adopting the XRPL can reduce operational costs for banks by up to 60% (Sergeenkov 2024).

Beyond payments, the XRPL network supports key applications, including tokenization, central bank digital currencies (CBDCs), and stablecoins, as well as Non-Fungible tokens (NFTs) (Ripple, n.d.-b). In comparison, other ecosystems like BTC primarily serve as digital assets, while networks such as Ethereum (ETH) focus on decentralized applications powered by smart contracts (Alahmad et al. 2023).

Ultimately, the XRPL also differs from other blockchain technologies in its consensus mechanism and fee system. The XRPL employs the Ripple Protocol Consensus Algorithm (RPCA), where trusted validators from a Unique Node List (UNL) vote on transaction validity. This process confirms transactions within seconds, enabling fast and efficient processing. In comparison, BTC's Proof of Work (PoW) mechanism relies on miners solving complex problems, resulting in an average block time of ten minutes. ETH transitioned from the PoW in 2022, to a less energy-intensive approach called Proof of Stake (PoS) that uses coin ownership for validation (Mauri, Cimato, and Damiani 2018; Alahmad et al. 2023). Thereby, the XRPL outperforms other blockchains in terms of transaction speed and fees, processing around 1,500 transactions per second with costs of approx. 0.00001 XRP (depending on network load) per transaction, which represents only a fraction of a dollar cent. (Mauri, Cimato, and Damiani 2018; Alahmad et al. 2023; Sergeenkov 2024)

The XRPL also differs in its data and ledger structure. It uses an account-based system that directly tracks each account's balance, unlike BTC's Unspent Transaction Output (UTXO) model, where balances are spread across multiple addresses and transactions involve multiple inputs and outputs. This simplifies transaction validation, as XRPL accounts can spend part of their balance while retaining the rest, using a single address repeatedly for both sending and receiving funds. (Mauri, Cimato, and Damiani 2018; Alahmad et al. 2023; Akcora, Gel, and Kantarcioglu 2022)

Eventually, each XRP transaction includes detailed information such as the sender and receiver addresses, transaction amount, fee, transaction type (from over 40 types), and additional metadata such as source tags for user identification, optional memos for supplementary information, and flags or settings related to the transaction. (Ripple, n.d.-a; 2024a)

3 Fraud Detection on DLTs

This section reviews related research on detecting fraudulent accounts and highlights the types of illicit activities that can be identified in DLT ecosystems. While blockchain was initially perceived as secure due to its cryptographic foundations and consensus mechanisms, emerging security concerns have prompted researchers to investigate methods for detecting fraud across various ledgers (Shayegan, n.d.). Therefore, uncovering fraudulent transactions is of significant academic and practical relevance (Sun Yin et al. 2019).

Initially, it is essential to understand the various types of fraud schemes that target blockchain networks, to gain an understanding of illegal activities on DLTs. A prominent example of a fraudulent activity type is *Ponzi* schemes. They represent fraudulent investment strategies where returns for earlier participants are financed using funds from new investors. The organizers usually attract new participants, promising high returns with minimal or no risk

(SEC 2017). An example of a famous cryptocurrency *Ponzi* scheme was *PlusToken*. *PlusToken* was marketed as a wallet service offering users high returns in exchange for purchasing its associated *PLUS* tokens. The fraudsters promised these returns would come from exchange profits, mining, and referrals, ultimately defrauding victims of over 3 billion USD in cryptocurrency (Chainalysis 2019). Similar strategies involve *giveaway* scams. Fraudsters claim to be offering free cryptocurrency, typically asking participants to first send a specific amount to a designated address upfront (Kraken, n.d.).

Market manipulation strategies are also widespread on blockchain networks and include methods like *pump and dump* schemes and *wash trading*. In these schemes, the value of a token is artificially inflated by creating a false sense of interest through artificially increased trading activity, either by rapidly buying and selling an asset or coordinating misleading market signals (Morgia et al. 2024; Victor and Weintraud 2021).

Some other strategies focus on obscuring illicit transactions, like *money mixing,* where transactions are combined to obscure the trails and origins of assets. In other schemes, like *Double spending*, attackers attempt to use the same funds in several transactions, trying to fool the network‘s consensus mechanism. Furthermore, *token theft* schemes, where funds are stolen through user hacking, have already caused significant financial losses in the past, with millions of stolen assets from users every year. (Ul Hassan, Rehmani, and Chen 2023; Ramezan and Leung 2020)

Other forms of fraud involve network attacks that target the blockchain’s core infrastructure. These include gaining majority control to manipulate transactions or creating alternate chains to disrupt consensus and rewrite transaction history (Ul Hassan, Rehmani, and Chen 2023; Shayegan, n.d.)

The different forms of fraudulent activities within blockchain ecosystems raise important questions about the defensive and preventive measures to safeguard the networks. Some systems have measures integrated into their architecture. On the XRPL, for instance,

accounts need to maintain a minimum balance of 20 XRP to remain active and create ledger entries, addressing the problem of abusive ledger spam (Mauri, Cimato, and Damiani 2018). However, unlike centralized systems, blockchains lack mediators to regulate transactions, making robust theft detection crucial to prevent major losses. Additionally, blockchain's inherent pseudo-anonymity poses difficulties for law enforcement agencies attempting to trace illicit transactions (Ul Hassan, Rehmani, and Chen 2023; Subashi 2024).

To address these challenges, various measures are employed, many of which rely on automated detection techniques. In cases of illegal activities, such as money laundering or illicit trade, detection methods combined with regulatory measures act as deterrents and help mitigate abuse. Automated fraud detection also plays a crucial role in preventing *token thefts*, which could otherwise enable fraudulent transactions to be added to the ledger. Additionally, advanced detection systems are employed to identify hacking attempts targeting blockchain code and infrastructure, preventing potential damage (Shayegan et al. 2022).

Our study aims to apply supervised machine learning algorithms to automate this very detection of fraudulent accounts, specifically on the XRPL. By doing so, we focus on the fraud types inherent in our labeled dataset presented in Section 4 of this work.

Related works on ETH and BTC Networks

From a broader perspective, related studies employ both supervised and unsupervised machine learning approaches to identify fraudulent accounts, primarily on ETH and BTC. A study by Lorenz et al. (2021) tested unsupervised against supervised models on BTC. The authors concluded that their unsupervised algorithms performed significantly below their supervised Random Forest baseline model. The illicit transactions of their flagged dataset did not represent outliers in their unsupervised approaches, making unsupervised anomaly detection methods rather ineffective for fraud detection at an account level.

Thereby, several other studies have proved that off-chain data enables effective supervised training of models to accurately detect fraudulent behavior on DLTs.

Ostapowicz and Żbikowski (2019), for instance, sourced a dataset from *Etherscan.io* containing 2,200 wallets labeled as fraudulent and 349,999 non-fraudulent wallets randomly selected from the ETH network. Random Forest, Support Vector Machine (SVM), and XGBoost classifiers were tested. The aggregated features included transaction metrics such as the amount and value of incoming and outgoing transactions, average transaction values, and timing metrics like the average time between transactions and the active duration of the account. Random Forest achieved the best recall rate of 84.92% when classifying fraudulent accounts, with a FPR of 9.69%. SVM showed high recall but had significantly high false positive rates, making it less effective for practical deployment.

Farrugia, Ellul, and Azzopardi (2020) employed an XGBoost classifier to detect illicit accounts in the ETH network. A total of 4,681 accounts (2,179 flagged by the ETH community and 2,502 normal accounts randomly selected) were used in the analysis. The best-performing model had an accuracy of 0.96, and an F1 score of 0.96. The most important features were the time difference between the first and last transaction, which showed a significant difference between normal accounts (average of 136.9 days) and illicit accounts (average of 38.4 days), as well as the total Ether balance of an account and the minimum value received in Ether. The authors state that the model demonstrated generalization capabilities and resilience towards overfitting, having high practical applicability in real-world scenarios.

Kılıç, Sen, and Özturan (2022) collected ETH transaction data between November 2019 and October 2020 as well as blacklisted addresses from *Etherscan Label Word Cloud, CryptoScamDB,* and *MyEtherWallet.* A total of 1,430 blacklisted addresses and 16,107,079 non-blacklisted addresses were collected. After resampling with the Synthetic Minority Over-sampling Technique (SMOTE) and undersampling, the training set comprised 25,500 fraud and 150,000 non-fraud addresses, while the test set included 1,430 fraud and 150,000 non-fraud

addresses. A total of 22 features were utilized, including transaction metrics (e.g., counts, total, average, minimum, and maximum values), temporal metrics (e.g., activity span and time differences), directional metrics (e.g., last transaction details), and graph-based metrics (e.g. PageRank). Eventually, Gradient Boosting achieved the best performance in terms of accuracy (0.985), precision (0.682), and F1 score (0.743). Random Forest had the best recall (0.876), indicating its effectiveness in identifying the blacklisted addresses.

Other studies, such as Nerurkar et al. (2021), applied non-binary classification tasks, to detect fraudulent accounts in the BTC ecosystem. They achieved the best accuracy scores with Random Forest, at 92 %, in classifying categories such as darknet markets, exchanges, gambling sites, Ponzi schemes, and unclassified wallets. However, the authors emphasized the need to incorporate explainability algorithms to improve the interpretability of their results, a crucial factor for practical deployment.

In conclusion, the existing studies (summarized in Table 1) show that supervised models, especially tree-based models, are highly effective in detecting fraudulent accounts. Challenges are balancing false positive and negative rates, feature selection, imbalanced datasets, and ensuring the explainability of models to better understand the patterns driving predictions. While considerable advancements have been made in fraud detection for BTC and ETH, there remains a noticeable gap in research focused on XRP. This study builds on these successful approaches, adapting them to the unique characteristics of the XRPL while exploring effective ways to maximize the use of available data. The primary goal is to assess the feasibility of automatically identifying fraudulent accounts on the XRPL and to develop a reliable model for practical application in real-world fraud detection scenarios.

4 Methodology

This section provides a concise overview of the proposed methods and model setups, as well as the data utilized in this work, including preprocessing and feature engineering steps. Subsection 4.1 presents an overview of the main datasets as well as the features engineered. Subsection 4.2 provides a detailed description of the data subsets utilized, feature selection processes, and the machine learning models as well as their evaluation methods.

4.1 Data and Feature Engineering

The transactional dataset for this study comprises over 2 billion XRPL transactions spanning 36 million ledgers (ledger 50,000,000 to 86,000,000), collected in a decentralized manner over 1,618 days (from 2019-09-13 to 2024-02-16). The main source of this data is a publicly available dataset with transactions from a full history ripple node provided by XRPL Labs developer *Wietse Wind* (Wind 2024).

During preprocessing, only successful transactions were retained to capture actual value transfers and avoid double-counting, as failed transactions are often retried. The XRPL supports over 40 transaction types, many of which relate to account settings or niche applications (Ripple, n.d.-a). To ensure a focus on economically significant activities, the analysis included transaction types such as *Payment, OfferCreate, OfferCancel, and TrustSet*. Additionally, escrow-related transactions *(EscrowCreate, EscrowCancel, etc.*) and specific NFT operations (*NFTokenCreateOffer, NFTokenAcceptOffer, NFTokenMint, etc.*) were included to account for value-transfer and contract-related activities. The selected transaction types account for over 99% of all transactions in the dataset. Other types, such as *AccountDelete, AccountSet, and TicketCreate*, were excluded as they do not represent asset movement or economic activity.

Feature Engineering

A total of 55 features were created on account level and are presented in Table 3. The foundation for engineering these features are related studies discussed in Section 3, along with features specifically adapted to the characteristics of the XRPL. The features involve general transaction metrics that provide insights into overall account activity, including transaction count, active periods, and value movements, such as total amounts sent, received, and net balance changes. Payment ratios, divided into outgoing and ingoing, reflect how frequently accounts send or receive payments. Offer-related ratios and trust set & escrow ratios capture account involvement in creating offers or managing trust lines and escrows. NFT ratios, based on NFT related transaction types, describe activities related to NFT creation and trading. Tag presence ratios show how often tags are used, indicating interactions with platform-based entities that use tags to link transactions to end users. Finally, interaction metrics, such as unique transaction partners and ratio of interaction with the 20 largest exchanges, assess the diversity and characteristics of an account's interactions.

Aggregating XRP transaction data at the address level posed several challenges. The high volume of transactions made computations demanding, while the diversity of transaction types added complexity in defining consistent features. This was addressed by focusing on the most relevant transaction categories, as outlined earlier. Also, significant variance in activity levels between highly active accounts and smaller accounts with minimal transactions posed challenges in drawing meaningful insights across all accounts equally. To address this, ratio-based features were introduced to normalize activity levels and ensure account comparability. Additionally, data incompleteness and missing values created complications, particularly due to missing information on transaction amounts for non-XRP token payments. To address this, token-specific features were developed to distinguish between XRP and non-XRP transactions, ensuring a clearer data representation.

Ground Truth Data

A separate dataset containing fraudulent addresses, provided by *XRP Forensics*, was used for our supervised fraud detection models and will be referred to as the 'fraud dataset' in this work. The dataset includes 12,351 fraudulent XRP accounts, consisting of a combination of flagged and auto-traced cases. The accounts were involved in a wide range of fraudulent activities, primarily consisting of addresses related to *PlusToken* fraud cases (32.27%), spam (27.19%), theft (19.31%), and giveaway scams (17.32%) as well as a minority of other types, such as token scams and different *Ponzi* schemes. Definitions of these fraud types were presented in Section 3. The dataset does not include other forms of illegal activity, such as involvement in the trade of illegal goods, money laundering, or terrorist financing.

4.2 Supervised Fraud Detection

In the third part of our study, we focus on predicting fraudulent accounts on the XRPL by formulating a binary classification problem. Supervised learning models allow us to leverage real-world fraud indicators, ensuring the validation of our models through our off-chain fraud dataset. This approach facilitates robust data triangulation, enabling us to verify and measure the reliability of our predictions (Section 3). Figure 1 shows the complete methodological framework of this part of the study.

As highlighted in Section 3, related works have demonstrated that tree-based models are particularly effective in detecting fraud within DLTs. Building on these findings, we set our focus on utilizing a Decision Tree as our baseline model. We then expanded our focus to two ensemble techniques: Random Forest and LightGBM, to enhance complexity and predictability.

To ensure a comprehensive evaluation, we also tested additional ensemble methods, including AdaBoost, XGBoost, and CatBoost, alongside linear and kernel-based approaches

such as Logistic Regression and Support Vector Machines (SVMs). However, these models presented several limitations early on; their performance is not detailed in this work, as it would exceed the scope of the study. The additional ensemble methods suffered from substantial overfitting, while linear and kernel-based models significantly underperformed on key metrics such as recall and F1 score for the fraud class. Furthermore, linear models required preprocessing steps, such as feature scaling, which increased methodological complexity and reduced interpretability. Based on these insights, we prioritized the three selected tree-based techniques: Decision Tree, Random Forest, and LightGBM.

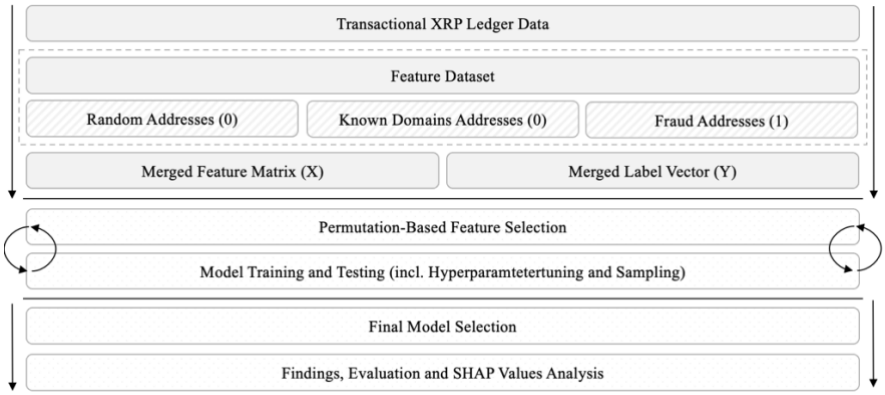

Figure 1: Methodological framework for supervised fraud detection

Data

The utilized dataset is a subset of the feature dataset, and includes 100,000 randomly selected addresses, 2310 verified known addresses and 12,351 addresses flagged as fraudulent. Similarly, as prior studies (Ostapowicz and Żbikowski 2019; Farrugia, Ellul, and Azzopardi 2020; Kılıç, Sen, and Özturan 2022), we assume that the vast majority of randomly selected addresses from the ledger represent legitimate users. The 2310 known addresses were verified, labeled, and checked manually on potential involvement in fraudulent activities, enhancing the reliability of the negative class. Eventually, addresses with fewer than five transactions were removed, resulting in a final set of 54,220 accounts in a 4:1 ratio (43,915 non-fraud accounts and 10,305 fraud accounts). This threshold excluded accounts with insufficient data, ensuring

the models could effectively learn the underlying patterns. The ratio of fraud and non-fraud addresses aligns with prior studies on ETH (Farrugia, Ellul, and Azzopardi 2020); however, this ratio was set arbitrarily and may not perfectly reflect real-world distributions of fraudulent activity on the XRPL. Nevertheless, the absence of prior research quantifying the amount of XRP fraud, this approach presented the most practical option.

Handling Class Imbalance and Probability Thresholds

To address class imbalance, we employed SMOTE. SMOTE generates synthetic examples for the training data between each minority class instance and its k-nearest neighbors (Chawla et al. 2002). We applied a sampling strategy of 0.5 to increase the number of fraudulent samples to a ratio of 2:1 (non-fraud:fraud), ensuring a more balanced distribution of both classes. Additionally, model weights were adjusted to place more emphasis on the underrepresented fraud addresses, further helping the models to learn from the minority class effectively. Probability thresholds, typically defaulting to 0.5, were fine-tuned to optimize the trade-off between precision and recall. Thereby, lowering the threshold below 0.5 increases recall, allowing more fraud cases to be detected, but potentially raising the number of false positives.

Feature Selection and Proposed Models

Features for each model type were selected using permutation-based feature selection, which measures the impact of each feature by evaluating the increase in model prediction error when its values are randomly shuffled (Molnar 2024). Features with zero or negative importance scores were excluded. The final feature set is detailed in Table 4. The hyperparameters of all approaches were fine-tuned using GridSearch. A concise overview of the models utilized is provided below.

1. Decision Tree Classifiers

Decision Tree served as our baseline due to its simplicity and interpretability. Decision Trees are non-parametric supervised models capable of handling both numerical and categorical data. In the learning process, Decision Trees recursively partition the feature space, grouping similar target values. At each node, candidate splits are evaluated, consisting of a feature and threshold, to partition the data aiming to minimize an impurity function, typically Gini Impurity. Impurity measures the degree of misclassification within a node. This process continues until the maximum allowable depth, which limits the number of splits, is reached, or all nodes become pure (containing only one class) (Scikit-Learn 2024).

2. Random Forest Classifiers

Random Forest is an ensemble learning technique that constructs multiple decision trees and aggregates their predictions to enhance overall accuracy. Each tree is built on a bootstrap sample from the training set and features are randomly selected at each node for splitting. This helps to reduce overfitting and de-correlate the trees. The effectiveness of Random Forests lies in balancing the strength of the individual trees while minimizing the correlations between them, resulting in high accuracy even with noisy datasets. (Breiman 2001)

3. LightGBM Classifiers

LightGBM is a Gradient Boosting Decision Tree (GBDT) method developed by Microsoft (Ke et al. 2017). GBDT models are built in a way that each new tree is sequentially added to enhance the accuracy of the previous ensemble. Each tree specifically addresses the residual errors left by earlier iterations (Scikit-Learn, n.d.; Chen and Guestrin 2016). LightGBM introduces two key optimizations: Gradient-based One-Side Sampling (GOSS), which focuses on high-gradient instances to retain informative samples while reducing training size, and Exclusive Feature Bundling (EFB), which combines mutually exclusive features to improve training speed in sparse feature spaces, making it both fast and effective for large-scale data analysis. (Ke et al. 2017)

Model Evaluation Approach and SHAP Value Analysis

To evaluate our models, the dataset was initially split into an 80/20 ratio, with 80% used for training and 20% reserved as a holdout set for the final evaluation of the best-performing model. During the training phase, all models underwent 10-fold cross-validation, where the training data was iteratively split into an 80/20 ratio for each fold. This comprehensive approach ensured robust evaluation across all models, minimized variance in performance estimates, and supported effective model refinement.

We prioritized recall and F1 score for the fraud class in all evaluations. Recall measures the proportion of actual fraud cases that were correctly identified by the model and is defined as $\frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ \ Negatives\ (FN)}$. Recall is particularly important in fraud detection, as we assume the highest costs in missing actual fraud cases (FNs). We aim to minimize the chances of fraudulent activities going undetected.

However, it is crucial to avoid unnecessary investigations and potential negative impacts on legitimate users by minimizing FPs. Precision is defined as $\frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)}$ and indicates the proportion of predicted fraud cases that were fraudulent.

The F1 score is the harmonic mean of precision and recall, balancing both metrics. A high F1 score indicates that the model effectively detects true fraud cases (high recall) while minimizing false alarms (high precision). This ensures a balanced approach to fraud detection, prioritizing both detection accuracy and reducing FPs.

Additionally, the receiver operating characteristic (ROC) curve and the area under the curve (AUC) were calculated to evaluate the ability of the models to balance TP and FPRs. An AUC of 0.5 indicates random guessing, while values closer to 1 signify stronger discriminatory power.

To assess whether our models are overfitting, we compared the learning curves by analyzing training and validation scores across different training sizes using 10-fold cross-validation. We focused on the F1 score for the fraud class, as well as the Macro F1 score, to get a comprehensive understanding of model performance for both classes while still giving special attention to the fraud cases.

The best-performing model was validated on the holdout set to confirm robustness. Selection criteria focused on maximizing Class 1 recall while maintaining a reasonable balance with precision to minimize missed fraud cases while ensuring reliability and efficiency across different data conditions. The model's ability to handle varying data sizes and computational efficiency were also considered to ensure practical applicability. This balance aims to achieve effective fraud detection, stable performance, and efficiency in terms of runtime.

To interpret the predictions of the best-performing model, we utilized Shapley (SHAP) values. SHAP values estimate each feature's contribution by averaging its impact over all possible combinations of features. This involves computing the difference in model output when including versus excluding a feature and averaging this difference across all feature subsets. The SHAP values also provides local insights on a feature level, ensuring the reliability of the explanation so that drivers and key patterns of fraudulent behavior can be identified, validating the interpretability of the selected model (Lundberg and Lee 2017).

5. Results and Discussion

This section presents and discusses the results of our fraud detection models, provides an analysis of key features, offers proposals for practical implementation, and discusses the implications within the XRP ecosystem and XRPL research. Thereby, the study's primary objective is addressed: evaluating the predictability of fraudulent accounts within the XRP network.

Model Comparison

Table 2 shows the performance of all classifiers benchmarked against the Decision Tree.

1. Decision Tree

The baseline Decision Tree model demonstrated strong overall performance. The ROC curve (Figure 2) showed an AUC of 0.92, indicating that the model effectively discriminates between fraudulent and non-fraudulent accounts. For the negative Class 0, it achieved an F1 score of 0.97, reflecting strong precision and recall in predicting the majority class. However, for the positive minority Class 1 (fraudulent cases), the model showed a precision of 0.86, a recall of 0.88, and an F1 score of 0.87. While these values are relatively high, the recall value of 0.87 suggests that the model missed some of the actual fraud cases. Specifically, 825 out of 6595 were misclassified. Examining the learning curves, both the Validation F1 macro and F1 for Class 1 remained stable, with only minor fluctuations as the training set size increased. The difference between training and validation scores remained under 10% for F1 macro and less than 15% for Class 1 F1 score, indicating good generalization with minimal overfitting. To address the class imbalance, two techniques were compared: balancing class weights and over-sampling with SMOTE. Applying a balanced weights parameter resulted in nearly identical results on all metrics, with a small improvement of Class 1 precision to 0.87. This indicates that the class imbalance did not significantly affect the baseline model. Applying SMOTE improved Class 1 recall to 0.89, demonstrating its ability to identify more fraud cases. However, precision dropped to 0.84, with AUC unchanged at 0.92. Overall, the runtime for the SMOTE model was significantly higher.

2. Random Forest

Our Random Forest classifiers demonstrated performance improvements over the Decision Tree baseline (Table 2). At the default threshold (0.5), Random Forest increased Class 1 precision from 0.86 to 0.97, Class 1 F1 score from 0.87 to 0.92, and the AUC from 0.92 to 0.99,

highlighting the model's ability to generalize and better distinguish between classes. However, Class 1 recall slightly decreased from 0.87 to 0.86, indicating a persistent challenge in improving the detection of actual fraud cases. Lowering the probability threshold ultimately improved Class 1 recall. As SMOTE did not outperform the balanced weights approach and required longer runtimes, it was not selected. Our best-performing Random Forest classifier achieved a Class 1 recall of 0.92 at a threshold of 0.23, with a precision and F1 score at 0.91. This configuration allowed the model to capture more fraudulent cases with minimal trade-offs in precision. The confusion matrix in Figure 3 shows that only 552 out of the 6,595 actual fraud cases were misclassified, compared to 825 misclassifications with the baseline model.

Eventually, the learning curves (Figure 3) demonstrate that Random Forest exhibited less overfitting compared to the baseline model, with stable performance over different training set sizes. The difference between training and Validation F1 macro remained approximately within 5%, and the Class 1 F1 score difference stayed under 10%.

3. LightGBM

Our LightGBM classifier improved the predictability compared to Random Forest, with superior computational efficiency. As shown in Table 2, the performance improved by 0.01 for Class 1 recall, precision and F1 score. A balanced weights approach was chosen over SMOTE, as Class 1 precision was higher by 0.1 with identical recall when tested at the default threshold. Ultimately, LightGBM achieved the best results with a probability threshold of 0.35, balancing precision and recall more effectively, with less significant threshold adjustments compared to Random Forest. Examining the learning curves (Figure 4), the validation scores for LightGBM were essentially similar to those of Random Forest, with a slight improvement as validation scores started marginally higher and improved slightly faster with additional data.

Table 5 shows a comparison in computational efficiency. The LightGBM model outperformed Random Forest in both training and inference times. It completed training in 1.03 seconds, compared to 49.88 seconds for Random Forest. This improvement is likely attributed

to LightGBM's optimized Gradient Boosting approach, which is faster and more efficient than Random Forest's ensemble method, where trees are trained independently. Similarly, LightGBM's inference time per instance was 0.002 ms, shorter than Random Forest's 0.051 ms. In terms of memory usage, Random Forest slightly outperformed LightGBM, consuming 458.30 MB compared to LightGBM's 491.61 MB. The minimal difference is likely due to LightGBM's optimization techniques, which require additional computational memory.

Final Model Selection and Holdout Test

The initial Decision Tree model served as a simple baseline, while overall model accuracy improved with the use of Random Forest and LightGBM algorithms. Notably, the baseline model's decent performance underscores the quality of the selected features, which provide strong discriminative power between fraudulent and non-fraudulent accounts. Eventually, the LightGBM model with balanced weights and a probability threshold of 0.35 was selected as the final model for its ability to balance performance, computational efficiency, and practical applicability. LightGBM effectively detected most fraud cases while maintaining a balanced F1 score, exhibited minimal overfitting, and demonstrated stable learning curves. This is critical, as consistent performance across different sample sizes ensures the model's reliability in detecting fraudulent behavior without excessive variance. Furthermore, the model achieved significantly faster training and inference times, making it the preferred choice for tasks requiring rapid processing and real-time predictions, which are likely relevant in a potential implementation scenario.

Ultimately, the LightGBM classifier was tested on a holdout set to evaluate its performance on unseen data. When comparing the validation results to its performance on the holdout (Table 2), the model maintains consistent effectiveness and generalization capabilities. Key metrics for the negative Class 0 remained unchanged, along with the AUC, confirming the model's overall robustness. While the metrics for the positive fraud Class 1 showed a slight

reduction on the holdout set, this decrease was minimal and did not compromise the model's reliability.

In absolute numbers (

Figure 5), the model detected 1,905 out of 2,061 fraudulent accounts, resulting in only 156 FNs. This ensures that most fraudulent accounts are identified. Additionally, the model flagged a total of 2,115 accounts as fraudulent (TPs + FPs), of which 1,905 were correctly identified. This balance between minimizing FNs and maintaining a low number of FPs is critical for ensuring the efficient use of investigative resources while maintaining high accuracy in fraud detection. For legitimate accounts, the model correctly classified 8,573 accounts as legitimate while misclassifying 210 accounts as fraudulent.

Eventually, these results highlight the model's ability to handle both fraudulent and legitimate accounts effectively, ensuring robust performance across all classes. The consistent metrics across validation and holdout sets underscore the model's strong generalization ability, reinforcing its suitability for practical implementation in fraud detection systems.

Comparing our results to related works presented in Section 3, our final models exhibit performance comparable to those reported by Ostapowicz and Żbikowski (2019) (Random Forest recall: 0.849), Kılıç, Sen, and Özturan (2022) (Random Forest recall: 0.876), and Farrugia, Ellul, and Azzopardi (2020) (XGBoost F1 score: 0.960). Thereby, our results reinforce the advantages of tree-based models and the computational efficiency of ensemble methods. Although these studies were conducted on different DLT ecosystems, our models demonstrate that leveraging supervised machine learning algorithms achieves at least comparable, if not superior, performance on the XRP Ledger, thereby supporting the broader applicability of these methods beyond BTC and ETH.

Model Interpretability and XRPL Fraud Patterns

Beyond predictability and generalization, interpretability is crucial to understanding what drives the model's predictions and what they reveal about fraudulent account behavior specific to the XRPL. Figure 6 ranks the top 15 features by absolute mean SHAP values, showing their global impact on the model's decisions. Figure 7 visualizes both the magnitude of each feature's impact and the relationship between feature values and different predicted outcomes.

The *payment_without_XRP_ratio* is the most influential variable in our data, with a mean absolute SHAP value of approximately 1.6. Our final model strongly relies on the proportion of non-XRP payments within data samples when making predictions, whereby accounts with a high ratio are likely indicative of fraudulent behavior, as suggested by the SHAP values in Figure 7. We hypothesize that such behavior may be linked to token manipulation scams or the use of non-XRP tokens to obscure transaction origins. This aligns with the labeled dataset, where PlusToken Ponzi-based scams represent the most prevalent type of fraud, resulting in the model strongly focusing on patterns associated with this fraud scheme.

Interestingly, Figure 7 shows that the *trust_set_ratio* for fraudulent accounts tends to be low, despite their heavy involvement in non-XRP token transactions. This suggests that fraudulent accounts might focus on a narrow set of non-XRP tokens and are unlikely to act as issuers themselves. Unfortunately, the data does not provide information about the token types involved and limits further analysis of this trend. Yet, we assume that fraudsters transact heavily within specific token ecosystems and primarily interact with other users rather than issuers, minimizing issuer involvement after the initial trust line setup.

active_days (absolute mean SHAP value ~0.65) and *absolute_active_period* (absolute mean SHAP value ~0.6) are the next most influential features, reflecting persistent activity and account longevity of accounts, respectively. Figure 7 shows that accounts with higher activity levels often correlate with fraud, presumably due to their involvement in recurring schemes requiring sustained operations. Fraudulent accounts also demonstrate a higher *transaction_frequency* compared to legitimate ones. This might indicate automated or bot-like

activity, where frequent transactions are used to manipulate patterns or evade detection. Interestingly, the relationship between *absolute_active_period* and fraud detection appears nonlinear: both short activity timeframes of an account (likely one-off fraudulent schemes) and extremely long activity periods (suggesting more sophisticated, long-term setups) contribute to the model's fraud predictions, while middle range activity durations (purple) show minimal influence on the model's classifications.

The *unique_transaction_partner_ratio* demonstrates a more linear relationship with the model's predictions, where higher values are associated with fraudulent accounts and lower values with normal addresses. Fraudsters on the XRPL appear to interact with a disproportionately high number of unique addresses. This behavior may reflect attempts to distribute funds widely or avoid repeated interactions with victims, thereby reducing traceability after fraudulent activities. This aligns with the high values observed for fraudulent accounts in the *unique_destination_partner* feature, which again reflects potential fund dispersion strategies or, alternatively, interactions with numerous victims to recruit new participants in Ponzi schemes. Furthermore, fraudulent accounts tend to exhibit higher values in the *payment_as_account_ratio*, indicating that they primarily initiate payments. This again may result from accounts being used to quickly disperse funds to other addresses, effectively obscuring their origin.

Eventually, as shown in Figure 7, fraudulent accounts tend to have lower values for the *ratio_known_counterparty_transactions* feature, which measures the proportion of transactions with the 20 largest known exchanges. This indicates that fraudulent actors may prefer less prominent counterparties, possibly to evade detection by systems monitoring interactions with regulated or well-known platforms.

Overall, the SHAP value analysis demonstrates that the model's predictions for fraudulent addresses are primarily driven by non-XRP token payments, high interaction ratios with unique counterparties, and consistent account activity over time, effectively distinguishing

fraudulent accounts from legitimate ones. Future analyses could explore nonlinear relationships and feature interactions, as well as broaden the understanding of fraud on the XRPL, ideally using larger, verified datasets with a wider selection of fraud types. Nevertheless, the insights provided can enable stakeholders to target specific fraud-related behaviors on the XRPL, allowing for more precise monitoring and intervention, and proactively identify potential fraudulent actors before significant damage occurs.

Practical Implications

Our final model has the potential to be implemented in the following scenarios:

1. Real-time fraud detection systems: The model can be integrated into automated monitoring systems to analyze XRPL transactions in real-time, identifying accounts that exhibit fraudulent behavior. This would enable proactive fraud prevention, enhancing the security of the ledger and reducing systemic risks.

2. Compliance and prosecution: Regulatory authorities could use the model to detect fraudulent accounts, ensure alignment with financial regulations, and support prosecution by identifying accounts involved in suspicious activities. Retrospective analysis of historical data could further uncover compliance gaps and refine regulatory frameworks focusing on XRP.

3. Project evaluation for Ripple: Ripple could use the model to assess whether funded projects were used as intended and delivered expected returns. By identifying signs of misuse, such as fraud or scams, the model supports accountability and ensures future funding decisions are better informed.

4. Increasing the security of the ledger: By identifying and flagging fraudulent accounts, the model could contribute to initiatives that improve the overall security and reliability of the XRPL. The findings from the model could lead to measures that reduce systemic risks, protect legitimate users, and promote trust in the ledger ecosystem.

Ultimately, our results demonstrate that detecting fraudulent XRPL accounts is highly effective using supervised machine learning. Our findings provide valuable insights and a practical framework for a wide range of XRP stakeholders involved in combating fraud on the network. Therefore, a final implementation and real-world testing of the model is necessary to assess its true generalization capabilities. Testing the model iteratively in diverse real-world scenarios will be crucial for validating its effectiveness and reliability in detecting fraudulent accounts. Furthermore, our approach not only enhances understanding within the XRP ecosystem but also offers a framework that can be adapted to other blockchain networks, thereby broadening its relevance and impact in DLT fraud detection.

6 Limitations

Our study is subject to several limitations that impact the scope and depth of the analysis.

The feature dataset was restricted to a specific time frame due to computational constraints, preventing the examination of transaction patterns beyond this period. The random sampling of data used for different approaches introduces variability that could impact the consistency of results. Crucial metadata, such as details of non-XRP token transactions, is absent due to the limitations of the public data source used. The study is also limited by its feature set. Time-based variables could not be fully computed due to computational resource constraints, and account-specific metrics, such as total address balances, were unavailable. Eventually, the inclusion of graph-based features could have enhanced the analysis by providing deeper insights into relationships and interactions between accounts.

Also, the dynamic nature of blockchain usage presents a hurdle. Methodologies effective today may become less applicable over time as user behaviors and network structures evolve (Victor 2020).

Specific to our methodology, the real-world applicability of our fraud detection models depends on how well the types of fraud represented in the labeled dataset align with actual fraud patterns. Thereby, the distribution of our ground truth data may not reflect real-world proportions. *Plus Token* frauds – for instance – dominate, potentially biasing the model towards detecting these cases while overlooking less common fraud types. Future work could explore fraud types beyond the scope of this study, such as phishing and wash trading, or focus on detecting accounts linked to activities like money laundering or terrorist financing.

Other limitations stem from the selected data subsets and adopted premises. With sufficient information, selecting a balance between fraud and non-fraud addresses that reflects the actual distribution on the XRPL would increase the generalizability of the final model. A limitation also stems from randomly sampling 100,000 addresses and assuming they are in the majority non-fraudulent. While this approach is common in related research (Section 3), it remains an approximation that may introduce inaccuracies.

Ultimately, a more in-depth analysis of incorrectly classified accounts and feature influence – ideally supported by more complete data on non-XRP token types – could enhance model interpretability and provide deeper insights into XRPL fraud patterns.

7 Conclusion

By analyzing XRPL transaction data on account-level, this study demonstrated the capability of supervised machine learning in effectively detecting fraudulent XRP accounts.

The need for greater transparency on the XRPL arises from its role in facilitating large-scale cross-border transactions, token issuance, decentralized exchange activities, NFT operations, all conducted in a pseudonymous manner. This lack of transparency presents challenges for regulators, compliance bodies, investors, financial institutions, and the broader XRP ecosystem in assessing the network's reliability.

Our fraud detection models demonstrated that identifying illicit accounts on the XRPL is feasible using supervised learning algorithms. The final model accurately detected nearly all fraud cases in the dataset and exhibited robust generalization capabilities, making it a strong candidate for real-time detection systems as well as for historical analyses of past XRP fraud cases. The insights gained into fraud patterns and the key features driving the classification showed that XRPL fraudsters tend to use non-XRP tokens and exhibit transactions with a large base of counterparties in high frequencies. Eventually, our insights offer valuable guidance for tailoring regulatory frameworks to the XRPL. To enhance our findings, future research could build on our approach by incorporating additional fraud types not represented in the current dataset.

Eventually, certain limitations of our study must be acknowledged. Constraints in data availability, including limited metadata and imbalances in the ground through datasets, affected the generalizability of results. Additionally, the assumptions made during data sampling processes may influence the reliability of the findings. Future work should address these limitations by leveraging more extensive datasets, incorporating advanced feature engineering, and adapting the methodologies to the ever-evolving blockchain dynamics.

Ultimately, the insights and methodology developed in this study extend beyond the XRPL, offering a framework that can be adapted to other blockchain ecosystems to address similar challenges in fraud detection. By bridging on-chain analysis with off-chain validation, reliable insights can be generated to enhance trust and accountability in decentralized systems. To build on our findings, further research should prioritize collaborations between blockchain developers, regulatory bodies, and academic institutions to refine data collection processes and standardize methodologies for addressing fraud. Strengthening transparency and mitigating risks in blockchain ecosystems will be key to enforce broader adoption and acceptance of decentralized technologies in financial markets, and beyond.

List of References

Ahmadova, Sevinj, and Mustafa Salim Erek. 2022. ‘A Review on Ripple, a Financial Intermediary Coin’. *Journal of Academic Projection* 7 (2): 117–30.

Akcora, Cuneyt Gurcan, Yulia R. Gel, and Murat Kantarcioglu. 2022. ‘Blockchain Networks: Data Structures of Bitcoin, Monero, Zcash, Ethereum, Ripple, and Iota’. *Wiley Interdisciplinary Reviews. Data Mining and Knowledge Discovery* 12 (1): e1436. https://doi.org/10.1002/widm.1436.

Alahmad, Mohammed, Adel Alfouderi, Ahmad Alonaizi, and Meshal Aldhamen. 2023. ‘Comparison Study of the Top 5 Leading Cryptocurrencies Based on General Consensus Protocol: Bitcoin, Ethereum, Tether, XRP and Bitcoin Cash’. *WSEAS TRANSACTIONS ON COMPUTER RESEARCH* 11 (April):23–32. https://doi.org/10.37394/232018.2023.11.3.

Breiman, Leo. 2001. ‘Random Forests’. *Machine Learning* 45 (1): 5–32. https://doi.org/10.1023/A:1010933404324.

Chainalysis. 2019. ‘PlusToken Scammers Didn’t Just Steal $2+ Billion Worth of Cryptocurrency. They May Also Be Driving Down the Price of Bitcoin.’ *Chainalysis* (blog). 16 December 2019. https://www.chainalysis.com/blog/plustoken-scam-bitcoin-price/.

———. 2024. ‘The 2024 Crypto Crime Report’. https://www.chainalysis.com/wp-content/uploads/2024/06/the-2024-crypto-crime-report-release.pdf.

———. n.d. ‘Key Players In Crypto Report’. Chainalysis.

Chase, Brad, and Ethan MacBrough. 2018. ‘Analysis of the XRP Ledger Consensus Protocol’. arXiv. https://doi.org/10.48550/arXiv.1802.07242.

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. 'SMOTE: Synthetic Minority Over-Sampling Technique'. *Journal of Artificial Intelligence Research* 16 (June):321–57. https://doi.org/10.1613/jair.953.

Chen, Tianqi, and Carlos Guestrin. 2016. 'XGBoost: A Scalable Tree Boosting System'. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–94. San Francisco California USA: ACM. https://doi.org/10.1145/2939672.2939785.

CoinMarketCap. 2024. 'Cryptocurrency Prices, Charts And Market Capitalizations'. CoinMarketCap. 2024. https://coinmarketcap.com/.

European Union Agency for Cybersecurity (ENISA). n.d. 'Blockchain'. Page. ENISA. Accessed 28 November 2024. https://www.enisa.europa.eu/topics/incident-response/glossary/blockchain.

Farrugia, Steven, Joshua Ellul, and George Azzopardi. 2020. 'Detection of Illicit Accounts over the Ethereum Blockchain'. *Expert Systems with Applications* 150 (July):113318. https://doi.org/10.1016/j.eswa.2020.113318.

Hellwig, Daniel, Goran Karlic, and Arnd Huchzermeier. 2020. *Build Your Own Blockchain: A Practical Guide to Distributed Ledger Technology*. Management for Professionals. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-40142-9.

Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. 'LightGBM: A Highly Efficient Gradient Boosting Decision Tree'. In *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

Kılıç, Baran, Alper Sen, and Can Özturan. 2022. 'Fraud Detection in Blockchains Using Machine Learning'. In *2022 Fourth International Conference on Blockchain Computing and Applications (BCCA)*, 214–18. https://doi.org/10.1109/BCCA55292.2022.9922045.

Kraken. n.d. ‘Beware of Crypto Giveaway Scams | Kraken’. Accessed 27 November 2024. https://support.kraken.com/hc/en-us/articles/360057159411-Beware-of-crypto-giveaway-scams.

Li, Xiaofan, and Andrew Whinston. 2020. ‘Analyzing Cryptocurrencies’. *Information Systems Frontiers* 22 (February). https://doi.org/10.1007/s10796-019-09966-2.

Lorenz, Joana, Maria Inês Silva, David Aparício, João Tiago Ascensão, and Pedro Bizarro. 2021. ‘Machine Learning Methods to Detect Money Laundering in the Bitcoin Blockchain in the Presence of Label Scarcity’. arXiv. http://arxiv.org/abs/2005.14635.

Lundberg, Scott, and Su-In Lee. 2017. ‘A Unified Approach to Interpreting Model Predictions’. arXiv. http://arxiv.org/abs/1705.07874.

Mauri, Lara, Stelvio Cimato, and Ernesto Damiani. 2018. ‘A Comparative Analysis of Current Cryptocurrencies’: In *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 127–38. Funchal, Madeira, Portugal: SCITEPRESS - Science and Technology Publications. https://doi.org/10.5220/0006648801270138.

Molnar, Christoph. 2024. *Interpretable Machine Learning A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book/feature-importance.html.

Morgia, Massimo La, Alessandro Mei, Francesco Sassi, and Julinda Stefa. 2024. ‘Pump and Dumps in the Bitcoin Era: Real Time Detection of Cryptocurrency Market Manipulations’. arXiv. http://arxiv.org/abs/2005.06610.

Nerurkar, Pranav, Sunil Bhirud, Dhiren Patel, Romaric Ludinard, Yann Busnel, and Saru Kumari. 2021. ‘Supervised Learning Model for Identifying Illegal Activities in Bitcoin’. *Applied Intelligence* 51 (June):1–20. https://doi.org/10.1007/s10489-020-02048-w.

Ostapowicz, Michał, and Kamil Żbikowski. 2019. ‘Detecting Fraudulent Accounts on Blockchain: A Supervised Approach’. In , edited by Reynold Cheng, Nikos Mamoulis,

Yizhou Sun, and Xin Huang, 11881:18–31. Lecture Notes in Computer Science. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-34223-4_2.

Peduzzi, Gaspard, Jason James, and Jiahua Xu. 2021. 'Jack the Rippler: Arbitrage on the Decentralized Exchange of the XRP Ledger'. In *2021 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, 1–2. https://doi.org/10.1109/BRAINS52497.2021.9569833.

Ramezan, Gholamreza, and Cyril Leung. 2020. 'Analysis of Proof-of-Work-Based Blockchains Under an Adaptive Double-Spend Attack'. *IEEE Transactions on Industrial Informatics* 16 (11): 7035–45. https://doi.org/10.1109/TII.2020.2977689.

Rella, Ludovico. 2020. 'Steps towards an Ecology of Money Infrastructures: Materiality and Cultures of Ripple'. *Journal of Cultural Economy* 13 (2): 236–49. https://doi.org/10.1080/17530350.2020.1711532.

Ripple. 2024a. 'Transaction Metadata'. 9 October 2024. https://xrpl.org/docs/references/protocol/transactions/common-fields.

———. 2024b. 'What Is the XRP Ledger?' https://xrpl.org/docs/introduction/what-is-the-xrp-ledger.

———. n.d.-a. 'Transaction Types'. XRPL. Accessed 6 November 2024. https://xrpl.org/docs/references/protocol/transactions/types.

———. n.d.-b. 'XRP Ledger - Use Cases & Featured Projects'. Accessed 28 November 2024. https://xrpl.org/about/uses.

———. n.d.-c. 'XRP Ledger History'. Accessed 28 November 2024. https://xrpl.org/about/history.

Scikit-Learn. 2024. 'Scikit Learn Documentation: Decision Trees'. 2024. https://scikit-learn/stable/modules/tree.html.

———. n.d. ‘Ensembles: Gradient Boosting, Random Forests, Bagging, Voting, Stacking’. Scikit-Learn. Accessed 7 November 2024. https://scikit-learn/stable/modules/ensemble.html.

SEC. 2017. ‘Investor Alert: Ponzi Schemes Using Virtual Currencies’. https://www.sec.gov/files/ia_virtualcurrencies.pdf.

Sergeenkov, Andrey. 2024. ‘What Is Ripple (XRP)?’ Forbes. 27 October 2024. https://www.forbes.com/sites/digital-assets/article/what-is-ripple-xrp/.

Shayegan, Mohammad Javad. n.d. ‘A Collective Anomaly Detection Method Over Bitcoin Network’. https://arxiv.org/pdf/2107.00925.

Shayegan, Mohammad Javad, Hamid Reza Sabor, Mueen Uddin, and Chin-Ling Chen. 2022. ‘A Collective Anomaly Detection Technique to Detect Crypto Wallet Frauds on Bitcoin Network’. *Symmetry* 14 (2): 328. https://doi.org/10.3390/sym14020328.

Statista. n.d. ‘Number of Cryptocurrencies 2013-2024’. Statista. Accessed 30 November 2024. https://www.statista.com/statistics/863917/number-crypto-coins-tokens/.

Subashi, Roland. 2024. ‘Cryptocurrencies and Money Laundering’. *Balkan Journal of Interdisciplinary Research* 10 (1): 55–62. https://doi.org/10.2478/bjir-2024-0005.

Sun Yin, Hao Hua, Klaus Langenheldt, Mikkel Harlev, Raghava Rao Mukkamala, and Ravi Vatrapu. 2019. ‘Regulating Cryptocurrencies: A Supervised Machine Learning Approach to De-Anonymizing the Bitcoin Blockchain’. *Journal of Management Information Systems* 36 (1): 37–73. https://doi.org/10.1080/07421222.2018.1550550.

Ul Hassan, Muneeb, Mubashir Husain Rehmani, and Jinjun Chen. 2023. ‘Anomaly Detection in Blockchain Networks: A Comprehensive Survey’. *IEEE Communications Surveys & Tutorials* 25 (1): 289–318. https://doi.org/10.1109/COMST.2022.3205643.

Victor, Friedhelm. 2020. ‘Address Clustering Heuristics for Ethereum’. In *Financial Cryptography and Data Security*, edited by Joseph Bonneau and Nadia Heninger, 12059:617–33. Lecture Notes in Computer Science. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-51280-4_33.

Victor, Friedhelm, and Andrea Marie Weintraud. 2021. ‘Detecting and Quantifying Wash Trading on Decentralized Cryptocurrency Exchanges’. arXiv. https://doi.org/10.48550/arXiv.2102.07001.

Wind, Wietse. 2024. ‘WietseWind/Fetch-Xrpl-Transactions’. https://github.com/WietseWind/fetch-xrpl-transactions.

APPENDIX

Study	Data	Methods	Key Results
Ostapowicz & Żbikowski (2019)	ETH (2,200 fraudulent wallets, 349,999 non-fraudulent wallets)	Random Forest , SVM, XGBoost	Random Forest: Best recall (84.92%), FPR (9.69%); XGBoost: similar to Random Forest, RF slightly better; SVM: High recall, but high FPR (less practical).
Farrugia et al. (2020)	ETH network (4,681 accounts, 2,179 flagged, 2,502 normal accounts)	XGBoost	Best accuracy: 0.96, F1 score: 0.96. Generalization and resilience towards overfitting.
Lorenz et al. (2021)	BTC transactional dataset (203,769 transactions), comparing supervised vs. unsupervised models	Supervised: Random Forest Unsupervised: KNN, PCA, Isolation Forest	Unsupervised models performed below supervised baseline. Fraudulent transactions not detected as outliers.
Nerurkar et al. (2021)	BTC (non-binary classification task: darknet markets, exchanges, gambling, Ponzi, unclassified)	Log Regression, SVM, Random Forest, XGBoost	Random Forest: best accuracy (0.92)

Table 1: Overview of key work of fraud detection

		Decision Tree baseline	**Decision Tree** SMOTE	**Decision Tree** balanced weights	**Random Forest**	**Random Forest** SMOTE	**Random Forest** balanced weights	**LightGBM** Balanced weight	**LightGBM** Holdout Test
	Prob. Threshold	0.50	0.50	0.50	0.50	0.35	0.23	0.35	0.35
Class 0	*Precision*	0.97	0.97	0.97	0.97	0.98	0.98	0.98	0.98
	Recall	0.97	0.96	0.97	0.99	0.98	0.98	0.98	0.98
	F1	0.97	0.97	0.97	0.98	0.98	0.98	0.98	0.98
Class 1	*Precision*	0.86	0.84	0.87	0.97	0.91	0.91	0.92	0.90
	Recall	0.87	0.89	0.87	0.86	0.91	0.92	0.93	0.92
	F1	0.87	0.86	0.87	0.92	0.91	0.91	0.92	0.91
Class 0, 1	*F1 macro*	0.92	0.92	0.92	0.95	0.94	0.95	0.95	0.95
	F1 weighted	0.95	0.95	0.95	0.97	0.97	0.97	0.97	0.97
	AUC	0.92	0.92	0.92	0.99	0.99	0.99	0.99	0.99

Table 2: Evaluation metrics of fraud detection classifiers

Feature	Description
payment_with_xrp_ratio	Ratio of payment transactions with a specified XRP amount to total transactions.
payment_without_xrp_ratio	Ratio of payment transactions without a specified XRP amount to total transactions.
offer_create_with_xrp_ratio	Ratio of offercreate transactions with a specified XRP amount (TakerGets/TakerPays) to total transactions.
offer_create_without_xrp_ratio	Ratio of offercreate transactions without a specified XRP amount to total transactions.
offer_cancel_ratio	Ratio of offercancel transactions to total transactions.
trust_set_ratio	Ratio of trustset transactions to total transactions.
escrow_ratio	Ratio of escrow transactions (all escrow-related types) to total transactions.
nftoken_create_offer_ratio	Ratio of nftokencreateoffer transactions to total transactions.
nftoken_cancel_offer_ratio	Ratio of nftokencanceloffer transactions to total transactions.
nftoken_accept_offer_ratio	Ratio of nftokenacceptoffer transactions to total transactions.
nftoken_mint_ratio	Ratio of nftokenmint transactions to total transactions.
total_transactions	Count of all transactions (incoming and outgoing) involving the account.
average_sent_amount_xrp	Average XRP amount sent in payment transactions where the address is the sender.
average_received_amount_xrp	Average XRP amount received in payment transactions where the address is the receiver.
median_sent_amount_xrp	Median XRP amount sent in payment transactions where the address is the sender.
median_received_amount_xrp	Median XRP amount received in payment transactions where the address is the receiver.
stddev_sent_amount_xrp	Standard deviation of XRP amounts sent in payment transactions where the address is the sender.
stddev_received_amount_xrp	Standard deviation of XRP amounts received in payment transactions where the address is the receiver.
payment_as_account_ratio	Ratio of payment transactions where the address is the sender (account) to total transactions.
payment_as_destination_ratio	Ratio of payment transactions where the address is the receiver (destination) to total transactions.
source_tag_present_ratio	Ratio of transactions with a non-null SourceTag when the address is the sender to total transactions.
destination_tag_present_ratio	Ratio of transactions with a non-null DestinationTag when the address is the receiver to total transactions.
unique_transaction_partner_ratio	Ratio of unique transaction partners in payment transactions to total transactions.
payment_small_amounts_sender_ratio	Ratio of payment transactions under specific amounts (10, 20, 50, 100, 200, 300, 500 XRP) sent to total transactions.
payment_small_amounts_receiver_ratio	Ratio of payment transactions under specific amounts (10, 20, 50, 100, 200, 300, 500 XRP) received to total transactions.
payment_large_amounts_sender_ratio	Ratio of payment transactions over specific amounts (500, 1,000, 10,000, 100,000, 1,000,000 XRP) sent to total transactions.
payment_large_amounts_receiver_ratio	Ratio of payment transactions over specific amounts (500, 1,000, 10,000, 100,000, 1,000,000 XRP) received to total transactions.
absolute_active_period	Difference in days between the first and last transaction timestamps for the account, indicating the span of activity.
active_days	Count of unique days during which the account conducted at least one transaction.
transaction_frequency	Number of transactions over the active period of the account.
unique_destination_partners	Count of unique transaction partners where the address is the sender.

unique_account_partners	Count of unique transaction partners where the address is the receiver.
transaction_ratio_20_largest_exchanges	Ratio of transactions associated with the 20 largest exchanges to total transactions.
total_sum_sent	Sum of all outgoing transaction values for the account over the observed period.
total_sum_received	Sum of all incoming transaction values for the account over the observed period.

Table 3: Account level feature set

Features for LightGBM
absolute_active_period
active_days
average_received_amount_xrp
average_sent_amount_xrp
destination_tag_present_ratio
median_received_amount_xrp
median_sent_amount_xrp
nitoken_create_offer_ratio
offer_cancel_ratio
offer_create_with_XRP_ratio
offer_create_without_XRP_ratio
payment_as_account_ratio
payment_over_100000_xrp_receiver_ratio
payment_over_100000_xrp_sender_ratio
payment_over_1000_xrp_receiver_ratio
payment_over_1000_xrp_sender_ratio
payment_over_10000_xrp_receiver_ratio
payment_over_10000_xrp_sender_ratio
payment_over_500_xrp_receiver_ratio
payment_over_500_xrp_sender_ratio
payment_under_10_xrp_sender_ratio
payment_under_100_xrp_receiver_ratio
payment_under_20_xrp_receiver_ratio
payment_under_200_xrp_receiver_ratio
payment_under_200_xrp_sender_ratio
payment_under_50_xrp_sender_ratio
payment_under_500_xrp_receiver_ratio
payment_without_XRP_ratio
ratio_known_counterparty_transactions
source_tag_present_ratio

stddev_received_amount_xrp
total_sum_received
total_sum_sent
transaction_frequency
trust_set_ratio
unique_account_partners
unique_destination_partners
unique_transaction_partner_ratio

Table 4: Final selection of features for LightGBM classifier

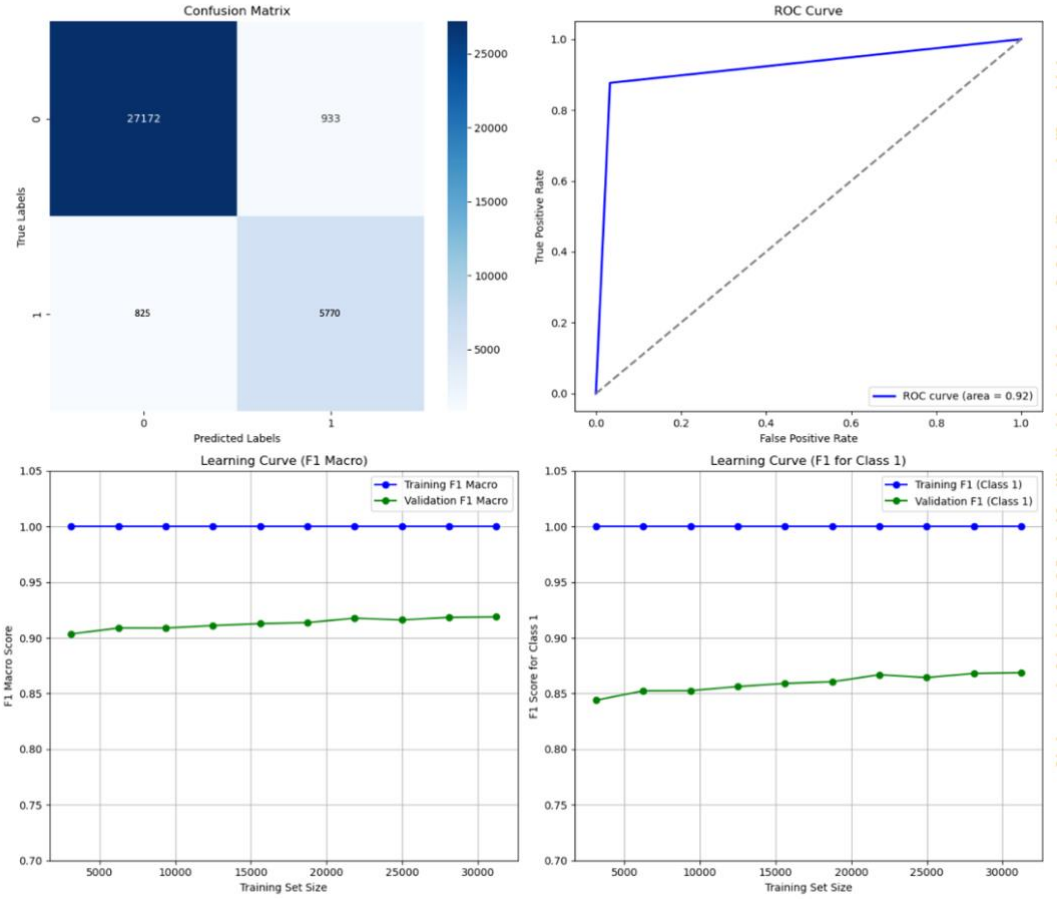

Figure 2: Evaluation metrics for decision tree baseline

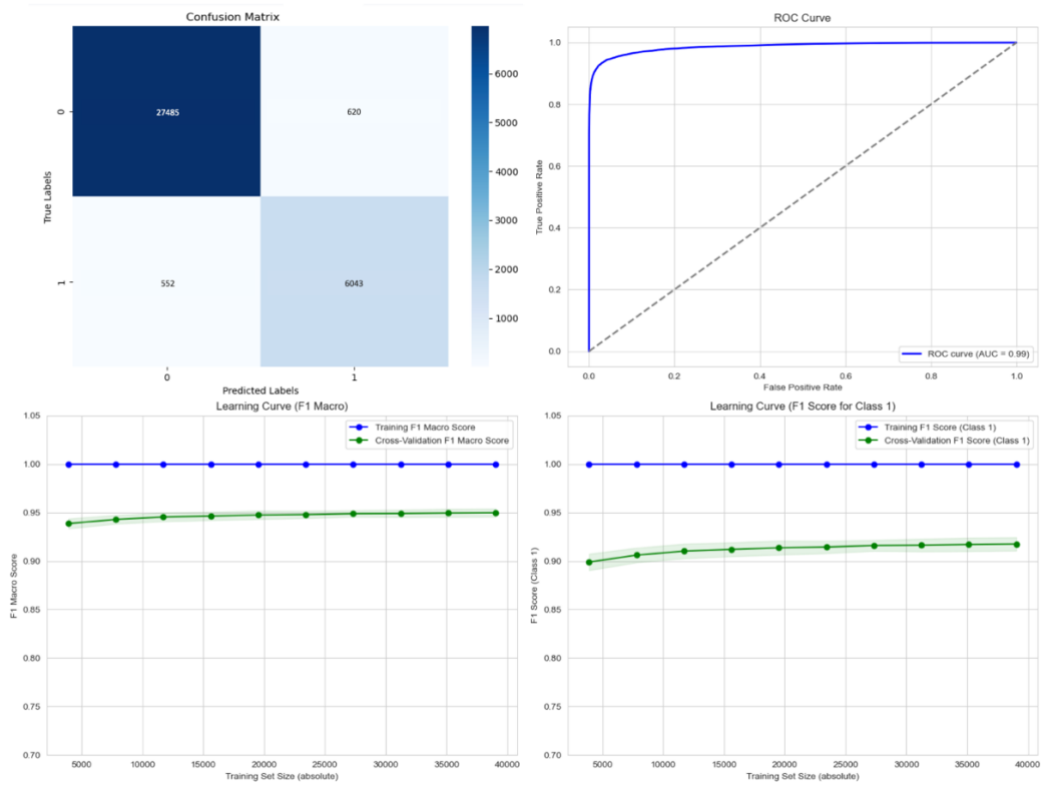

Figure 3: Evaluation metrics for best Random Forest Classifier - threshold 0.23 - balanced weights

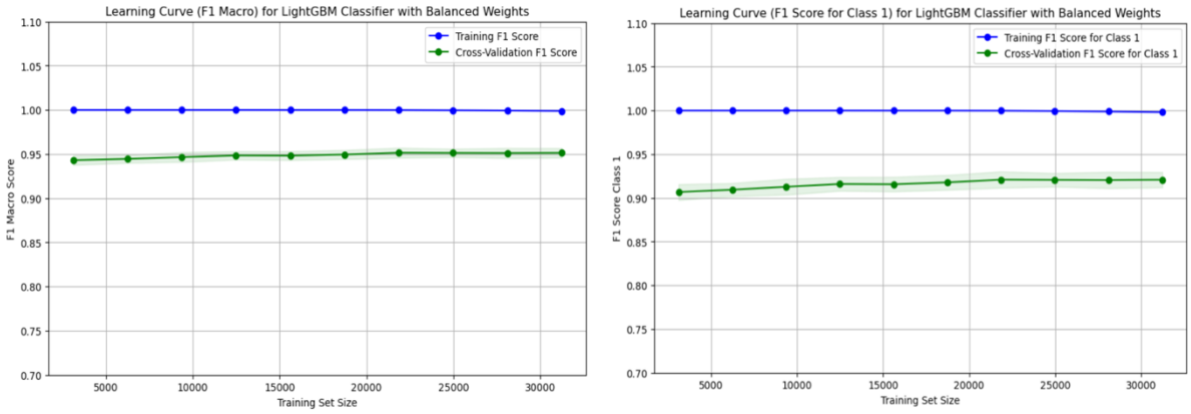

Figure 4: Learning curves LightGBM

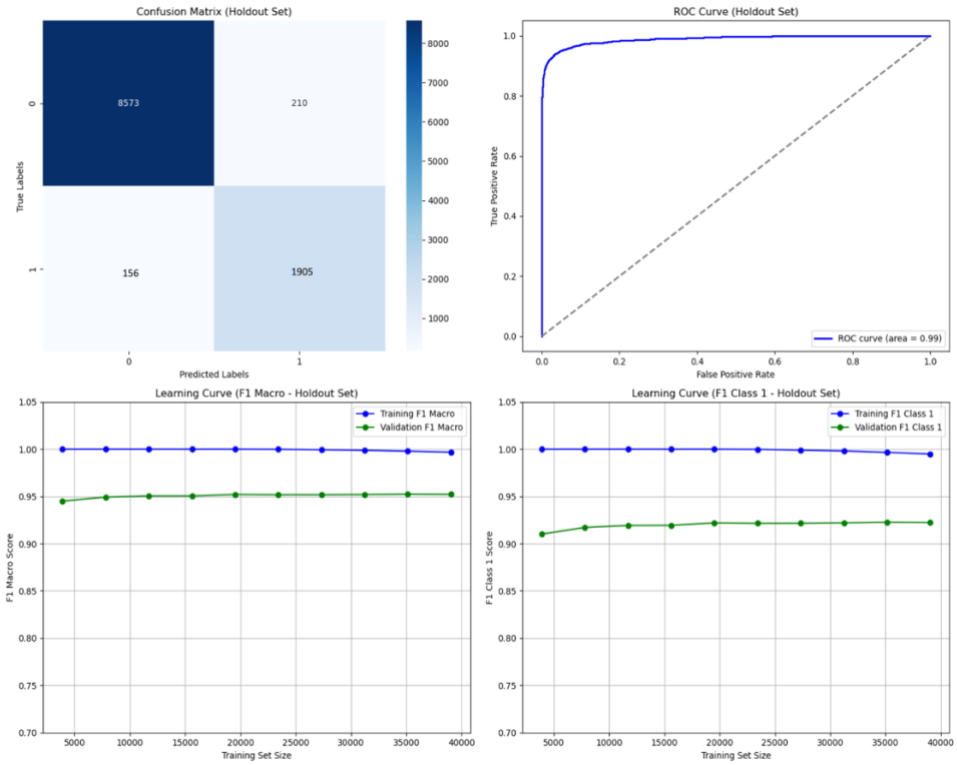

Figure 5: Evaluation metrics for LightGBM - holdout set - threshold 0.35 - balanced weights

Metric	Best Random Forest	Best LightGBM
Training Time (s)	49.88	1.03
Inference Time per Instance (ms)	0.050	0.002
Peak Memory Usage (MB)	458.30	491.61

Table 5: Comparison of computational performance

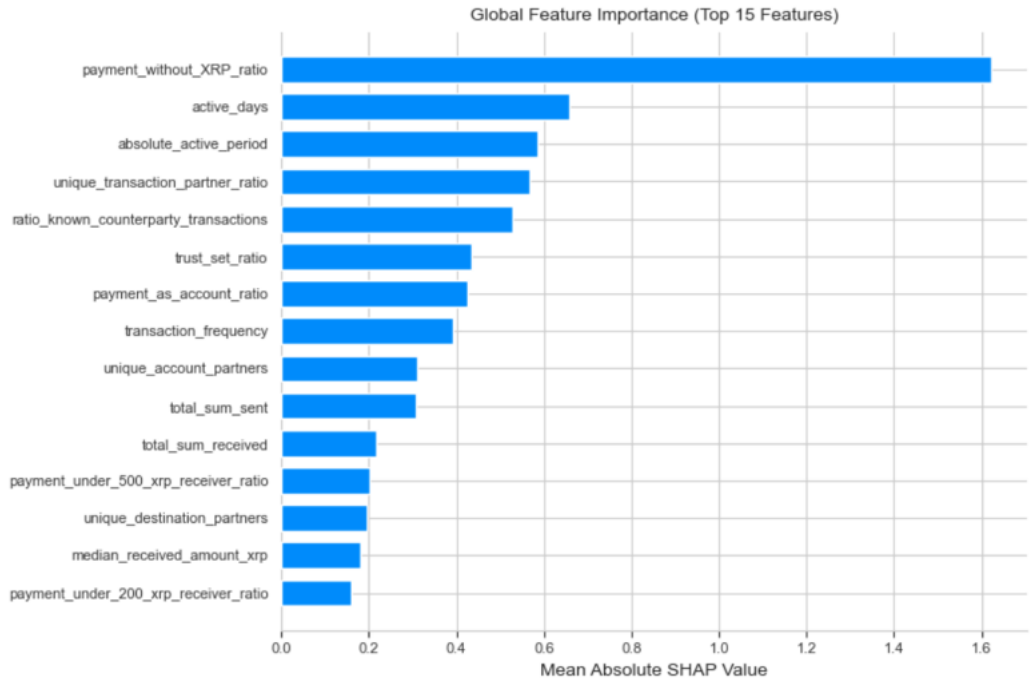

Figure 6: Global feature importance top 15 – LightGBM

Figure 7: SHAP values top 15 LightGBM