Machine learning methods to detect money laundering in the Bitcoin blockchain in the presence of label scarcity

Joana Lorenz
joana.lorenz@outlook.de
NOVA IMS & Feedzai

Maria Inês Silva
maria.silva@feedzai.com
Feedzai

David Aparício
david.aparicio@feedzai.com
Feedzai

João Tiago Ascensão
joao.ascensao@feedzai.com
Feedzai

Pedro Bizarro
pedro.bizarro@feedzai.com
Feedzai

ABSTRACT

Every year, criminals launder billions of dollars acquired from serious felonies (e.g., terrorism, drug smuggling, or human trafficking), harming countless people and economies. Cryptocurrencies, in particular, have developed as a haven for money laundering activity. Machine Learning can be used to detect these illicit patterns. However, labels are so scarce that traditional supervised algorithms are inapplicable. Here, we address money laundering detection assuming minimal access to labels. First, we show that existing state-of-the-art solutions using unsupervised anomaly detection methods are inadequate to detect the illicit patterns in a real Bitcoin transaction dataset. Then, we show that our proposed active learning solution is capable of matching the performance of a fully supervised baseline by using just 5% of the labels. This solution mimics a typical real-life situation in which a limited number of labels can be acquired through manual annotation by experts.

CCS CONCEPTS

• **Computing methodologies** → *Supervised learning by classification*; *Anomaly detection*; *Active learning settings*; • **Applied computing** → **Economics**.

KEYWORDS

anti money laundering, cryptocurrency, supervised classification, anomaly detection, active learning

ACM Reference Format:
Joana Lorenz, Maria Inês Silva, David Aparício, João Tiago Ascensão, and Pedro Bizarro. 2020. Machine learning methods to detect money laundering in the Bitcoin blockchain in the presence of label scarcity. In *ACM International Conference on AI in Finance (ICAIF '20),October 15–16, 2020, New York, NY, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3383455.3422549

1 INTRODUCTION

Money laundering is a high-impact problem on a global scale. Criminals obtain money illegally from serious crimes and inject it into the financial system as seemingly legitimate funds. Money laundering schemes usually involve large amounts of money and, when caught, typically result in large fines for financial institutions. Recent examples are the 1MDB [29] and the Danske Bank scandals [27].

Governments and international organizations are building tighter regulations around money laundering and are broadening them to include cryptocurrencies [28, 39], where criminals benefit from pseudonymity.

In the financial sector, Anti-Money Laundering (AML) efforts often rely on rule-based systems [21]. However, vulnerabilities derive from the relative simplicity of publicly available rule-sets, leading to high false-positive rates (FPR) and low detection rates [41]. Machine learning (ML) techniques overcome the rigidity of rule-based systems by inferring complex patterns from historical data, and can potentially increase detection rates and decrease FPRs.

Recently, Weber et al. [42] released a dataset, consisting of a sample of 200k labeled Bitcoin transactions, and evaluated various supervised models on it. Unfortunately, supervised methods are often unfeasible as institutions do not possess large-scale labeled datasets. This lack of labels is due to two main reasons. First, given the evolving complexity of money laundering schemes, it is unlikely to be possible to identify all (or even most) of the entities involved in money laundering. Second, labels resulting from law enforcement investigations are not immediate, and manual annotation is costly. Thus, in order to properly evaluate the practical feasibility of ML for AML, strategies that require no labels (unsupervised learning) or just a few labels (active learning) are paramount.

We address the real-world challenge of *how to detect money laundering in a dataset with few labels.* Particularly, we show that:

(1) Detecting money laundering cases in the Bitcoin network without any labels is impossible since illicit transactions hide within clusters of licit behaviour (Section 4.2).
(2) With just a few labels (approximately 5% of the total), one can match the results of a supervised baseline by using Active Learning (AL) (Section 4.3). This setting mimics a real-world scenario with limited availability of human analysts for manual labeling.

We extend the existing research on unsupervised illicit activity detection in cryptocurrency and financial transactions by benchmarking different methods on a real-world dataset with a relatively large number of positive cases. In this way, we overcome the typical limitation of evaluating on synthetic data or real data with few positive samples. Besides, to the best of our knowledge, we are the

ICAIF '20, October 15–16, 2020, New York, NY, USA
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-7584-9/20/10.
https://doi.org/10.1145/3383455.3422549

first work to apply AL to AML on a large transaction dataset and in the cryptocurrency setting, specifically.

We organize the remainder of the paper as follows. Section 2 presents the related work. Section 3 details the experimental setup and introduces the relevant anomaly detection methods as well as AL concepts. In Section 4 we present our results. Finally, the main conclusions are discussed in Section 5.

2 RELATED WORK

In this section, we present previous research on ML for AML in the context of both financial transactions and, more specifically, cryptocurrencies. For a thorough survey of ML approaches for AML, we refer the reader to Chen et al. [10].

Although approaches greatly vary, many methods assume money laundering cases to be outliers, i.e., illicit instances (a minority) should exhibit significantly different behaviours from legitimate ones (the majority). Typically, these approaches use unsupervised anomaly detection methods to model licit behaviour and find the instances that deviate from it [5, 9, 14, 18, 22, 23, 33, 38, 40].

Overall, the results of these studies are encouraging, reporting low FPRs [9, 22, 23] and good detection rates [9, 14, 23, 40]. Some studies even report that the ML approaches were able to detect money laundering patterns that were previously unknown [38] or not caught by rule-based systems [5]. However, a fair comparison between methods is impossible, given the heterogeneity of the evaluation setups. In these studies, researchers use real-world datasets labeled by analysts [5, 22, 23], with simulated illicit transactions [14, 38, 40], or no labels at all [18, 33]

Generally, authors are openly doubtful about real-world reproducibility of good results, in the face of intricate patterns and incomplete labels [9, 14, 40]. The question arises on whether reliable anomaly detection is possible in non-synthetic data, as criminals could intentionally mimic normal behaviour. In our research, we contribute to assess the reproducibility of such results by conducting the first in-depth benchmark of anomaly detection methods in a labeled real-world cryptocurrency dataset and comparing their performance against a supervised baseline.

Previous studies on money laundering in cryptocurrencies in particular are scarce and inconclusive due to a lack of labels for evaluation. Some conclude that supervised models perform well [3, 17, 26]. Others report low detection rates for unsupervised methods in extremely imbalanced data [25, 26, 31, 32, 43]. Often, the evaluation of anomaly detection methods consists of checking whether the anomalies represent extreme cases [31, 32] or behaviour deemed suspicious by human analysis [16].

Active Learning has been proposed as a method to reduce the number of labels needed for the training of an effective classifier by iteratively sampling the most informative samples for labeling from an initially unlabeled pool [34]. Given the apparent label scarcity in money laundering data, it is a highly relevant setting for the practical implementation of ML-based AML systems. Previously, Deng et al. [12] applied AL to detect money laundering in financial transactions. In an account-level classification of 92 real-life accounts, they report that their method can accurately estimate the threshold hyperplane with only 22% of the labels. AL has also successfully been applied in other fraud-related use-cases such as credit card fraud [6] and network intrusion detection [1, 15, 36], reporting the sufficiency of as few as 1.5% of the original labels to achieve near-optimal performance [15].

We conduct experiments with AL, assuming an unlabeled dataset and the capacity to acquire labels progressively to train a supervised classifier. We hereby extend the study by Deng et al. [12] to a transaction-level analysis in a much larger cryptocurrency dataset.

3 EXPERIMENTAL SETUP

3.1 Data

We use the Bitcoin dataset[1] released by Elliptic, a company dedicated to detecting financial crime in cryptocurrencies [42]. It includes 49 graphs sampled from the Bitcoin blockchain at different sequential moments in time (time-steps), as presented in Figure 1. Each graph is a directed acyclic graph, starting from one transaction, and including subsequent related transactions on the blockchain, containing approximately two weeks of data.

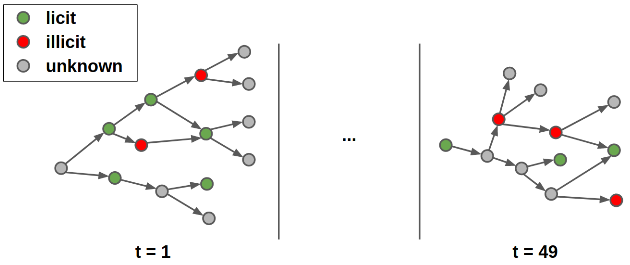

Figure 1: Structure of the dataset (taken from Bellei [4]).

Bitcoins transactions are transfers from one Bitcoin address (e.g., a person or company) to another, represented as nodes in the graph. Each transaction consumes the output of past transactions and generates outputs that can be spent by future transactions. The edges in the graph represent the flow of Bitcoins between transactions.

The dataset consists of 203,769 transactions, of which 21% are labeled as licit, and 2% as illicit, based on the *category* of the bitcoin address that created the transaction. The remaining transactions are unlabeled. Illicit categories include scams, malware, terrorist organizations, and Ponzi schemes. Licit categories include exchanges, wallet providers, miners, and licit services. Each transaction has 166 features, 94 of which represent information about the transaction itself. The remaining features were constructed by Weber et al. [42] using information one-hop backward/forward from the transaction, such as the minimum, maximum, and standard deviation of each transaction feature. All features, except for the time-step, are fully anonymized and standardized with zero mean and unit variance.

3.2 Methods

In this section, we give an overview of the methods used in our experiments and discuss our experimental setup. Following Weber et al. [42], we split the data into sequential train and test datasets for all experiments. The train set includes all labeled samples up to the 34th time-step (29894 transactions), and the test set includes all

[1] Available at https://www.kaggle.com/ellipticco/elliptic-data-set

labeled samples from the 35th time-step, inclusive, onward (16670 transactions). Like Weber et al. [42] we evaluate all methods using the F1-score for the illicit class, hereafter referred to as illicit F1-score.

3.2.1 *Supervised Learning.*

In order to benchmark unsupervised methods and AL, we first reproduce the results of Weber et al. [42] as our baseline.

We train each supervised model on the train set using all 166 features and then evaluate them on the entire test set. To measure performance over time, and following Weber et al. [42], we also report the illicit F1-score per time-step in the test set. We use the scikit-learn [30] implementation of logistic regression (LR) and random forest (RF) as well as the Python implementation of XGBoost [8]. We present the results achieved using default parameters, as in Weber et al. [42].

3.2.2 *Unsupervised Learning.*

Anomaly detection methods are unsupervised learning techniques to detect outliers in a dataset. Literature suggests their effectiveness in the AML context (Section 2). For a thorough review of anomaly detection, we refer the reader to the surveys by Chandola et al. [7] and Domingues et al. [13].

The standard definition of outliers refers to instances that are unlikely to be drawn from the same distribution as the train data or instances that are far from other data points in the feature space. Although we focus mainly on unsupervised anomaly detection, some methods are semi-supervised discriminators trained to learn a boundary around normal instances. In that context, outliers are instances that fall outside of the boundary [14].

We test seven common anomaly detection algorithms with readily available Python implementations: Local Outlier Factor (LOF), K-Nearest Neighbours (KNN), Principal Component Analysis (PCA), One-Class Support Vector Machine (OCSVM), Cluster-based Outlier Factor (CBLOF), Angle-based Outlier Detection (ABOD), and Isolation Forest (IF). We aim at a diversity of strategies. We use the PyOD package implementations [44] with default parameters.

LOF and KNN start by computing the distance of each instance to its k nearest-neighbour. Then, KNN defines that distance as the outlier score. LOF uses the distance to compute the instance's density, and if the density is substantially lower than the average density of its k nearest-neighbours, the instance is declared anomalous.

PCA and OCSVM define anomalies as observations that deviate from normal behaviour. They detect anomalous instances as observations with a large distance to the principal components (PCA) of non-anomalous observations or instances that lay outside of the decision boundary (OCSVM) learned around them.

CBLOF uses the outcome of a clustering algorithm on the instances (in our case k-means) and classifies each cluster as either *small* or *large.* It calculates an anomaly score for each instance, marking instances that belong to small clusters or that are far from big clusters as anomalous. ABOD computes the pairwise cosine similarities between all points and classifies those with a low average radius and variance as anomalies. Lastly, IF isolates anomalies by performing recursive random splits on attribute values. Based on the resulting tree structure, anomalies are instances that are easy to isolate, i.e., have shorter paths.

The introduced methods use different anomaly scores and scales. Thus, a fair comparison requires evaluation at different *contamination levels*, defined as the expected proportion of outliers in the dataset, and used to set a threshold for the decision function. Whereas the original PyOD implementation applies the contamination level on the scores of the train set, we apply it on the test set scores to guarantee that the desired percentage of positive cases (anomalies) in the test set is the same across methods. The contamination level here is analogous to a fixed alert rate in real AML systems, i.e., the percentage of cases flagged for further investigation by an analyst. We evaluate the illicit F1-score for each model at contamination levels between 0 and 1, with increments of 0.05. We also present the illicit F1-score of the RF supervised baseline, where we define the model threshold by setting the contamination level as the predicted positive rate (or alert rate), for comparison.

3.2.3 *Active Learning.*

AL is an incremental learning approach that interactively queries instances for labeling (e.g., by human analysts) and uses the increasing number of labeled instances to (re-)train a supervised model. It fits the AML context by addressing label scarcity and has previously been successfully applied to detect money laundering accounts based on financial transaction history. For an extensive survey on AL, we refer the reader to Settles [34].

The goal of AL is to minimize the number of labels necessary to achieve adequate classifier performance. The process starts with a pool of unlabeled instances (the *unlabeled pool*), although sometimes there is a residual number of labels. At each iteration, a *query strategy* queries a batch of instances for manual labeling. After labeling, the instances go into the *labeled pool.* Finally, a supervised algorithm (the *classifier*) is trained on the labeled pool and evaluated on a test set. If the performance is not satisfactory, the querying process continues to enrich the labeled pool incrementally. To mimic the manual labeling process in our experiments, we append the labels to the queried instances.

In the literature, query strategies build on various models and uncertainty criteria. In this study, we focus on four query strategies trained on the labeled pool to find informative instances in the unlabeled pool. Two of them, uncertainty sampling and expected model change, are supervised, requiring an underlying supervised model to define queries. The other two, elliptic envelope and Isolation Forest (IF), are unsupervised and find outlying instances with regards to the labeled pool. We use random sampling as a baseline. This setup was based on previous work done on Feedzai's active learning annotation tool, which was used to run the experiments.

Expected model change [34, 35] assumes that instances are more informative if they influence the model more strongly. It queries the unlabeled instances that lead to the most significant change in the model parameters by measuring the impact of labeling one unlabeled instance on the gradient of the model's loss function. Thus, this strategy applies only to gradient-based classifiers. In our experiment, we use LR. The expected model change is a weighted sum over all possible labels since the labels of the instances are unknown before querying. Then, at each iteration, we query the labels of the instances with the largest expected gradients.

Uncertainty sampling is one of the most commonly used query strategies [20, 34]. It queries the instances about which a model is most uncertain. Assuming a probabilistic learning model and a

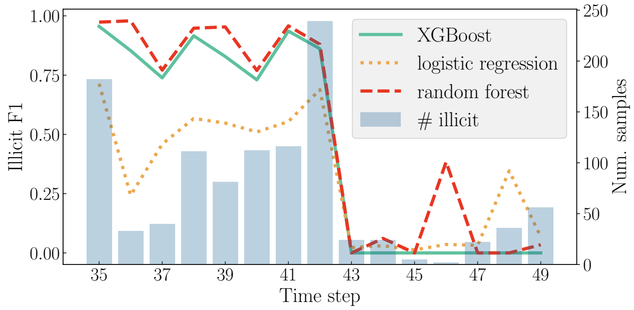

Figure 2: Illicit F1-score for each supervised baseline, across time-steps.

binary classification problem, this translates to querying the instances with the predicted score closest to 0.5. In our study, we use the same type of classifier for uncertainty sampling and evaluating on the test set; for instance, if the classifier is RF, we also conduct uncertainty sampling using RF.

The two unsupervised query strategies used are IF and elliptic envelope. Outliers are transactions with high anomaly scores (IF) or a large Mahalanobis distance to a multivariate Gaussian distribution fit on the labeled pool (elliptic envelope).

We combine unsupervised and supervised query strategies in our experiments, depending on the number of illicit instances in the labeled pool. After an initial random sample of one batch of instances, we use an unsupervised *warm-up learner* that samples instances until the labeled pool includes at least one illicit instance. When we reach this threshold, we either switch to a supervised *hot learner* or continue to use the warm-up learner. As the classifier, we use the three supervised models evaluated in the supervised baselines: RF, XGBoost, and LR. We compare all AL setups against a baseline that queries random instances at each iteration (also used as a warm-up learner). We use a batch size of 50 instances sampled at each iteration for all experiments. Each AL setup is run five times with different random seeds to ensure the robustness of the results. We assess the performance of each AL setup through the median illicit F1-score and the confidence intervals at each labeled pool size. This setup and parameter choices follow the work by Barata et al. [2] developed at Feedzai concurrently.

4 RESULTS

In this section, we present the experimental results for the supervised baseline, followed by the anomaly detection and the AL benchmarks.

4.1 Supervised baselines

We are able to reproduce the results presented by Weber et al. [42] closely. Over five runs (with different seeds), we achieve an illicit F1-score on the test set of 0.76 for XGBoost, 0.45 for LR, and 0.83 for RF. Thus, the best supervised baseline is achieved with the RF model. As Weber et al. [42], we observe that model performance is profoundly affected by a sudden dark market shutdown at time-step 43.

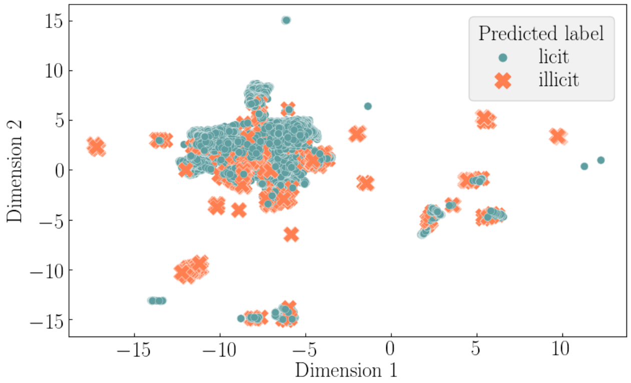

Figure 3: UMAP projection of the test set, colored by the labels predicted by IF.

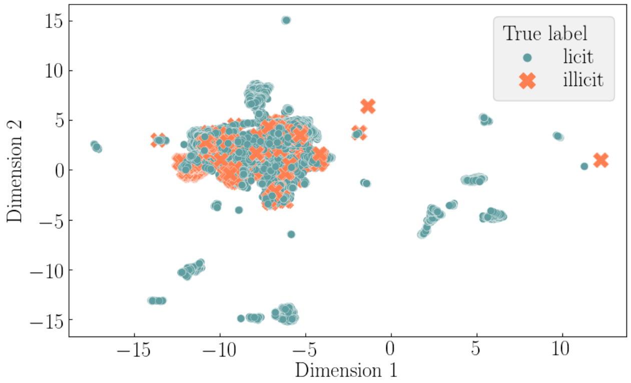

Figure 4: UMAP projection of the test set, colored by the true labels.

4.2 Anomaly detection

In Table 1, we present the illicit F1-score of the explored anomaly detection methods as well as the RF supervised baseline at different contamination levels. Recall that, at each contamination level, we define the threshold of the RF model so that it leads to an alert rate equal to that contamination level.

Table 1: Anomaly detection methods illicit F1-score by contamination level (RF supervised baseline for reference).

Model	**Contamination level**			
	0.05	**0.1**	**0.15**	**0.2**
RF supervised baseline	0.82	0.58	0.46	0.39
LOF	0.11	0.15	0.19	0.18
ABOD	0.07	0.07	0.07	0.07
KNN	0.03	0.04	0.05	0.06
OCSVM	0.01	0.03	0.03	0.04
CBLOF	0.01	0.02	0.03	0.04
PCA	0.01	0.01	0.02	0.02
IF	0.00	0.00	0.00	0.01

Anomaly detection methods perform significantly below the RF supervised baseline across all contamination levels. These results are not consistent with past studies, where anomaly detection methods perform adequately for AML (Section 2). However, we note that these studies often use synthetically generated anomalous data points that are outliers by design. Furthermore, there could be differences between money laundering patterns in financial transactions and Bitcoin transfers. In the real-life Bitcoin transaction dataset, we see that illicit cases are indeed not outlying.

To illustrate this, we apply the Uniform Manifold Approximation and Projection (UMAP) [24] to the test set and build two plots with the resulting projection. In the first (Figure 3), we color each observation based on the predicted label of the worst-performing method (IF), while in the second (Figure 4), we color each observation based on the true label. We can then see that the IF classifies most outlying instances as illicit, as intended. Still, the true labels presented in Figure 4 reveal that the illicit instances in the dataset are not actually outlying, but instead hiding among licit transactions.

The observation that not all outliers are illicit and that not all illicit transactions are outliers is reasonable in AML as sophisticated criminals obfuscate their activity by mimicking normal behaviour, hiding in regions of high nominal density. This problem was previously acknowledged by Das et al. [11]. Thus, we conclude that anomaly detection methods are ineffective for the unsupervised classification task in this real-life Bitcoin dataset.

4.3 Active learning

Table 2 summarises the results of the AL benchmark for each of the three different classifiers used for the supervised baselines. We conclude that switching to a supervised hot-learner significantly improves performance over the continued use of an unsupervised warm-up learners. Among hot-learners, however, there is no clear best policy.

Furthermore, we can see that random sampling as the warm-up learner leads to a faster improvement in model performance (i.e., better performance for smaller labeled pool sizes) compared to anomaly detection methods. This observation aligns with previous considerations on the poor performance of anomaly detection methods (Section 4.2). Since these methods fail to detect illicit instances, they are ineffective at querying illicit instances to be added to the labeled pool to improve the performance of a supervised classifier quickly. We observe that elliptic envelope performs above IF (also consistent with previous results).

Table 2 additionally shows that XGBoost and LR temporarily surpass their supervised baseline, i.e., the performance they achieved when trained on the entire train set (Section 4.1. The classifiers perform better when trained only on a sample of the labeled data but eventually converge to their supervised baseline as the labeled pool increases over time. This result can be because the labeled pool consists of the most relevant samples at the beginning of the AL process and, at the same time, the class imbalance increases over time. Laws and Schätze [19] acknowledge that, in some cases, early stopping of an AL process might prevent this model degradation. Note, however, that even if XGBoost and LR surpass their own supervised baselines, they do not surpass the best supervised baseline, which was achieved with the RF model.

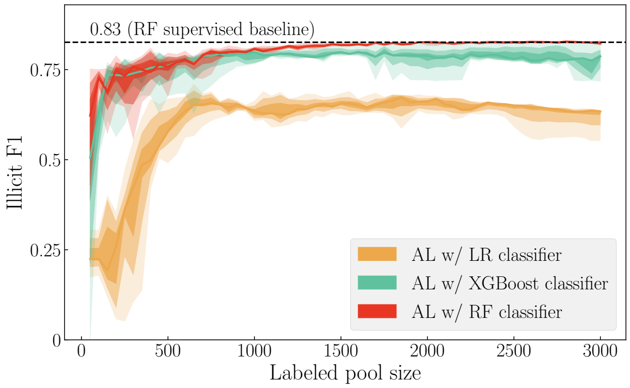

Figure 5: Best AL setups for each classifier and the RF supervised baseline.

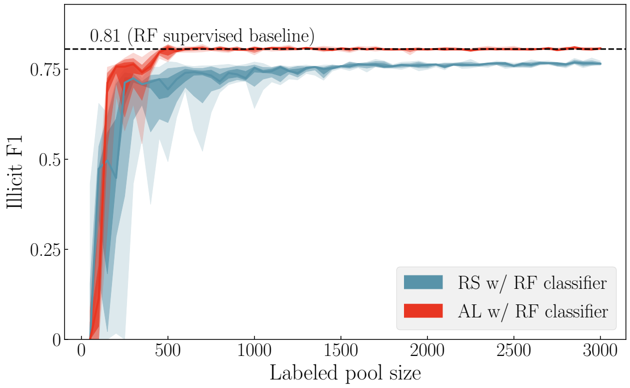

Figure 6: AL versus Random Sampling (RS) with a RF classifier and the RF supervised baseline at 2% illicit rate.

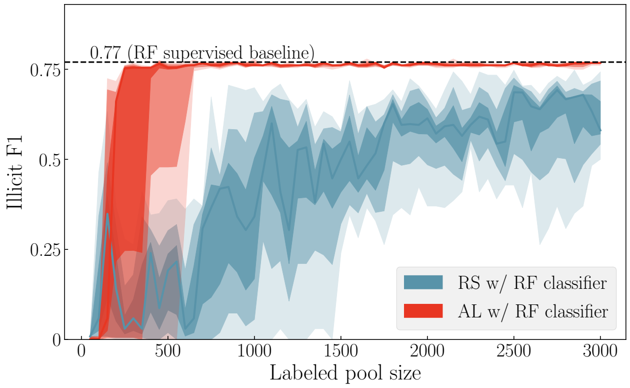

Figure 7: AL versus Random Sampling (RS) with a RF classifier and the RF supervised baseline at 0.5% illicit rate.

Figure 5 shows the performance over time of the best AL setup for the three classifiers tested. For comparison, it also includes the performance achieved by the best supervised baseline, the RF supervised baseline. With the presented AL setup, all classifiers

Table 2: Average illicit F1-score over five runs for each AL setup, consisting of an unsupervised warm-up learner, an optional supervised hot learner and the classifier that is evaluated on the test set. We compare the results to each classifier's respective supervised baseline (Section 4.1). Results are ordered by the illicit F1-score with 3000 labels. Best values for each labeled pool size across classifiers are highlighted in bold.

Query strategies		**Classifier**	**Labeled pool size**					**Supervised baseline**
Warm-up learner	**Hot learner**		**200 (0.7%)**	**500 (1.7%)**	**1000 (3.3%)**	**1500 (5%)**	**3000 (10%)**	
isolation forest	uncertainty sampling	random forest	0.75	0.75	0.80	**0.82**	**0.83**	0.83
random sampling	uncertainty sampling		0.73	0.75	**0.81**	**0.82**	0.82	
elliptic envelope	uncertainty sampling		0.65	**0.77**	0.80	**0.82**	0.82	
isolation forest	expected model change		0.56	0.61	0.77	0.79	0.81	
random sampling	expected model change		**0.76**	**0.77**	0.78	0.78	0.81	
elliptic envelope	expected model change		0.60	0.72	0.76	0.77	0.81	
random sampling	–		0.74	0.76	0.76	0.78	0.80	
elliptic envelope	–		0.50	0.53	0.56	0.65	0.70	
isolation forest	–		0.67	0.65	0.59	0.63	0.62	
isolation forest	uncertainty sampling	XGBoost	0.67	**0.77**	0.80	0.79	0.80	0.76
elliptic envelope	expected model change		0.65	0.75	0.77	0.75	0.79	
random sampling	expected model change		0.70	0.75	0.79	0.80	0.78	
isolation forest	expected model change		0.60	0.75	0.77	0.76	0.75	
elliptic envelope	–		0.53	0.64	0.53	0.61	0.68	
elliptic envelope	uncertainty sampling		0.62	0.62	0.64	0.80	0.64	
random sampling	uncertainty sampling		0.72	0.76	0.64	0.60	0.64	
random sampling	–		0.66	0.58	0.75	0.74	0.59	
isolation forest	–		0.38	0.38	0.46	0.44	0.57	
isolation forest	expected model change	logistic regression	0.22	0.59	0.63	0.66	0.62	0.45
elliptic envelope	expected model change		0.20	0.48	0.61	0.61	0.61	
random sampling	expected model change		0.44	0.54	0.58	0.64	0.60	
elliptic envelope	uncertainty sampling		0.41	0.52	0.63	0.63	0.60	
isolation forest	uncertainty sampling		0.37	0.53	0.61	0.60	0.58	
random sampling	uncertainty sampling		0.40	0.50	0.57	0.58	0.55	
random sampling	–		0.36	0.36	0.36	0.37	0.39	
elliptic envelope	–		0.28	0.25	0.24	0.24	0.22	
isolation forest	–		0.25	0.24	0.29	0.21	0.02	

stabilize after 1000 labels, with RF and XGBoost exhibiting faster performance increase. RF reaches its baseline's performance with only 5% of the original labels, or 1500 out of the original 30000 labels (Figure 5). We can even see a near-optimal performance with as few as 500 labels.

From Table 2, we can observe that the random sampling baseline achieves a similar performance to the more sophisticated AL strategies. Our intuition is that the classifier will start approaching good performances when the labeled pool includes a sufficient number of illicit instances and, because the dataset has approximately 10% of illicit cases, random sampling can quickly reach that sufficient number.

In reality, financial crime is extremely rare among licit transactions, and thus datasets are highly imbalanced Sudjianto et al. [37]. Since we are interested in the practical relevance of AL, we compare the best performing AL setup against random sampling in a dataset with a higher, more realistic class imbalance. Specifically, we apply a random undersampling of the minority class of the Ellipic dataset to achieve illicit rates of 2% and 0.5%. The results are plotted in Figures 6 and 7, respectively. For comparison, we indicate the RF supervised baseline performance at the respective reduced fraud rates.

As expected, the AL query strategies increasingly outperform random sampling as imbalance increases. For highly imbalanced datasets, the best setup uses random sampling (warm-up) followed by uncertainty sampling (hot learner).

5 CONCLUSION

In this study, we conducted experiments to detect illicit activity on the Bitcoin transaction dataset released by Elliptic. Using a supervised setting similar to Weber et al. [42] as our baseline, we studied the detection ability of machine learning models in a more realistic setting with restricted access to labels, using unsupervised methods, and Active Learning (AL).

Our results indicate that unsupervised anomaly detection methods have poor performance, and we present evidence that anomalies in the feature-space are not indicative of illicit behaviour. This finding highlights that experiments conducted on (partially) synthetic data can be misleading and emphasizes the importance of conducting experiments on real-life datasets to draw reliable conclusions.

To improve upon the unsupervised performance, we studied the case where few labels can be obtained by using AL and determined the minimum amount of labeled instances necessary to achieve a performance close to the best supervised baseline. This setting is realistic and akin to asking money laundering analysts to review cases that an AL model indicates as informative. We obtained similar performance to the best supervised baseline by using just a few hundred labels (5% of the total).

It remains to explore if the distribution of classes that we found in the Bitcoin dataset holds for other real-life datasets and different labeling strategies. Furthermore, given the need for proper AML processes in the entire financial system, it is crucial to conduct similar benchmarks on other verticals such as bank transfers, deposits or loans, using real datasets with proper labels.

ACKNOWLEDGMENTS

We want to thank the entire Feedzai Research team, who always gave insightful suggestions. In particular, we want to give special thanks to Marco Sampaio, Ricardo Barata, and Miguel Leite for their support with the AL experiments.

Note on reproducibility

The code to reproduce the results presented in Sections 4.1 and 4.2 is available online at https://github.com/feedzai/research-aml-elliptic. We do not include the code for the AL experiments for intellectual property purposes.

REFERENCES

[1] Magnus Almgren and Erland Jonsson. 2004. Using active learning in intrusion detection. *Proceedings of the Computer Security Foundations Workshop* 17, 88–98.

[2] Ricardo Barata, Miguel Leite, Ricardo Pacheco, Marco O. P. Sampaio, João Tiago Ascensão, and Pedro Bizarro. 2021. Active learning for online training in imbalanced data streams under cold start. arXiv:cs.LG/2107.07724

[3] Massimo Bartoletti, Barbara Pes, and Sergio Serusi. 2018. Data mining for detecting Bitcoin Ponzi schemes. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 75–84.

[4] Claudio Bellei. 2019. The Elliptic Data Set: opening up machine learning on the blockchain. *Medium* (Aug. 2019). https://medium.com/elliptic/the-elliptic-data-set-opening-up-machine-learning-on-the-blockchain-e0a343d99a14

[5] Ramiro Daniel Camino, Radu State, Leandro Montero, and Petko Valtchev. 2017. Finding Suspicious Activities in Financial Transactions and Distributed Ledgers. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 787–796.

[6] Fabrizio Carcillo, Yann-Aël Le Borgne, Olivier Caelen, and Gianluca Bontempi. 2018. Streaming active learning strategies for real-life credit card fraud detection: assessment and visualization. *International Journal of Data Science and Analytics* 5, 4 (2018), 285–300.

[7] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41, 3 (2009), 1–58.

[8] Tianqi Chen 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*. 785–794.

[9] Zhiyuan Chen, Amril Nazir, Ee Na Teoh, Ettikan Kandasamy Karupiah, et al. 2014. Exploration of the effectiveness of expectation maximization algorithm for suspicious transaction detection in anti-money laundering. In *2014 IEEE Conference on Open Systems (ICOS)*. IEEE, 145–149.

[10] Zhiyuan Chen, Le Dinh Van Khoa, Ee Na Teoh, Amril Nazir, Ettikan Kandasamy Karuppiah, and Kim Sim Lam. 2018. Machine learning techniques for anti-money laundering (AML) solutions in suspicious transaction detection: a review. *Knowledge and Information Systems* 57, 2 (2018), 245–285.

[11] Shubhomoy Das, Weng-Keen Wong, Thomas Dietterich, Alan Fern, and Andrew Emmott. 2016. Incorporating expert feedback into active anomaly discovery. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 853–858.

[12] Xinwei Deng, V Roshan Joseph, Agus Sudjianto, and CF Jeff Wu. 2009. Active learning through sequential design, with applications to detection of money laundering. *J. Amer. Statist. Assoc.* 104, 487 (2009), 969–981.

[13] Rémi Domingues, Maurizio Filippone, Pietro Michiardi, and Jihane Zouaoui. 2018. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition* 74 (2018), 406–421.

[14] Zengan Gao. 2009. Application of cluster-based local outlier factor algorithm in anti-money laundering. In *2009 International Conference on Management and Service Science*. IEEE, 1–4.

[15] Nico Görnitz, Marius Kloft, Konrad Rieck, and Ulf Brefeld. 2009. Active learning for network intrusion detection. In *Proceedings of the 2nd ACM workshop on Security and artificial intelligence*. 47–54.

[16] Jason Hirshman, Yifei Huang, and Stephen Macke. 2013. Unsupervised approaches to detecting anomalous behavior in the bitcoin transaction network. *3rd ed. Technical report, Stanford University* (2013).

[17] Yining Hu, Suranga Seneviratne, Kanchana Thilakarathna, Kensuke Fukuda, and Aruna Seneviratne. 2019. Characterizing and Detecting Money Laundering Activities on the Bitcoin Network. *arXiv preprint arXiv:1912.12060* (2019).

[18] Asma S Larik and Sajjad Haider. 2011. Clustering based anomalous transaction reporting. *Procedia Computer Science* 3 (2011), 606–610.

[19] Florian Laws and Hinrich Schätze. 2008. Stopping criteria for active learning of named entity recognition. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 465–472.

[20] David D Lewis and Jason Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*. Elsevier, 148–156.

[21] Xurui Li, Xiang Cao, Xuetao Qiu, Jintao Zhao, and Jianbin Zheng. 2017. Intelligent anti-money laundering solution based upon novel community detection in massive transaction networks on spark. In *2017 fifth international conference on advanced cloud and big data (CBD)*. IEEE, 176–181.

[22] Xuan Liu and Pengzhu Zhang. 2010. A scan statistics based Suspicious transactions detection model for Anti-Money Laundering (AML) in financial institutions. In *2010 International Conference on Multimedia Communications*. IEEE, 210–213.

[23] Xuan Liu, Pengzhu Zhang, and Dajun Zeng. 2008. Sequence matching for suspicious activity detection in anti-money laundering. In *International Conference on Intelligence and Security Informatics*. Springer, 50–61.

[24] Leland McInnes, John Healy, and James Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. (2018).

[25] Patrick Monamo, Vukosi Marivate, and Bheki Twala. 2016. Unsupervised learning for robust Bitcoin fraud detection. In *2016 Information Security for South Africa (ISSA)*. IEEE, 129–134.

[26] Patrick M Monamo, Vukosi Marivate, and Bhesipho Twala. 2016. A multifaceted approach to bitcoin fraud detection: Global and local outliers. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 188–194.

[27] Brian Monroe. 2020. Ousted Danske Bank CEO faces nearly $400 million lawsuit tied to historic money laundering scandal. https://www.acfcs.org/ousted-danske-bank-ceo-faces-nearly-400-million-lawsuit-tied-to-historic-money-laundering-scandal/ Library Catalog: www.acfcs.org Section: Uncategorized.

[28] Financial Crimes Enforcement Network. 2019. Application of FinCEN's Regulations to Certain Business Models Involving Convertible Virtual Currencies | FinCEN.gov. https://www.fincen.gov/resources/statutes-regulations/guidance/application-fincens-regulations-certain-business-models

[29] Laura Noonan, Stefania Palma, and Kadhim Shubber. 2020. The 1MDB scandal: what does it mean for Goldman Sachs? *Financial Times* (Jan 2020). https://www.ft.com/content/3f161eda-3306-11ea-9703-eea0cae3f0de

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[31] Thai Pham and Steven Lee. 2016. Anomaly detection in bitcoin network using unsupervised learning methods. *arXiv preprint arXiv:1611.03941* (2016).

[32] Thai Pham and Steven Lee. 2016. Anomaly detection in the bitcoin system-a network perspective. *arXiv preprint arXiv:1611.03942* (2016).

[33] Saleha Raza and Sajjad Haider. 2011. Suspicious activity reporting using dynamic bayesian networks. *Procedia Computer Science* 3 (2011), 987–991.

[34] Burr Settles. 2009. *Active learning literature survey.* Technical Report. University of Wisconsin-Madison Department of Computer Sciences.

[35] Burr Settles, Mark Craven, and Soumya Ray. 2008. Multiple-instance active learning. In *Advances in neural information processing systems*, Vol. 20. 1289–1296.

[36] Jack W Stokes, John Platt, Joseph Kravis, and Michael Shilman. 2008. Aladin: Active learning of anomalies to detect intrusions. (2008).

[37] Agus Sudjianto, Sheela Nair, Ming Yuan, Aijun Zhang, Daniel Kern, and Fernando Cela-Díaz. 2010. Statistical methods for fighting financial crimes. *Technometrics* 52, 1 (2010), 5–19.

[38] Jun Tang and Jian Yin. 2005. Developing an intelligent data discriminating system of anti-money laundering based on SVM. In *2005 International conference on machine learning and cybernetics*, Vol. 6. IEEE, 3453–3457.

[39] European Union. 2018. Directive (EU) 2018/843 of the European Parliament and of the Council of 30 May 2018 amending Directive (EU) 2015/849 on the

prevention of the use of the financial system for the purposes of money laundering or terrorist financing, and amending Directives 2009/138/EC and 2013/36/EU. *Official Journal of the European Union* L 156 (June 2018), 43–74.

[40] Xingqi Wang and Guang Dong. 2009. Research on money laundering detection based on improved minimum spanning tree clustering and its application. In *2009 Second international symposium on knowledge acquisition and modeling*, Vol. 2. IEEE, 62–64.

[41] Yingfeng Wang, Huaiquing Wang, Caddie Shi Jia Gao, and Dongming Xu. 2008. Intelligent money laundering monitoring and detecting system. In *European, Mediterranean and Middle Eastern Conference on Information Systems 2008*. Brunel University, 1–11.

[42] Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. 2019. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591* (2019).

[43] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[44] Yue Zhao, Zain Nasrullah, and Zheng Li. 2019. PyOD: A Python Toolbox for Scalable Outlier Detection. *Journal of Machine Learning Research* 20, 96 (2019), 1–7.