University of Zurich[UZH]

MASTER THESIS – Communication Systems Group, Prof. Dr. Burkhard Stiller

Identifying Anomalous Transactions in Blockchains

Dmytro Polyanskyy
Zurich, Switzerland
Student ID: 16-909-160

Supervisor: Rafael Hengen Ribeiro, Eder John Scheid
Date of Submission: September 20, 2021

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

ifi

Master Thesis
Communication Systems Group (CSG)
Department of Informatics (IFI)
University of Zurich
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland
URL: http://www.csg.uzh.ch/

Abstract

As criminals have become more sophisticated in the manner they launder their illicit funds, financial institutions and regulators across the world have been quick in their rush to adapt much more stringent Anti Money Laundering (AML) controls. Unfortunately, AML laws oftentimes come at the expense of the most financially vulnerable in society [1]. In fact, many banks today would rather reject low-value or low-income customers (whether officially or bureaucratically) than risk high costs or fines associated with complicated AML procedures.

The relatively new phenomenon of cryptocurrencies has lowered the regulatory barrier and cost for both domestic transactions as well as international remittances for people all over the world. At the same time, it is important to acknowledge that money laundering and illegal transactions do occur on the blockchain. However, the nature and openness of the blockchain has presented an opportunity for machine learning algorithms to make the financial system safer by detecting and tracing such illicit funds moving throughout the network.

This thesis is motivated to improve the AML process for both financial institutions, businesses, as well as ordinary people. In particular, a machine learning model (XGBoost) is presented that not only depicts a robust way to detect anomalous transactions on the Bitcoin blockchain, but also to explain what drives its underlying decisions. With that said, having an accurate model alone is not enough for such an endeavour due to the regulatory landscape surrounding AML laws. In fact, explaining why a model arrived at the result it did - in particular, why something was flagged as an anomaly is a salient of a concept as the performance metrics of the model itself.

Acknowledgments

I would like to express my gratitude to my supervisors, Rafael Hengen Ribeiro and Eder John Scheid for their continuous support, guidance, feedback, valuable input and related work throughout the whole development of this Master Thesis. Furthermore, I would also like to acknowledge Elliptic: Blockchain Analytics & Crypto Compliance Solutions for their release of a very comprehensive Bitcoin dataset which made this work possible. I would also like to express my appreciation towards the Communication Systems Group and specifically Prof. Dr. Burkhard Stiller for the opportunity to conduct this Master Thesis with his department and for his support throughout my entire Master's degree.

Zusammenfassung

Straftäter im Bereich Geldwäscherei basieren auf immer raffinierteren Methoden aufgrund dieser Fonds, Finanzinstitute und Aufsichtsbehörden auf der ganzen Welt der Bekämpfung hinterherhinken, dies meist zu Lasten von wirtschaftlich schwächer gestellten Personen. Dies führt bedauerlicherweise dazu, dass Banken Neukunden lieber ablehnen, anstatt hohe regulatorische Kosten oder sogar Bussgelder zu riskieren. Die neue Asset Klasse Kryptowährungen bieten ebenfalls Möglichkeiten zur Umgehung von Geldwäschereigesetzen.

Transaktionen via Kryptowährungen bieten jedoch einen entscheidenden Vorteil: Sie finden auf der Blockchain statt. Somit können maschinellen Lernalgorithmen dazu beitragen, das Finanzsystem sicherer zu machen, indem solche illegalen Gelder, die sich im Netzwerk bewegen, erkannt und verfolgt werden können. In der vorliegenden Arbeit wird ein Modell für maschinelles Lernen (XGBoost) vorgestellt, welches nicht nur eine robuste Methodik, um anomale Transaktionen in der Blockchain zu erkennen, verfolgt sondern ebenfalls versucht zu erklären, auf welchen Vorkommnissen die zugrunde liegenden Entscheidungen für diese Anomalien beruhen. Nur ein genaues Modell zu haben reicht für ein solches Unterfangen aufgrund der regulatorischen Landschaft rund um AML nicht aus. Wichtiger ist zu erklären, wie ein Modell zu den jeweiligen Ergebnissen gelangt ist, insbesondere im Zusammenhang mit den besprochenen Anomalien.

Contents

Chapter 1

Introduction

Anomaly detection has long been heralded as a salient method for fraud detection and its subsequent prevention within financial systems. Within the scope of this thesis, we define an anomaly as something that is irregular or unlikely to occur by honest participants of a financial network. Criminals who deal with the proceeds of crime – or in other words, commit money laundering offenses are oftentimes anomalous in their activities. Thus, as others in this field have done, we can use this anomalous behavior as a *'proxy'* for suspicious financial behavior [2].

This thesis will focus on detecting anomalies on Bitcoin network utilizing supervised machine learning algorithms. By detecting suspicious activity within nodes, it is possible to uncover illicit activity on the network and prevent large monetary damage to victims of crime – both to private individuals as well as corporations. It is also possible to lower the costs associated with Anti Money Laundering (AML) regulations – which are oftentimes passed on towards ordinary people and have a disproportionate negative effect on lower income households [1].

We present methods that seek to pinpoint more accurately cases of financial crime while also keeping false negatives at a reasonable level. Such constraints are in the interests of feasibility from a compliance standpoint. In addition, for the purposes of AML and being compliant with global regulations, the models presented in this paper must be explainable or at the very least endeavour to make a sophisticated effort in justifying why they are actually explainable.

1.1 Description and Motivation for Work

The ultimate goal of this thesis is twofold. The first is to find viable ways to detect anomalies on the blockchain, while the second is also to do have clear justification behind why a transaction is considered an anomaly. Currently many low income households are "unbanked" and therefore they bear the brunt of ever-so tightening AML regulations [1]. This is because traditional banks tend to reject high-risk, low-income clients rather than

risk high fines for being uncompliant with AML procedures. Improving AML on the blockchain by detecting anomalies through Machine Learning is an opportunity for all of society to make payments accessible to everyone in a safe, controlled and resilient manner [1]. The most noteworthy goals of the thesis will be as follows:

1. An objective overview of the current approaches for anomaly detection as well as past publications
2. Reproduce and analyze the Machine Learning methods presented in the first public paper on the Elliptic Dataset
3. Propose a new/better method which increases detection of anomalies, but is also feasible to be used for AML purposes from a regulatory standpoint (e.g., Explainable AI - XAI)
4. Discover and target the most salient features/important variables that differentiate licit transactions from illicit ones

1.2 Thesis Outline

This Master Thesis will be organized in the following manner:

1. Background:
    - Brief introduction to Bitcoin, its architecture and the misconceptions surrounding the technology
    - What is Money Laundering?
    - Why should we care about Money Laundering on the blockchain?
2. Related Work:
    - Various Clustering Algorithms (Agglomerative, K-means), Graphic Centric Analysis, Mahalanobis distance, U-SVM
    - K-means, Random Forest
    - Geographic Identification of Bitcoin clients
    - Explainable Artificial Intelligence (XAI) in Machine Learning
    - Why anomaly detection requires good explainable and interpretable models
3. Implementation - Frameworks used and the reasoning for doing so:
    - Exploratory Data Analysis of dataset
    - Ensemble Methods in Machine Learning
    - Feature Importance
4. Evaluation - Results of the Machine Learning Model(s):

- Performance metrics of Machine Learning model(s)
- Analysis of Interpretability of model(s)
- Limitations of experiments

Chapter 2

Background

Depending on who you ask - Bitcoin can either be viewed as one of the biggest digital developments since the Internet, or it can also be viewed as a tool for criminals to launder funds. This chapter will give a brief background on Bitcoin and also aspects of money laundering. Not only will this give some background to the reader about these concepts, but it will also seek to motivate issues currently surrounding these closely interlinked topics.

2.1 Bitcoin

There were many ideas of a digital asset or electronic money in the 1990's - ranging from HashCash to Bit Gold and B-money. However, it was not until 2008, when the pseudonymous Satoshi Nakamoto released his paper: *Bitcoin: A Peer-to-Peer Electronic Cash System* that the first mainstream cryptocurrency was actually launched [3]. With his protocol focusing on a decentralized architecture, Satoshi was able to revolutionize payments between users. Bitcoin not only allowed users to make irreversible transactions with each other without the need of a trusted third party, but also took care of minting currency without the use of a central authority.

2.1.1 Transacting in Bitcoin

Through its distributed peer-to-peer network, people are able to transfer Bitcoin in a decentralized manner and also have every transaction stored and available for anyone to review at any time. The fact that anyone is now able to send a transaction to anyone in the world without needing to wait for the servicing schedule of a money payment service is at the heart of Bitcoin. Cryptographic proof was able to replace the typical leap of faith when it came to trust and time that people attributed to countless third parties; Satoshi's creation made sending and receiving funds transparent and controllable in almost every manner - ranging from the involved wallet addresses, to the fees, and also the delivery time. Satoshi defined Bitcoin as a chain of digital signatures - where the owner would be

able to transfer the coins to another person by digitally signing a hash of the previous transaction with the public key of the intended recipient.

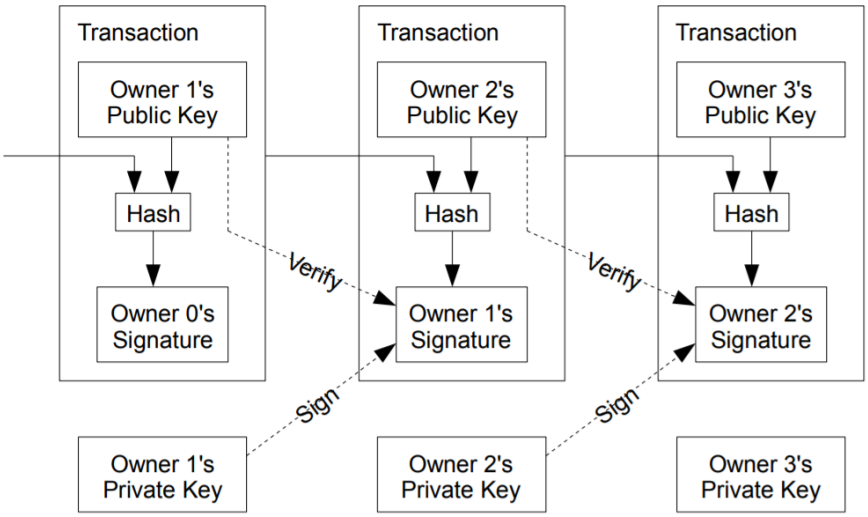

Figure 2.1: Bitcoin - A Chain of Digital Signatures [3]

2.1.2 Minting Bitcoin

New Bitcoins are minted by miners - individuals or groups who exchange their computing power which run complex mathematical algorithms in order to find the next valid block. Blocks contain transactions from the memory pool (mempool) – which are essentially transactions users broadcast that they would like to send. The job of the miners is to ensure the transaction is valid and only then include it in the next block. Usually, the miner will prioritize the transactions which contain the highest fee. Thus, as a result, users of the Bitcoin network can manually specify a fee at which they want to send their transaction based on how fast they need the transaction to be confirmed. In turn, miners who find a valid block are rewarded with a chunk of new Bitcoin (in addition to the transaction fees) – called the block reward. Roughly every four years, the block reward is halved (currently at 6.25 BTC). Finding a new block takes roughly 10 minutes, and the respective difficulty is adjusted, so that production of Bitcoin will eventually reach zero. This is currently predicated to be in 2033 as Bitcoin has a finite cap on the number of coins at 21 million [4].

2.1.3 UTXO Currencies

Bitcoin is based on the UTXO (unspent transaction output) framework. Many cryptocurrencies are based on such a mechanism, with the most popular being Bitcoin. Other notable ones are Bitcoin Cash, Litecoin, ZCash. In general however, UTXO based cryptocurrencies have certain features, characteristics and also complexities that other cryptocurrencies may not have. As a result, this thesis will primarily focus on the Bitcoin

blockchain - the largest and most popular (by market-cap and adoption) UTXO based cryptocurrency to date. Aside from being the most popular cryptocurrency, it also has the most research and analytic work behind it when it comes to anomaly detection.

UTXO is a way to manage the transaction balances on an append only ledger. Rather than the traditional account based system that we are used to in traditional banking (and also used by some other cryptocurrencies such as Ethereum, Ripple), UTXO's are mainly comprised of two pieces of information – the Ownership Data and Amount [5]. However, while this may seem similar to an account based system, in such a framework, there are actually no set accounts or wallets – instead the amount someone can relay as a transaction entirely depends on the series of unspent transaction outputs one has available to them. In essence, when a transaction is sent, it consumes already existing UTXO's and creates new UTXO's in their place.

UTXO's allow each Bitcoin transaction to be uniquely identified by an ID that is made up of a series of inputs and unspent transaction outputs. This also means that a user can use a single or a combination of multiple inputs to send a transaction (and can even specify which inputs to use for a transaction). UTXO's have some privacy preserving mechanisms since one can always generate a new address to accept coins. However, oftentimes when sending, depending on the wallet which the person uses, unspent coins are combined from various inputs. As a result, it is possible to link different coins to a single owner in many cases where additional steps of privacy increasing behaviour are not actively taken (e.g., by running a local node).

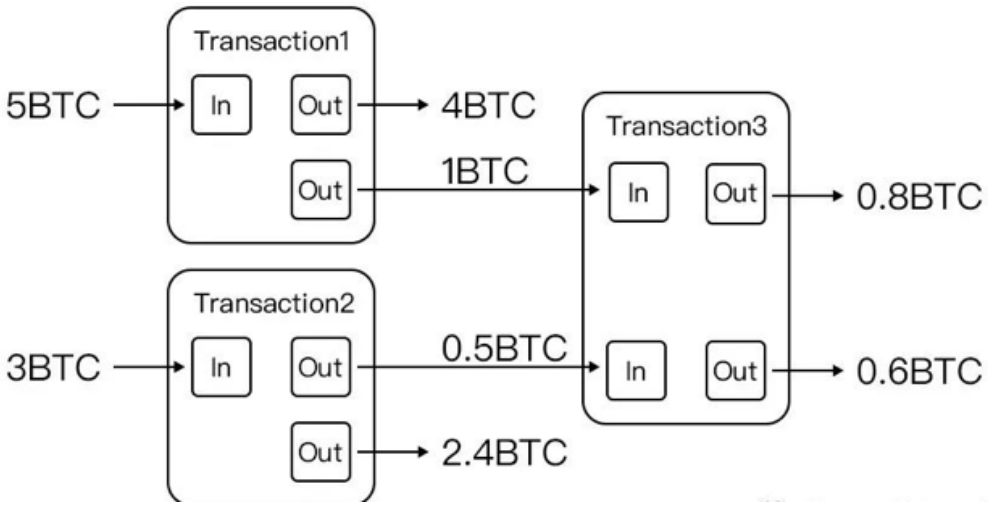

Figure 2.2: UTXO Model [5]

2.1.4 Implications of Bitcoin on the Financial System

Lastly, although it is easy to recognize the power behind Bitcoin's decentralized, append-only ledger system, it is just as important to recognize how the same decentralization has also removed the traditional safeguards that we are accustomed towards. The architecture

of Bitcoin makes it so that nobody can reverse a transaction even if an entire group agrees that the funds were stolen from an honest party; this makes dealing with the recovery or reversal of proceeds of crime much more difficult. Furthermore, since there is no initial onboarding or KYC (Know Your Customer) process, it is not possible to prevent an adversary or a certain group from accessing or using the Bitcoin network. While it is possible for individual miners or pools opt to not include a certain broadcasted transaction in the block, it is generally not practical to prevent a person or group from using the network for any significant period of time.

2.1.5 Misconceptions around Bitcoin

Today there are many misconceptions which surround Bitcoin. In popular culture, Bitcoin is often portrayed as an anonymous means of value transfer, whereas it is actually pseudonymous. The fact of the matter is that Bitcoin is actually extraordinarily transparent – every single payment that has ever been sent to any address can be visible by anyone at anytime and from anywhere in the world. Furthermore, this record of transfer is immutable – making it impossible for anyone to ever modify or delete entries on the ledger after being confirmed by the network.

Ultimately, while Bitcoin is pseudonymous, it does not mean that it is completely private. In fact, identities of people on the network can be readily uncovered when examining a variety of variables - ranging from transactions hashes, to wallet addresses and even IP addresses [6] [7]. More significantly, Bitcoin is often portrayed as an instrument which criminals use in order to evade detection by law enforcement when conducting illegal financial operations [8]. The usage of Bitcoin for criminal activities has been infamously highlighted many times over the last decade and has cast a dark shadow on the cryptocurrency as a result. However, while it is not possible to prevent criminals from using and transacting on the Bitcoin network (similarly to fiat systems), it is definitely possible to both uncover and trace their illicit transactions.

With the help of anomaly detection machine algorithms, such illicit transactions can be discovered on the blockchain, leading to many criminals being exposed for their transactions – including both the ones who do indeed hide in plain site and also ones who employ more sophisticated privacy measures on the network. Furthermore, because the Bitcoin network is accessible to everyone and is an append only ledger, the types of forensic analysis that is possible to conduct through machine learning is much more detailed. For instance, in a traditional fiat financial system, many banks must enforce rules on their own (usually within their own organizations) and once money is outside of their control, another entity is responsible; as a result, it is very difficult to get a complete picture of money flow without the cooperation and coordination of all the involved financial institutions. This is even more significant if the transactions in question also traverse national borders and require cooperation from various courts, governments and law enforcement agencies.

2.1.6 Why Bitcoin is not Anonymous?

While the actual blockchain does not store names and identifying personal information, the transactions one makes with their address can be linked to real-world identities in many cases. For instance, a transaction made to a merchant, a cryptocurrency exchange, or even a remittance to a family member allows the possibility of a connection to a real-world identity to be revealed. In essence, Bitcoin transactions leave one of the most detailed and honest paper trails for anyone to follow as every single transaction is publicly announced without fail. Breaking up the paper trail by sending through a myriad of addresses or using "tumblers" to obfuscate the origins of the Bitcoin exist, but can also be ultimately identified; these strategies will be addressed later in the thesis. Finally, it has been proved and demonstrated over the years that tracing money using the blockchain is possible and uncovering the identity of the users behind the addresses is possible in conjunction with available information that is external to the Bitcoin network [9]. As a result, identifying anomalous transactions on the blockchain is a very fruitful task for law enforcement and governments – since it often allows them to identify who is behind the illicit transaction and potentially recover illicitly obtained funds.

Perhaps one of the most prominent ways to uncover who is behind a transaction is in the onboarding/offboarding process of Bitcoin. When an entity wants to buy or sell Bitcoin, they usually do it from a centralized exchange. Most trustworthy exchanges (by volume and users) require users to undergo an in-depth KYC/AML process that asks for documents such as passports, ID cards, tax bills, residence documents, etc in order to allow the person to buy or sell Bitcoin. Thus, information can be secured about both where the Bitcoin came from as well as the involved parties during the onboarding/offboarding stages. Furthermore, because of the extremely open nature of the network, it oftentimes leads to other discoveries of other bad actors (e.g., criminals conducting transactions with other criminals) since "bad Bitcoins" have a tendency to stay in bad neighbourhoods [6].

2.2 Money Laundering

While money laundering has various levels of sophistication, the exact definition of it is when "a person or business deals in any way with another entities benefits from a crime" [10]. The impact of money laundering stretches on a global scale and the consequences of it can range from an individual or corporation being defrauded all the way to an entire government being undermined. The United Nations Office on Drugs and Crime estimated that between USD 800 billion and USD 2 trillion worth of money is laundered every year [11]. Amounting to around 5% of global GDP and financing both criminal and terrorist organization, this makes it one of the most prolific crimes in our world [11]. Classically, money laundering primarily consists of three main steps [11]:

1. `Placement` – the injection of unlawful money into the financial system
2. `Layering` – the movement of illicit funds through the financial system in order to disguise their origin and ownership. This obfuscation step is often the most crucial part of the scheme

3. `Integration` – the reintroduction or reinvestment of funds into the legitimate economy

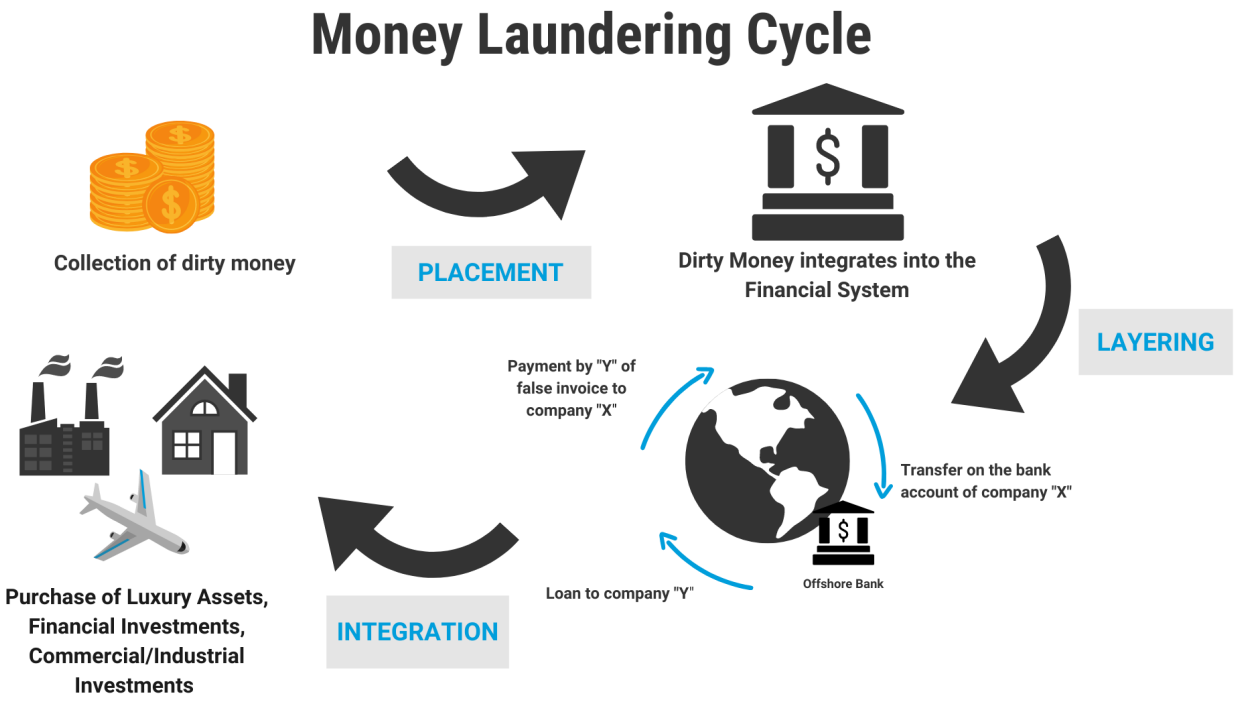

Figure 2.3: Money Laundering Cycle according to the United Nations, 2018 [11]

Combatting money laundering in a fiat currency system poses many challenges. Oftentimes, policies of KYC/AML can only detect the most blatant type of money laundering. Varying levels of KYC in certain regions of the world allow criminals to target regions where KYC can be circumvented entirely or where it is weak. Furthermore, given this global dimension of money laundering operations, cooperation from different countries is often paramount in order to successfully prosecute a criminal organization. However, in reality, cross border cooperation is difficult to establish due to hurdles from legal, monetary and regulatory standpoints. Oftentimes, such transnational cooperation from various parties over the globe is too unfeasible to conduct in order to investigate money laundering in a timely and cost effective fashion.

In a fiat money laundering operation, criminals will often use a network of money mules to help move proceeds of crime [12]. Money mules are used because they are able to move relatively small quantities of capital with a very low risk of being detected. Due to this, the head (or top) of the criminal organization remains both at low risk of getting prosecuted and also has a high likelihood of continuing the criminal enterprise since money mules are expendable and easily replaceable [13]. Furthermore, some money mules are not even aware that they are a participant of a criminal operation; a large portion of money mules are often recruited under the guise of a legitimate job advertisements, social media posts or training courses, with a particular focus on young adults and people who would otherwise be looking for quick cash [14].

Furthermore, criminal organizations with various levels of sophistication are often quite aware of the safeguards in place in traditional financial systems to combat money laundering. As a result, criminals have adapted and will often recruit money mules to help launder up to a statutory reporting threshold (e.g., less than 10,000 euros).

While the scale of funds laundered through Bitcoin is eclipsed by that of traditional fiat systems, it disproportionately receives a bad name for its "assumed ease" to do so. While it may be difficult and ignorant to turn a blind eye towards Bitcoin's presence in dark-web marketplaces and in various cybercrime activities (e.g., carding, ransomware), it must also be recognized that many prolific cyber-criminals have been apprehended after hiding their transactions in the Bitcoin blockchain [15].

Furthermore, investigating money laundering through Bitcoin has its advantages; certain aspects of the investigation are often easier in comparison to traditional fiat currency systems as less cooperation with different banks and intermediaries is required. This is due to the open ledger – anyone can view any transaction at any time from any place in the world. Furthermore, since the Bitcoin network provides everyone with access to its transaction graph – it is possible to uncover and also follow anomalous/suspicious events which are isolatable due to their rare occurrences.

Almost always, criminals will eventually want to convert their cryptocurrency into tangible assets. More often than not, various tools, such as prepaid credit cards that can be loaded with cryptocurrency are used [16]. The use of such methods is growing in popularity due to relatively relaxed KYC measures when onboarding customers for these cards. However, by leveraging machine learning methods, such techniques are able to infer complex patterns from historical data where money laundering was identified. This helps investigators understand the connections which surround those particular instances to help identify futures ones more easily.

2.3 The Enormous Impact of Money Laundering

Money laundering is an ever present and growing problem in our world today. As regulators and governments all over the world attempt to crack down on the issue, they are increasing regulations. There are many rules in place on who financial institutions may conduct business with. More importantly, these regulations also stipulate under which conditions they can conduct their business – both in the fiat space as well as in the cryptocurrency space. This means that financial institutions who are not compliant with all AML procedures are liable to massive fines and can ultimately face lengthy and costly court proceedings.

2.3.1 Costs Arising from Strict Regulations

Unfortunately, the cost associated with the extra regulation imposed on financial institutions is most often passed onto the customer. More specifically, it negatively and disproportionally affects those who are less well off, leading to almost 2 billion people around the world being unbanked according to the World Bank [1] [17]. This is because rather than risk fines and expensive bureaucratic procedures, financial institutions rather opt to refuse certain customers from certain regions. This refusal can take shape in a sort of official ban or by simply making it logistically impractical for the person to pass AML checks (e.g., requiring proof of income for undocumented workers or source of funds for unemployed people). In essence, while criminal organizations such as drug cartels, human smugglers, terrorists launder hundreds of billions of dollars, low-income households, immigrants, refugees and other vulnerable people are left unbanked as they are seen low value plus high risk by many financial institutions [18].

Not only is AML often tricky and expensive for such a person to pass, the transaction fees for using services such as remittances are excruciatingly high. For instance, the World Bank estimates that roughly 7% of remittances goes towards paying the bank's transaction fee [18] [17]. And while it is possible to argue that Bitcoin transaction fees can also reach such amounts when the network activity is high, there is still an opportunity to help lower income households in the aforementioned AML and compliance cost area. This is where many people, such as Weber have realized that while omitting transaction fees on remittances can prove to be challenging, improving the AML process using machine learning to detect anomalies is possible [1]. Not only is this good for the cryptocurrency space in general, but it also offers both a practical and economical advantages for nearly all parties involved (e.g., users, government, regulators, financial institutions).

2.3.2 Deterrence as a Form of Prevention

Having solid AML tools and anomaly detection algorithms in-place can also cut down on criminal activity on the blockchain. If a criminal understands that he or she is unlikely to successfully launder the funds, they are unlikely to use the blockchain to commit crime or as a medium for money laundering in general. If fact, as blockchain analytics companies have been steadily developing new algorithms to identify and track stolen funds, criminals have even returned ill-gotten gains on their own accord. Most often however, the stolen funds are recovered or seized after a person is apprehended by law enforcement.

Poly Network Hack Case

More recently, a phenomenon has taken place where cybercriminals were essentially forced to return stolen funds prior to their real world identities being exposed or them having been arrested by law enforcement. For instance, in August 2021, a decentralized financial network by the name of Poly Network was hacked and roughly $600 million (USD) was stolen from the wallets of the network [19]. The hacker, who exposed a vulnerability in

the contract calls of the platform was able to send funds to a series of personal wallets. Fortunately, for the hacked network, exchanges and various blockchain analytics groups convened together in unison to mention that they would all be monitoring and tracing the funds. Such a revelation prompted the hacker to subtly realize that while he may be able to retain control of the funds in his wallet, he may not be able to actual use the funds or launder it in any way which would not compromise their identity.

Thus, we can see that developing anomaly detection tools can not only help identify crime on the blockchain, but also serve as an extremely strong deterrent from it even initially occurring. This concept of deterrance as prevention can also be extended to many other money laundering schemes - such as ransomware demands and Ponzi schemes.

2.4 Decision Trees

In machine learning, decision trees are the underlying basis for some of the most popular and high performance algorithms. Since the models in our thesis will use decision trees, (e.g., random forest, XGBoost) we describe them here. The basic structure of a decision tree is quite intuitive for anyone to understand and consists of three main elements:

- `Decision Nodes` - these are a binary/boolean question (the ovals)
- `Leaf Nodes` - also referred to as prediction nodes as they are the ones which predict the outcome (the rectangles)
- `Edges` - the connections from one node to another node (the arrows)

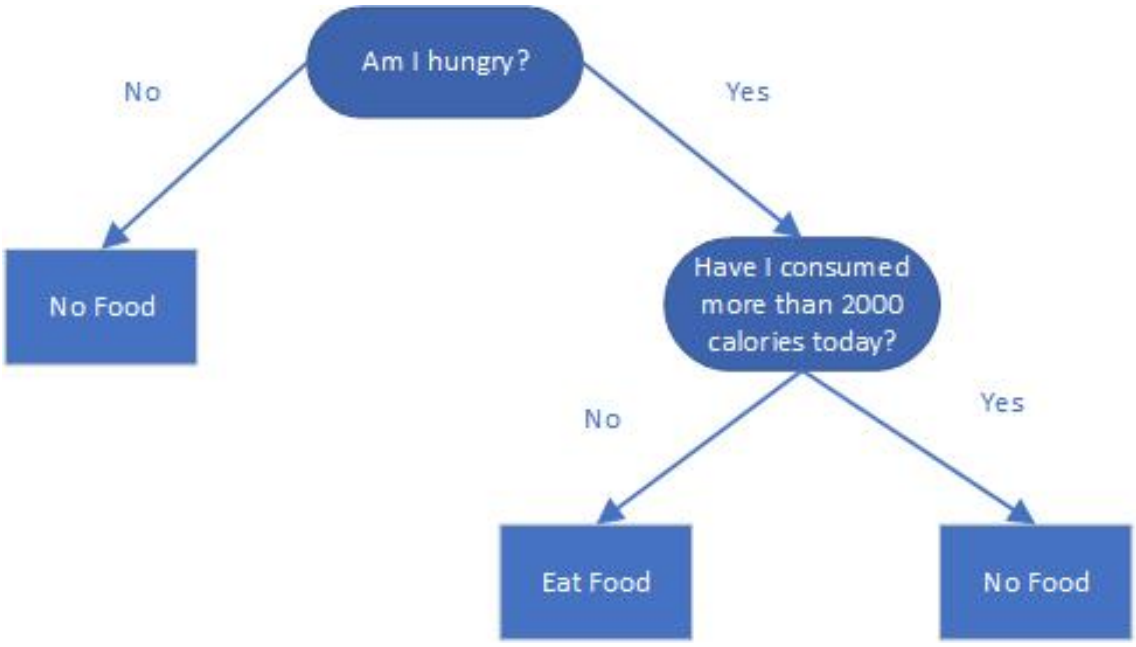

Figure 2.4: A very basic example of a decision tree

The tree is inverted from a classical real life tree. To traverse the tree, you start from the top of the tree (the first level). The next node you visit depends on the decision made at the first node and the process only terminates when you reach a leaf node. In general, traversing a decision tree is a relatively simple task to accomplish; the actual real work is knowing which questions to ask and in what order - or in other words, finding a good split for the tree. In general, the higher the level a feature exists on a tree, the more weight a feature has on a model. Above in Figure 2.4 is a very basic example of a decision tree.

2.5 Interpretable and Explainable AI (XAI)

This section will serve as an introduction to Interpretable and Explainable Artificial Intelligence (XAI) and why it is crucial for our model. In general however, it is important to note that XAI is not needed for every machine learning model. For low risk situations such as a multimedia recommending system, explainable AI is oftentimes both unnecessary and costly. However, Explainable AI is central to our model design for numerous reasons. As stated in the previous section, we need our model to be compliant with AML procedures and regulations around the world; in practice, this inherently implies that we need to be able to explain why a model arrived at the decision it did - particularly if it detected an anomaly.

2.5.1 Anomaly Detection and AML

Anomaly detection on blockchains is a somewhat challenging task due to the fact that most times nodes and transactions come in an unlabelled manner. As of today, not much work specifically on blockchain anomaly detection has been done due to the relative novelty of the cryptocurrency space in general. Furthermore, the oftentimes "black-box" nature various unsupervised models are characterized as, prevent the industry from adopting them for AML purposes on any large scale. This is because we are currently in a regulatory environment where it is not compliant to enforce AML without strict guidelines and well defined metrics; opaque models such as the ones mentioned above are simply not feasible.

There must be a clear explanation of wrongdoing or a basis for suspicion as an anomaly usually implies immediate consequences for a certain party. In essence, what this implies is that while many Unsupervised Methods and more specifically, active learning/graph convolutional network's (GCN's) may prove to be helpful in understanding certain patterns, they are ultimately unable to be put into practice due to constraints arising from the regulatory landscape [1][18]. Furthermore, the robustness of many of these active learning algorithms are questionable. For example, as certain Dark Markets have been shut down by law enforcement, these methods essentially stop working completely due to the complete change in the network structure [1].

While Explainable AI is unable to cover all possible cases of the model in advance, it can allow us to verify the reasoning of the model through its enumerated predictions [20]. Moreover, local explainability (e.g., explanations for specific observations) generally has

more of a useful effect when it comes to justification of a certain individual decision. Global interpretability on the other hand tends to identify general biases in the model or when a more high-level understanding of a particular task is desired.

2.5.2 Explainability (XAI) for Anomalies

Explainability is a critical aspect of our machine learning model (and in machine learing model in blockchain anomaly detection). Explainability is simply the ability of the parameters to justify the results, or in other words answer a few key questions [21]:

1. "Why should I trust the prediction of the model?"
2. "Which key factors ensured the model to be a success?"
3. "What are the most important categories (dependant variables) in a model which influence its decision (and also likewise, the ones that negatively affected its decision)"?

2.5.3 The Importance of Explainability in ML Models

Explainability is not always significant in many machine learning models – especially in a low risk situation. However, in our case, being able to justify why a certain transaction is indeed flagged as an anomaly is very critical. There are many reasons beyond just the legal and regulatory ramifications as to why an anomaly detection method needs to be explainable. In Doshi's paper, the most salient reasons include [20]:

1. `Multi-objective trade-offs` - there are trade-offs that are made which are not fully known. For instance, a decision between the tradeoffs of privacy and accuracy of the prediction
2. `Scientific Understanding` - in general a computer can find relationships that we do not initially see
3. `Ethics` - an explanation can guard against possible discrimination that were not seen initially in the dataset
4. `Mismatched Objectives` - sometimes a model will work to predict an incomplete objective. For instance, to optimize a control without considering adherence principles

Understanding why a machine learning model makes the predictions it does is one of the most crucial aspects of AML regulatory issues. In fact, the ability to explain why a model behaves and predicts the way it does is almost as quintessential as the accuracy itself. In a regulatory environment, being confident of a model's output means understanding the so called theoretical "black box" that machine learning algorithms are oftentimes compared

to. It is not enough to simply trust the results (no matter how accurate they seem), but humans using the outcomes need to understand what an anomaly truly entails. A cryptocurrency exchange or financial provider would quickly find itself in a detrimental situation from a regulatory standpoint if an algorithm flags an account or certain transaction as suspicious and they proceed to shut it down without fully understanding why the detector flagged it. For instance, can we be sure that the algorithm is not biased towards people from a certain region or towards a certain nationality based on previous fraud cases?

2.5.4 Case Study of XAI Importance

Not being able to explain why something is suspicious is a serious limitation. For instance, Mark Weber, who was one of the first people to work with the Elliptic Dataset (which we use as the basis in the Implementation and Evaluation section of this paper) and is quite known in the blockchain anomaly detection field, explains in his presentation of a case where a traditional bank account could not be shut down [18]. In that specific situation, the algorithm that compliance at the bank had used noticed that a client may have some suspicious business transactions; for two years, they were following the suspicious flag, but could not do anything because there was no clear wrongdoing or explanation of why it was suspicious. However, years after the initial flag, it turned out the bank account was involved in the financing of the 2005 London subway terrorist bombing [18]. Although this is definitely a more extreme case, this highlights the dilemma that both financial institutions face as well as the development hurdles data scientists face working in such situations.

When there are a few features in a model, it is quite understandable by humans (especially if the algorithm is based on relatively simple decision trees) – however, as the number of features in a model grows, interpretability of the model by a human declines rapidly. To help humans with this task, there have been many various visualization and analytical tools developed to understand what fuels the decisions of a model. One of the most popular tools employed to attempt to explain classification problems in machine learning are heat maps. However, oftentimes the use of these maps in truly understanding the output is questionable at best since we as humans have limitations when it comes to interpreting visualization data [22].

2.5.5 SHAP and ELI5 Values

In our paper, we elect to use SHAP (Shapley Additive exPlanations) as one of the primary frameworks to help explain our models performance. The model allows us to quantify and understand the drivers of predictions (feature importance) on both a global scale and also for each individual datapoint [23]. For example, while regular feature importance highlights the most critical variables for the global dataset, there may be another set of variables which have an extremely significant predictive power for a subset of transactions. We also implement ELI5 in our work for comparison. ELI5 (colloquially known as “Explain

Like I'm 5") is able to give feature importance's while also is able to explain individual predictions by showing feature weights [24]. A more formal definition and also results of our SHAP and ELI5 experiments can be found in the subsequent Implementation and Evaluation sections of this paper (respectively).

Chapter 3

Related Work

This chapter will discuss published research work on anomaly detection with regards to supervised, semi-supervised and unsupervised machine learning methods. The section will also show research that goes beyond simply identifying anomalies - more specifically, it will show that is it possible to identify users and certain fund flows with the information available on the blockchain.

3.1 Anomaly Detection

Anomaly detection in networks predates Bitcoin itself. Financial institutions have been using different anomaly detection methods as well as scoring models to identify (and prevent) fraudulent activities for many years [2]. However, in recent times, there has been a high interest in research work dedicated towards studying anomaly detection specifically on the blockchain. Although this is a relatively new area of study, principles from decades of anomaly detection in networks can readily be applied for blockchain related research. The common framework applied is to identify those who show atypical (i.e., anomalous) behaviour and highlight them, as these as the people who are likely to be involved in financial crime [25].

For anomaly detection in a network, it is possible to apply any of the three main types of machine learning methods typically applied 1) Supervised, 2) Semi Supervised, 3) Unsupervised.

With that said, there are many advantages and disadvantages to applying various types of machine learning. This section will serve as a discussion on which machine learning methods are applied in current blockchain anomaly detection literature. There will also be discussion on other supplementary methods which aid anomaly detection.

3.2 Supervised Machine Learning

In general, supervised machined learning has been applied on many different Bitcoin transaction datasets with various degrees of success. The challenge that supervised learning poses in this field is the lack of enough labeled datasets. Lorenz et al (2020) addresses the real-world challenges of detecting money laundering using machine learning algorithms that require labels: 1) it is unlikely to ensure with certainty that a dataset can identify all money laundering transactions and, 2) accurate labels are an administrative effort which take too much time and is often very costly [26].

3.2.1 K-means, Kd-trees with Random Forests

Monamo et al (2016), defined two various types of anomalies – global and local [27]. In their global definition of anomaly, they refer to an anomaly that is made with respect to many instances under study – while a local outlier only considers its surrounding neighbourhood. In their global approach, they used trimmed k-means and in their local approach, they used kd-trees. They subsequently assigned class labels as either global or local anomaly present (or not present). With regards to the dataset, the top 1% based on kd and trimmed k-means were labelled as anomalies while the remainder of instances were assigned a normal (licit) label. This was done in order to serve as proxies for licit and illicit transactions so that an adequate baseline could be established for supervised learning techniques.

Interestingly, Monamo finds that out of the 3 models they ran, Random Forest was the best performing classifier despite the class imbalance issue. They had 8 features in their model and Random Forest depicted almost near perfect performance. Furthermore, they show that similar results can be reached by only using 2 features instead of 8 features. Out of all financial attributes, the ones which were significant in predicting outliers were:

- Average value sent
- Total received
- Standard deviation sent
- Standard deviation received

3.3 Semi-Supervised and Unsupervised

3.3.1 Agglomerative Clustering Algorithms

Androulaki et al (2013) present a method to identify 40% of all users in a simulator that mimics the Bitcoin system in a closed setting [28]. This figure even includes users

who adopt recommended privacy measures such as not reusing addresses. They are able to accomplish this by leveraging behaviour-based clustering techniques (K-Means and Hierarchical Agglomerative Clustering algorithms also, 80% accuracy) even when users send transactions to their other pseudonyms (bitcoin addresses) to try and enhance their level of privacy. Change addresses, transaction in certain geographical locations and links to vendors were all shown as a very likely method for an adversary to link someone to their real-world identity. In general, Androulaki shows that two heuristics are responsible for exploiting privacy in Bitcoin [28]:

1. `Heuristic I--Multi-input Transactions`: This multi-input transaction event occurs when u wishes to perform a payment, and the payment amount exceeds the value of each of the available BTCs in u's wallet. When the payment is made from aggregated addresses, it is trivial to see that the input addresses are owned by the same user.

2. `Heuristic II--Shadow Addresses`: The protocol generates a change address to which the sender can claim back the "change" from the transaction. This exposure occurs when a transaction with 2 output addresses contains a new address and one with an old address. Thus it is possible to conclude that the new address is a shadow address.

3.3.2 K-means Clustering, Mahalanobis distance, and Unsupervised Support Vector Machine (SVM) learning

Pham and Lee (2016) propose using unsupervised learning methods to detect anomalies on the Bitcoin blockchain – namely, they use: k-means clustering, Mahalanobis distance, and Unsupervised Support Vector Machine (SVM) learning [2]. Pham and Lee use a two-pronged approach: they parse the data into a user graph, where each user owns a list of addresses and is a node, and the transactions between users are edges. They also use a transaction graph which shows the transactions as the nodes and the Bitcoin flow between transactions as edges. The basis for this two-pronged approach is to be able to identify both suspicious users and also suspicious transactions. By doing so, they can subsequently refer to the results of both graphs and determine whether or not any overlap exists with regards to anomalous activity. For instance, using this "Dual Evaluation", they can reinforce whether both suspicious transactions occurred in both the node study and also the edge study; if a suspicious observation was made in the node study, they would also see if it was made in the edge model study to increase confidence.

Their methods successfully identified 10% of all the known cases of financial loss. More specifically, out of the 30 known cases of financial crime they had, the algorithm successfully identified two known cases of theft and one known case of loss [2]. There is also an indication that it tends to identify large, and more pronounced losses as one of the cases involved a loss of 2600 BTC. As a result, we can see that unsupervised learning on the blockchain is immensely difficult.

Since many of the observations are initially unlabeled in the Bitcoin transaction graph, supervised learning methods are not always feasible for all existing datasets. With that said, the barrage of regulatory issues surrounding AML laws oftentimes also limit unsupervised methods to be used in somewhat meaningful ways. Pham and Lee's (2016) various unsupervised learning methods ranging from k-means clustering to their own modified Unsupervised Support Vector Machine (SVM) technique establish a baseline for anomaly detection on unlabelled datasets, but do not offer insightful explanations into the models decisions. And although it is difficult to establish evaluation metrics in unlabeled data, their proposed visualization evaluation still commands the characteristics of a black box model.

3.4 Uncovering Anomalies

This section motivates the discovery of anomalies and revealing identities of addresses after a machine learning algorithm detects an anomaly. Moreover, it shows why it is important, useful and also possible to identify users on the network even if their addresses are initially pseudonymous.

3.4.1 Graph Centric Analytics

Haslhofer et al (2016) showed that by building an analytics platform called GraphSense, they were able to semantically enrich and understand better the information from transaction graphs [6]. The ability to explore a transaction graph allows for the exploration of transactions and the subsequent characteristics of money flow. Following this, they are able to search for certain path and graph patterns that lead to anomalous transactions on a cryptocurrency network. In their work, they apply heuristics to group addresses in a blockchain cluster which are likely to be owned by the same real-world entities. They observed in their study the presence of around 2,000 super-clusters which were responsible for 23% of all transaction outputs [6]. They also postulate that such centralized clusters are often linked to major darknet markets, gambling rings, exchanges or mining pools which can be tagged and further explored accordingly.

Objectively, it is useful from an analytical standpoint to be able to leverage such graph tools in an investigation which allow searches by tag, transaction identifier, traversal (e.g., shortest path), or cluster. Furthermore, they seem to follow a concept that postulates "bad bitcoins tend to stay in bad neighbourhoods". Haslhofer's paper has shown how blockchain analysis techniques such as the multiple input heuristic and change heuristics are able to be used to detect anomalies. Furthermore, such de-anonymization efforts can also be directly applied to a wide range of other cryptocurrencies.

3.4.2 Bayesian approaches to identify Bitcoin users

Juhasz et al (2018) also show how clients can also be identified by IP addresses [29]. Since Bitcoin transactions must be publicly announced, IP address mappings can be used to narrow down a user's geographic location. They built a naïve Bayes classifier that assigned Bitcoin addresses to the clients who likely were controlling them. They were able to do this because of a known characteristic of Bitcoin transactions – addresses appearing on the input side of the same transaction typically belong to the same client. Furthermore, their model also adds an element to the transaction graph which helps further visualize the flow of bitcoins – geolocation. In general, they also expose the typical privacy issue in Bitcoin; in Figure 3.1 below, the left side depicts the transactions and the input Bitcoin addresses where the Bitcoins are sent from. Typically these Bitcoin addresses and transactions belong to the same user. Next, when a Bitcoin address appears in different transactions (marked by red and by bold), all the Bitcoin addresses can be merged and attributed to the same user.

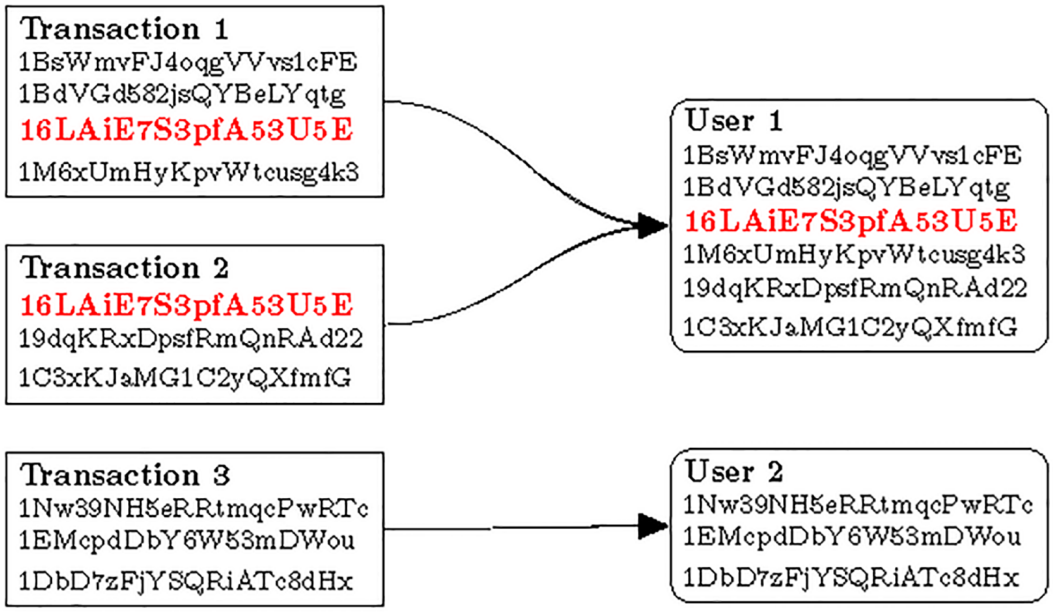

Figure 3.1: Input side client exposed from Juhasz [29]

Juhasz et al also contributed to identifying not only specific users, but also fund flows from one country to another. This is a useful feature for cross-border transactions and explicitly tracing monetary flows between bad actors. For instance, the paper found that the largest key flows occurred between 1) Germany and Argentina, 2) China and the Netherlands, 3) Domestically within the United States [29].

3.4.3 Dealing with Mixing/Tumbling Obfuscation Services

One of the main hurdles law enforcement agencies face when investigating financial crime with Bitcoin is the so called tumbling problem. Sophisticated criminal enterprises are

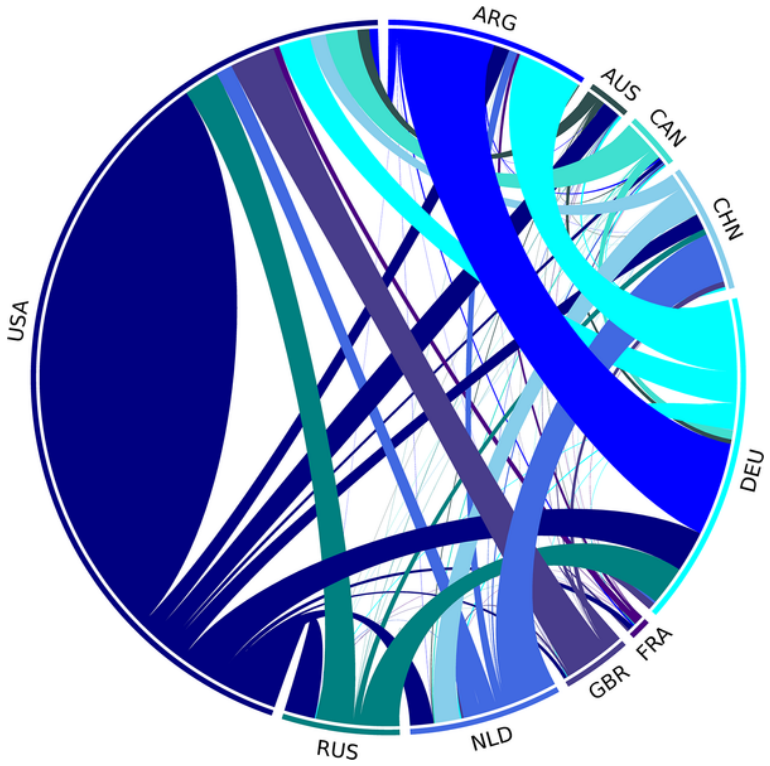

Figure 3.2: Geographic Mapping of Bitcoin Fund Flows [29]

able to add a layer to their existing money laundering operation by taking advantage of privacy oriented services such as tumblers and mixers.

These aforementioned services allow people the ability to send coins to a specified address; subsequently, after a series of transactions over a random period of time, the coins are relayed (after taking a nominal fee for the service) back to the original sender(s) in relative proportion. These services often add a time delay, split the output into many addresses and also use various levels of transaction fees to conduct the mixing operation. And while these services do have legitimate use cases – for example, for people who want to anonymize their coins after their address becomes known, criminals also often leverage this service as well to further obfuscate the origin of their funds.

Prado-Romero et al (2018) propose an algorithm by modelling Bitcoin as a social network and using community anomaly detection to uncover such mixing accounts [30]. The premise of their work is that just like in the fiat world, people tend to transact mostly within their communities and also with same known group of people. Since mixing involves combining coins from different users, and redistributing them, by definition this is anomalous behaviour as most of the transactions are outside of their typical expected transaction graph. Furthermore, there are potentially more anomalous traits as a very small minority of users actually use mixing services. The central thesis of their paper is that people who ultimately possess significantly more inter-community connections compared to the rest of users belonging to its same community are likely to be related to probable mixing sites. They develop an algorithm called InterScore that searches for this community outlier factor and analyzes each element in its community in an unsupervised

fashion. At the end of the analysis, a score of an outlier ranking for each user is outputted.

Although the authors stipulate they cannot guarantee with absolute certainty whether an address is truly affiliated with criminal activity, it is a good starting point for further analysis and investigation. Many services which take cryptocurrencies today as payment already have algorithms in place to detect whether a transaction input originated from a mixing service. Once such a transaction is identified, it is either frozen or returned to sender. For the purposes of our thesis, we refer and assume mixed coins to directly trigger a regulatory KYC/AML process when received by an exchange as this is likely to be considered anomalous activity.

Chapter 4

Implementation

This chapter contains the description for the code implementation part of the thesis - specifically the machine learning models. It includes the tools and frameworks used and provides reasoning as to why each one is used. In the subsequent Evaluation chapter, a discussion of the results from the models and its implications is conducted.

4.1 The Dataset

For the purposes of this study, we use to our knowledge one of the most comprehensive blockchain datasets which are publicly available – the Bitcoin dataset released by Elliptic [31]. Getting complete, real world blockchain transaction data is often cumbersome and impractical, so Elliptic is one of the best known complete datasets that exist to date. Furthermore, the company Elliptic itself is a leader in forensic cryptoanalysis; as a result, not only does this dataset allow us to conduct the research for this paper on industry standard data, but it also allows for direct comparison to work already done by other authors in the field, (as this dataset is often the baseline for many experiments conducted in published papers) such as the Weber paper [1]. More specifically, the dataset consists of:

- 203,769 total transactions
- 234,355 edges (directed flows)
- 46,564 definitively labelled transactions
- 9:1 licit to illicit transaction ratio
- 94 local features (e.g., time step, in/out count activity, transaction fee)
- 72 one hop aggregate features (e.g., maximum, minimum, standard deviation and correlation coefficients of the neighbour transactions)

Figure 4.1 shows the composition of transactions in the dataset (illicit, licit and unknown). Illicit transactions are typically defined to consist of various scams, crypto-exchange heists, ransomware payments and Ponzi schemes, while licit transactions are simply payments made between honest users of the network and for honest goods and services. After mapping the Bitcoin datasets to real entities (both belonging to illicit and licit categories), a graph is made such that nodes represent transactions, while the edges depict the flow of Bitcoin going from one transaction to the next.

Furthermore, each transaction has 166 features, out of which 94 are local features and the remaining 72 are called aggregate features. For instance, local features include things like time step, number of inputs/outputs, transaction fee, output volume, while aggregated features are constructed by using information one-hop backward/forward from the transaction; for example, the standard deviation of each transaction feature, the correlation coefficients of the neighbour transactions for the same information data (number of inputs/outputs, transaction fee, etc) [1]. The Elliptic dataset has been explored in depth and published by many since its release (in particularly on Kaggle) - for this we give credit to the various open source works available for starter code and charts [31]. With that said, we elect to include Figure 4.2 and Figure 4.3 to describe the dataset even further; Figure 4.2 shows the number of transactions at each Time Step, whilst Figure 4.3 highlights the type of transaction (unknown, licit or illicit) at each timestep. Finally, Figure 4.4 shows the head of our dataset after completing the necessary pre-processing for our model.

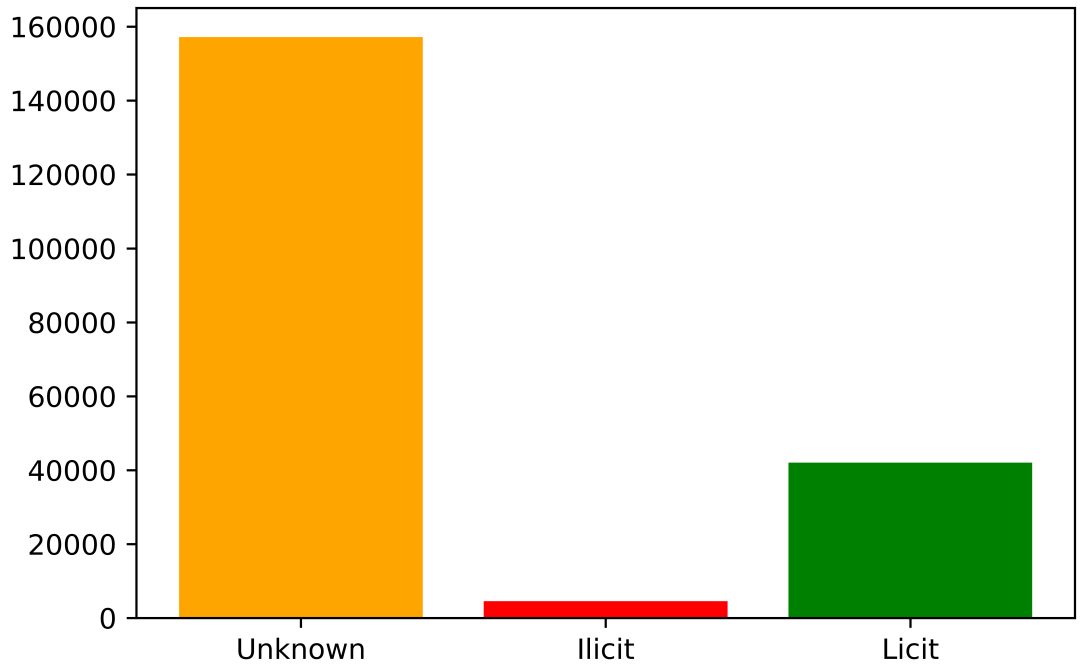

Figure 4.1: Distribution of Classes from the Elliptic Dataset

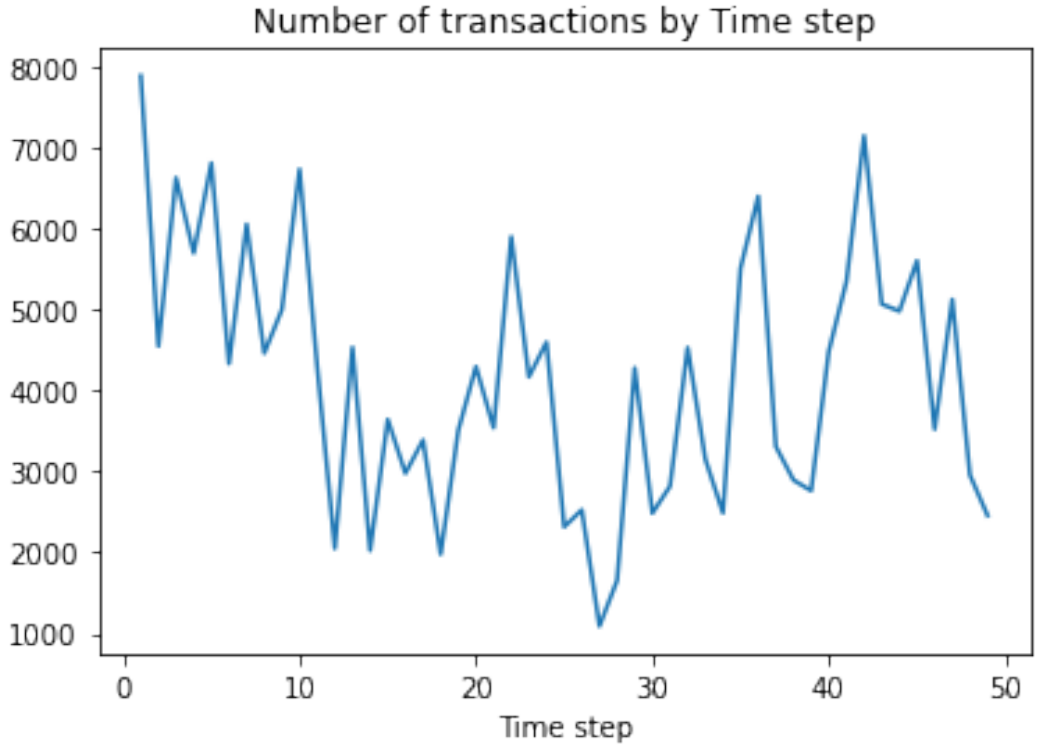

Figure 4.2: Number of transactions by time step

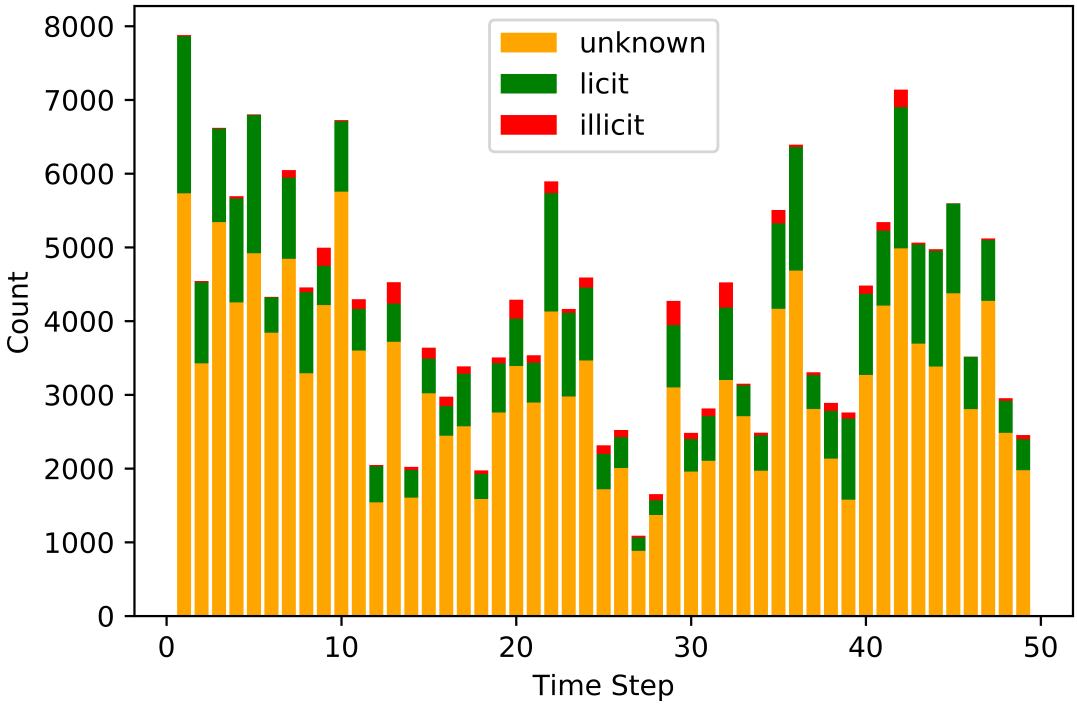

Figure 4.3: Number of transaction types at each time step

	txId	time_step	Local_TX_Feature_1	Local_TX_Feature_2	Local_TX_Feature_3	Local_TX_Feature_4	Local_TX_Feature_5	Local_TX_Feature_6
0	230425980	1	-0.171469	-0.184668	-1.201369	-0.121970	-0.043875	-0.113002
1	5530458	1	-0.171484	-0.184668	-1.201369	-0.121970	-0.043875	-0.113002
2	232022460	1	-0.172107	-0.184668	-1.201369	-0.121970	-0.043875	-0.113002
3	232438397	1	0.163054	1.963790	-0.646376	12.409294	-0.063725	9.782742
4	230460314	1	1.011523	-0.081127	-1.201369	1.153668	0.333276	1.312656
...	...	...	...	...	...	...	...	...
203764	173077460	49	-0.145771	-0.163752	0.463609	-0.121970	-0.043875	-0.113002
203765	158577750	49	-0.165920	-0.123607	1.018602	-0.121970	-0.043875	-0.113002
203766	158375402	49	-0.172014	-0.078182	1.018602	0.028105	-0.043875	0.054722
203767	158654197	49	-0.172842	-0.176622	1.018602	-0.121970	-0.043875	-0.113002
203768	157597225	49	-0.012037	-0.132276	0.463609	-0.121970	-0.043875	-0.113002

203769 rows × 168 columns

Figure 4.4: Processed dataset (203769 rows × 168 columns) [31]

4.2 Frameworks and Tools Used

For the purposes of the experiments of this thesis, the following main technology was used:

- **Python 3** - the programming language used to conduct all of the experiments within this thesis
- **Jupyter Notebooks** - computational notebook which allows the sharing and presentation of both live code and markdown in a user friendly way (thesis experiments can be downloaded via an .ipynb)
- **scikit-learn (sklearn)** - machine learning library that supports data pre-processing and ML algorithm implementation
- **networkx** - a Python library that allows us to visually create and manipulate graph networks
- **numpy** - a Python library for multi-dimensional arrays and matrices that are able to support mathematical transformations
- **pandas** - a Python library which allows for data manipulation and various data structure operations
- **matplotlib** - a Python library (extension of numpy) that allows for the creation of various plots and visualizations
- **seaborn** - a Python library for more visualizations
- **xgboost** - the primary machine learning algorithm used (also using Random Forest, Logistic Regression from scikit-learn)
- **SHAP** - a Python library for which allows us to get an explanation of how are machine learning algorithm makes decisions
- **ELI5** - another Python library to explain predictions made via our machine learning algorithms

4.3 Ensemble Methods and Gradient Boosting

An ensemble model is simple a technique that combines more than one model during its prediction stage. In general, models can arrive at incorrect predictions due to a variety of factors and reasons; however, if you take information from many models, it is possible and likely to get a progressively better model. Thus ensemble methods often solve various issues [32]:

- `Accuracy` - can increase accuracy by using more than just a single estimator
- `Variance` - can prevent high variance in its usable inputs
- `Feature Bias` - can prevent the model from heavily relying on noise and bias during the prediction stage

Gradient Boosting is a way to make weak learners into strong ones. A weak learner is defined as one whose performance is at least slightly better than complete pure random chance. One of the first gradient boosting algorithms created was known as Adaptive Boosting, (or colloquially known as) AdaBoost. There are three main elements in Gradient Boosting [33]:

- A loss function to be optimized
- A weak learner to make predictions
- An additive model to add weak learners to minimize the loss function

More intuitively, in such a gradient boosting framework, a decision tree is first trained in which each observation has the same (equal) weights. Then after the first tree iteration, the pre-existing weights are adjusted higher and lower based on the observations that are (respectively) more difficult / simpler to classify. By doing this stage-wise addition on another tree, the predictions of the first tree have the chance to be improved since the model now suddenly contains more trees. This is because newly created trees have the ability to classify observations which were not classified in the best way by the former trees. Then, once all the trees are grown, the (ensemble) model is the weighted sum of predictions made by the tree models. In essence, performance can be vastly improved by combining many simple models into a single model more optimized model.

4.4 Choosing XGBoost (Extreme Gradient Boosting)

As the Weber paper showed superiority of Random Forest over a multitude of other algorithms that they ran, this is the direction this thesis took [1]. In fact, the authors specifically invited further contributions to improve their baselines. One of the ways that random forest can be improved in certain situations is through Gradient Boosting. In

many studies, it has been shown that Gradient Boosting is favoured to simple Random Forests when it comes to both performance and predictive ability [32]. Thus, we will introduce XGBoost to our study (eXtreme Gradient Boosting) [34]. Extreme Gradient Boosting uses a gradient boosting framework that was mentioned in the previous subheading. However, it is much more optimized when it comes to speed and performance. It uses both a presorted algorithm and histogram based one to compute the very best split. Thousands of models on various subsets of the dataset are trained, and then votes are casted for the most accurate performing models. We elect to choose XGBoost as it has many advantages, with some very specific to our dataset [34]:

- it is great for unbalanced datasets (we have a definite class imbalance in our dataset as there are many more licit transactions vs. illicit transactions)
- XGBoost straight away prunes the tree with a score called "Similarity score" before entering into the actual modeling purposes
- XGBoost always gives more importance to functional space when reducing the cost of a model while Random Forest tries to give more preferences to hyperparameters to optimize the model
- the fastest gradient-boosting library for Python
- parallelization – implemented to train with multiple CPU cores
- prevention of severe overfitting – penalties assigned through its regularization process
- scalability – can process vasts amounts of data
- cross-validation – exists already implemented in the algorithm itself

Usually gradient boosted trees can choose a loss function depending on whether the problem is a classification or regression one. The size and number of trees can be adjusted as well. In XGBoost, rather than minimize the loss function, it minimizes the objective function L_m which in addition to the loss function also has a regularization term $\Omega(b_m)$ that limits the complexity of the model. More formally [35]:

$$L_m = \sum_{i=1}^{N} L((y_i, f_{m-1}(x_i) + b_m)x_i)) + \Omega(b_m)$$

where,

$$\Omega(b_m) = \gamma T + \frac{1}{2}\lambda ||s||^2$$

Notable Tradeoffs: As with any machine learning algorithm, there are definitely some trade-offs when it comes to applying XGBoost [35]:

- more susceptible to overfitting
- XGBoost is much harder to tune vs. Random Forest algorithm

A major advantage when it comes to using XGBoost is that it is resilient when many features exist. XGBoost allows us to give our model a large number of variables without having to commit to any hard decisions about the dataset prior to the learning itself.

Furthermore, XGBoost also allows us to satisfy the condition for the model to be explainable with regards to AML regulations; to do this we run the SHAP (SHapley Additive exPlanations) technique on our model. SHAP is a game theoretic approach to explain the output a Machine Learning model which has been discussed in the previous section. SHAP is based on the game theoretic Shapley values developed in the 1950's by the mathematician Lloyd Shapley [36]. The premise behind his concept was that he could assign a unique distribution of values (excess surplus) to a certain group who was playing a cooperative (coalition) game; for instance, those who contributed more to a positive result should reasonably expect to be compensated more than those who contributed less. This expectation is called consistency and it one of the first Fairness properties. The second fairness property is called Additivity – where the amounts must sum up to exactly to the final game result; you subsequently count the marginal contribution for each player – each running the game with and without this player for the entire subset of players [37][38].

4.4.1 GridSearchCV - Hyper parameter Tuning

Sklearn's `model_selection` package has a tool which assists with hyper parameter tuning of our model. We employ the GridSearchCV tool as a method to ensure that we have the most optimal hyper parameters for our specific dataset and subsequent model. GridSearchCV is a brute force way that tries every possible combination of hyper parameters and tells us in the end which one to use. This is one of the most salient steps in any machine learning problem as the performance of the model can vary drastically based on the values of the hyper parameters.

4.5 Feature Importance

Oftentimes, the generic feature importance plots for machine learning models contradict one another. This is because feature importance is based on the improvement in loss or impurity that has been contributed by all the occasions where the tree splits were based on that feature. While generic feature importance plots sometimes highlight variables which appear to make sense, they also can be misleading due to various definitions of "importance". Relatively recently though, namely in 2017, a very sophisticated approach that revamped feature importance was published (discussed in section 4.5.1 below).

Nevertheless, when running XGBoost, we are able to generate a Feature Importance plot for the model for the top 20 features. In the next Evaluation chapter, we show 3 different types of feature importance - cover, weight, gain. While each of these metrics have their own reasons to be interpreted, we propose to use a different method in order to not only identify global feature importance but also be able to pinpoint individual local decisions.

4.5.1 Shapley Values

In their paper *A Unified Approach to Interpreting Model Predictions* by Lundberg and Lee, they explain how a model's interpretability is as central as its underlying accuracy. They adapt the Shapley values from the 1950 to machine learning scenarios; intuitively, they explain how much a feature, p, contributes to $f(x)$, they count over all possible subsets of features and calculate the model result with and without feature p. This results in a marginal error which is the contribution with or without feature p, then after averaging it over all possible features, you can get the SHAP value for feature p. Summing over the contributions for all features p, you get a value that depicts how much information contained in the features causes the prediction to shift relative to a model prediction ϕ_0. By summing ϕ_0 and all the individual feature contributions $\phi_1(f, x), i = 1, 2, ..., p$, you can get the model prediction [37] [39]:

$$f(x) = \phi_0(f, x) + \sum_{i=1}^{p} \phi_i(f, x).$$

The Shapley value for a certain feature i (out of n total features), given a prediction p is:

$$\phi_i(p) = \sum_{S \subseteq N/i} \frac{|S|!(n - |S| - 1)!}{n!}(p(S \cup i) - p(S))$$

Behind the scenes, the SHAP package samples a certain amount of features instead of going over all the subset of features (you can set how many subsets you want). They posit that high accuracy in large datasets that are relevant today is often achieved by utilizing complex models (such as ensemble ones). They state that although there are an abundance of methods which seek to drive the explainability of models, it is often unclear which one is superior over the other. In fact, we demonstrate this issue in the next section where we show feature importance plots utilizing different bases. To address this debate on which one to use in which situation, Lundberg and Lee propose the SHAP (Shapley Additive exPlanation) framework. The entire purpose of SHAP is to get closer

to explaining individual predictions for an isolated observation rather than predictions on a global scale. By exploring SHAP, we show that the Machine Learning model is interpretable and explanatory – both of which are vital aspects of any AML model that is designed. Some factors which give the rise for such a need are the need to give consumer explanations, anti-discrimination measures, compliance costs and also risk management purposes. SHAP allows us to visualize explanations for a single transaction since it can be both applied for:

1. `Global Interpretability` – shows how much (either through a positive or negative relationship) each predictor affects the target variable
2. `Local Interpretability` – every single observation in our model can have its own SHAP values generated in a map

Local SHAP values can also be used to calculate overall importance. The global SHAP model has been shown to provide more accurate and useful metrics for model behavior. Shapley values result from average over all $N!$ possible ordering (which is NP-hard) and require us to find a way to compute these values extremely efficiently. At the end, each Shapley value is a measure of contributions that each predictor (feature) has on a machine learning model.

4.5.2 ELI5 Values

Similarly to SHAP values, a popular metric to explain Machine Learning Models is ELI5. An advantage is that it uses actual features for permutation rather than the inputs and outputs of the model itself to determine importance. ELI5 gives us two different ways to understand our model which can be directly compared to SHAP values:

1. Analyze model weights to understand the global performance of the model (Global Interpretability)
2. Analyze individual sample prediction to understand the local performance of the model. This can help us drill down as to why a particular prediction was made and which parameters played what role in that prediction (Local Interpretability)

ELI5 is especially superior than the feature importance available by default with XGBoost as it sets clearly defined permutation definitions. On the other hand, for the regular built-in feature importance on XGBoost, we must interpret whether we care about weight, cover or gain for each specific case. This also allows the model to be explained to a stakeholder who does not need the advance technical know-how in machine learning model debugging or understanding.

We can see all the aforementioned plots along with a discussion of them in Chapter 5.

Chapter 5

Evaluation

This chapter provides the results for all the machine learning models created for this thesis. Various metrics for evaluating machine learning models are employed. Furthermore, we compare our results to models, which already exist in publications (who also use the dataset released by Elliptic), to establish a confident baseline.

5.1 Performance Metric Summary

One of the most popular ways to evaluate the performance of a machine learning model is through a classification report or a confusion matrix. We first summarize the results of three models in a table below - Logistic Regression, Random Forest and XGBoost. The aforementioned models all used a 70:30 split - meaning 70% of the data was for training and the remaining 30% was for testing purposes. It is notable that XGBoost was the best performer out of the models we ran. Furthermore, we can compare our results to another published paper on the same dataset and can see that our precision score is the highest and recall still amongst the top performers. This shows that XGBoost is one of the preferred algorithms to use on this dataset.

Machine Learning Algorithms				
Method	Illicit Precision	Illicit Recall	F1	Micro F1
`Logistic Regr`	0.454	0.633	0.529	0.928
`Random Forest`	0.615	0.622	0.618	0.951
`XGBoost`	0.984	0.636	0.773	0.976

Table 5.1: Results of our Machine Learning Models

5.1.1 XGBoost Outperformance

Notably, we can see that tree based ensemble methods do outperform Logistic Regression. Both methods reduce the error rate for False Positives and also False Negatives in a

Method	Illicit Precision	Illicit Recall	Illicit F_1	MicroAVG F_1
Logistic Regr^{AF}	0.404	0.593	0.481	0.931
Logistic Regr^{AF+NE}	0.537	0.528	0.533	0.945
Logistic Regr^{LF}	0.348	0.668	0.457	0.920
Logistic Regr^{LF+NE}	0.518	0.571	0.543	0.945
RandomForest^{AF}	0.956	0.670	0.788	0.977
RandomForest^{AF+NE}	0.971	0.675	0.796	0.978
RandomForest^{LF}	0.803	0.611	0.694	0.966
RandomForest^{LF+NE}	0.878	0.668	0.759	0.973
MLP^{AF}	0.694	0.617	0.653	0.962
MLP^{AF+NE}	0.780	0.617	0.689	0.967
MLP^{LF}	0.637	0.662	0.649	0.958
MLP^{LF+NE}	0.6819	0.5782	0.6258	0.986

Figure 5.1: Results of Weber (2019) for Comparison [1]

noticeable way. However, we find that the XGBoost model is the most quite useful when it comes to anomaly detection. We tuned our XGBoost model even further by running GridSearchCV - this brute force way proved to be an efficient way to find the optimal hyper parameters for our underlying dataset (see `models.ipynb file`).

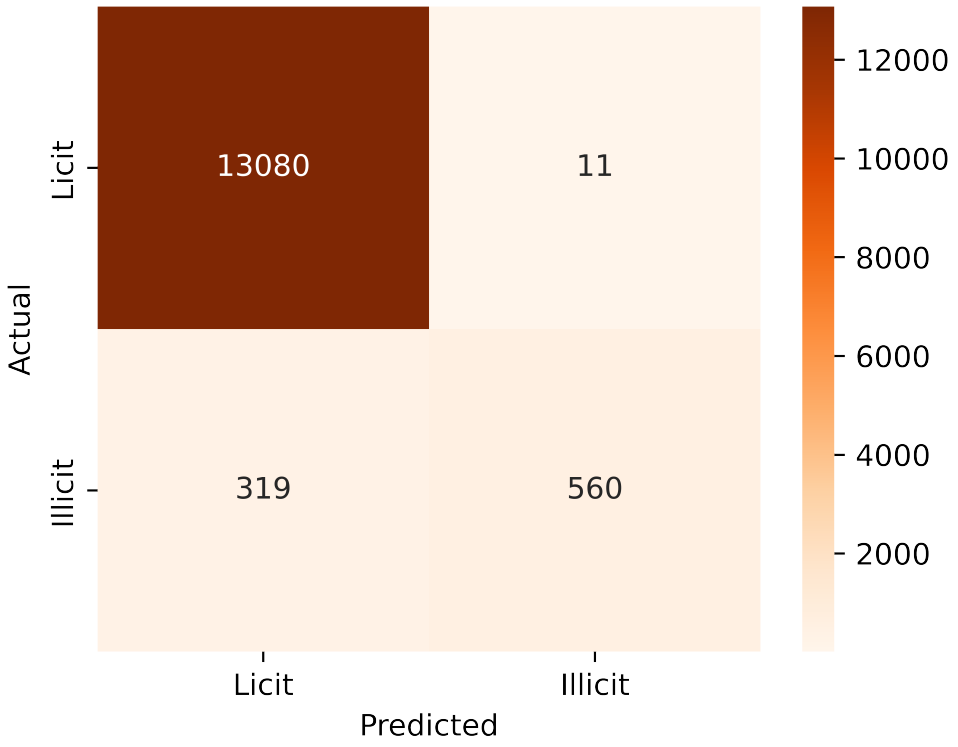

Figure 5.2: Confusion Matrix (for XGBoost)

5.1.2 Feature Importance

We examine XGBoost by first plotting the various feature importances - notably weight, cover, and gain [40]:

- The weight metric (also called frequency) is the percentage representing the relative number of times a particular feature occurs in the trees of the model

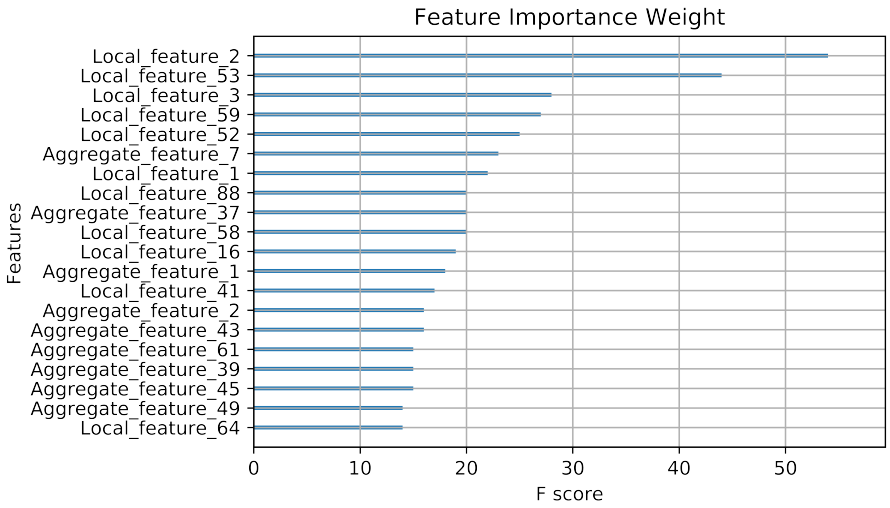

Figure 5.3: Feature Importance (type = weight) after running XGBoost

- The Coverage metric means the relative number of observations related to this feature. The cover is normally expressed as a percentage for all the features' cover metrics.

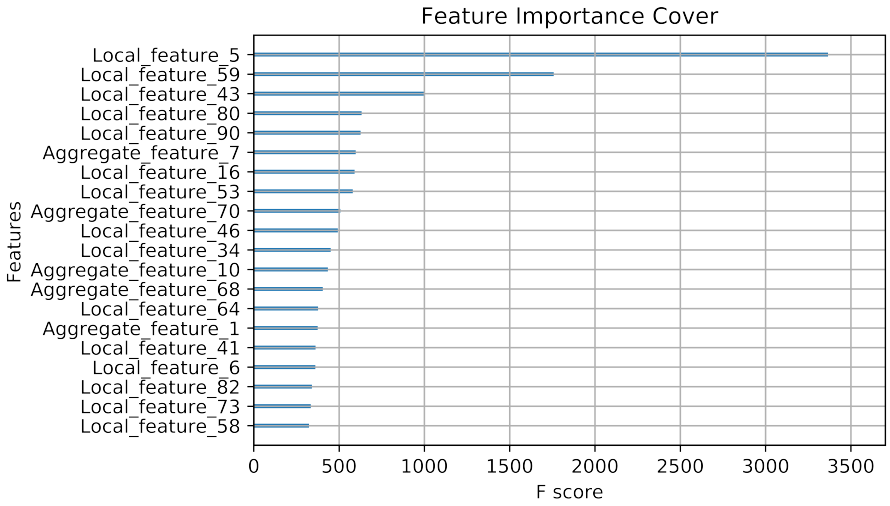

Figure 5.4: Feature Importance (type = cover) after running XGBoost

- The Gain implies the relative contribution of the corresponding feature to the model calculated by taking each feature's contribution for each tree in the model. A higher value of this metric when compared to another feature implies it is more important for generating a prediction. On the other hand, a lower value for this metric depicts it is less important for generating a prediction.

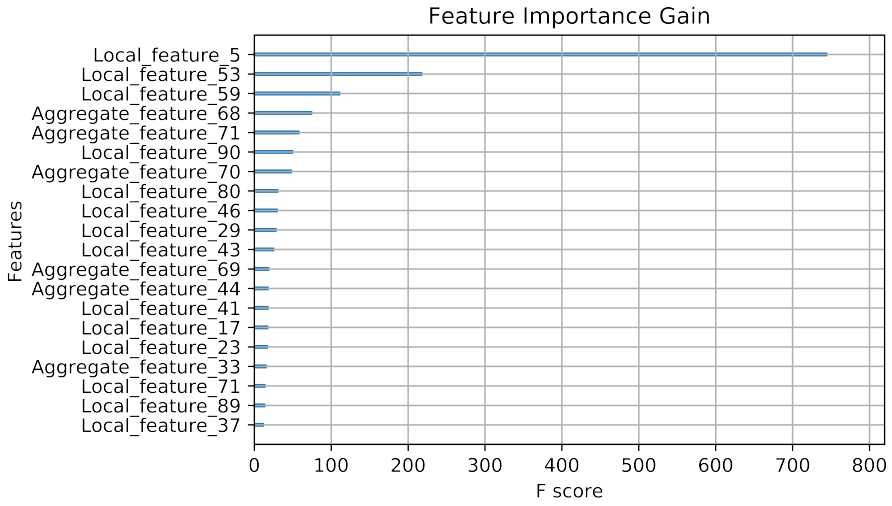

Figure 5.5: Feature Importance (type = gain) after running XGBoost

Global and Local Interpretability

The below Figure 5.6 presents the most important variables (in descending order) in our model to help understand what is driving its decisions. In other words, the plot is sorted in order to use SHAP values which show the distribution and subsequent impact that each feature carries on the actual model. This is an improvement to simply using the generic feature importance(s) which varied significantly depending on the importance metric used.

The SHAP Global Summary Plot figure gives us a combination of feature importance and also feature effects. Here, we have the most important 20 features and their subsequent effects. Moreover, we can see that the most important features are: `Local_TX_Feature_53`, `Local_TX_Feature_5` and `Local_TX_Feature_59` and the most important aggregate feature is `Aggregate_TX_Feature_69`. The position on the x-axis is determined by Shapley value. The colour of the feature ranges from high to low and represents the value of the feature and impact of class prediction. For instance, we can see that low values of `Local_TX_Feature_53` and `Local_TX_Feature_5` both have a very high impact on the model output. Interestingly, we can see in turn that `Local_TX_Feature_46` and `Local_TX_Feature_40` both have a very high impact when the values are medium to high - while low values tend to impact the model in the opposite direction.

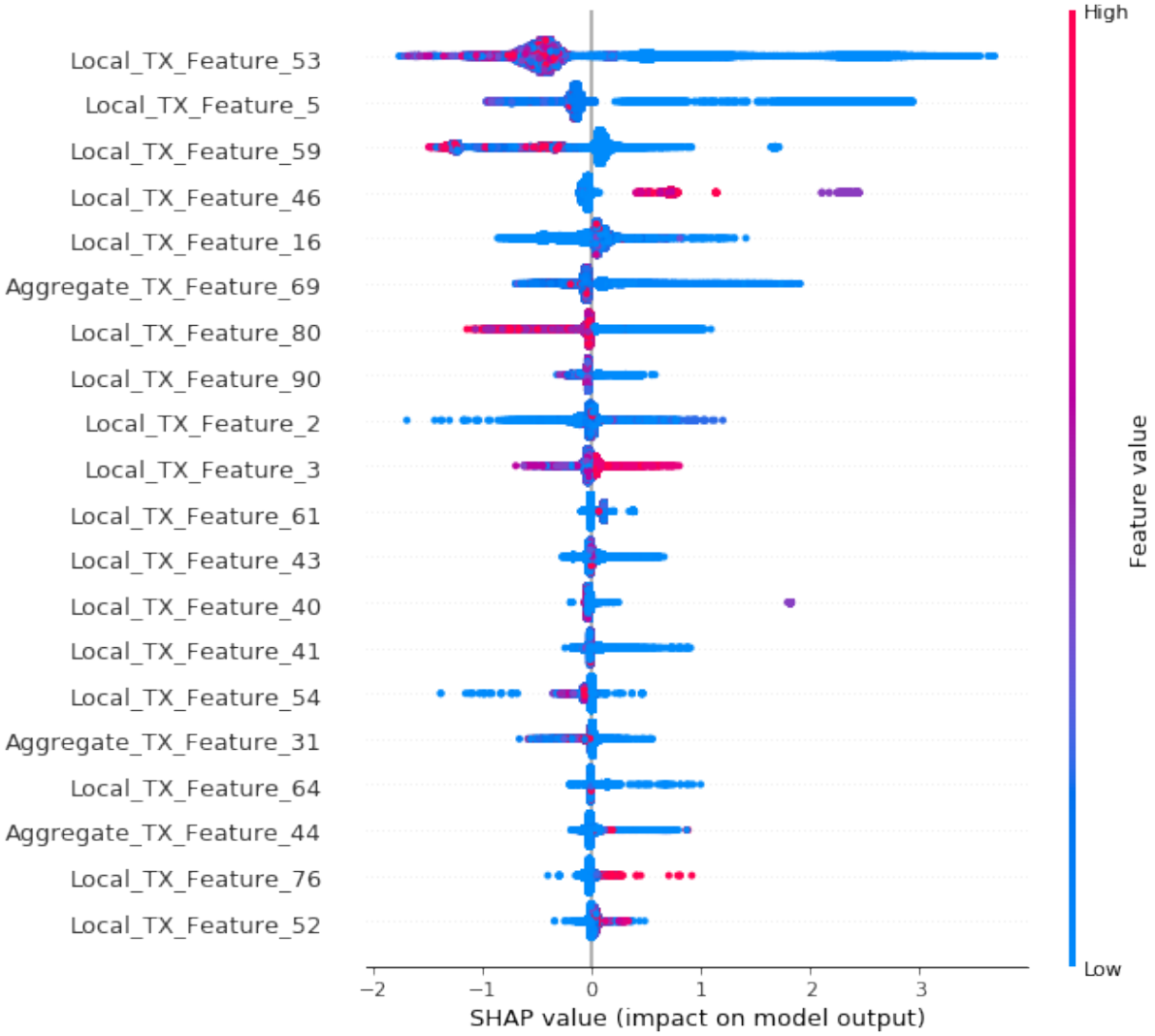

Figure 5.6: SHAP Global Summary Plot

5.2 Explaining Features

This section explains the results and rankings of variables for individual observations using the motivated SHAP values and ELI5 permutation technique.

5.2.1 SHAP Results

Moreover, we are interested in explaining individual observations that our model predicted. For this, we can use SHAP explanation force plots which were motivated in the previous section. Each feature in the below plot is one that either can increase or decrease the class prediction from the baseline. At the end, we can see why the prediction is closer to class 0 or class 1 (licit and illicit, respectively). As expected, we can see some overlap between the global SHAP plot and the local one. Here, we see the two most dominant features which impact the model the most - they are `Local_TX_Feature_53` and `Local_TX_Feature_31`.

The local plot starts with a base value of -4.412. As XGBoost is built on the log-odds scale, the negative values are valid and contributions are still readily visible. The model's base value is then impacted higher or lower based on the features for that particular observation. In this case, we end with $f(x) = -9.93$ which is markedly lower than our base value - this indicates that the model predicted class 0 or in other words, that the transaction was from the licit class. As the model predicted a lower value, we can now see the contribution of each individual feature to the model. For example, we can see that `Local_TX_Feature_53` and `Aggregate_TX_Feature_31` contributed the most to showing that this was indeed a legitimate and licit transaction. On the other hand, `Local_TX_Feature_5` and `Local_TX_Feature_90` were the main drivers for the model thinking that it could potentially belong to the illicit category or class 1.

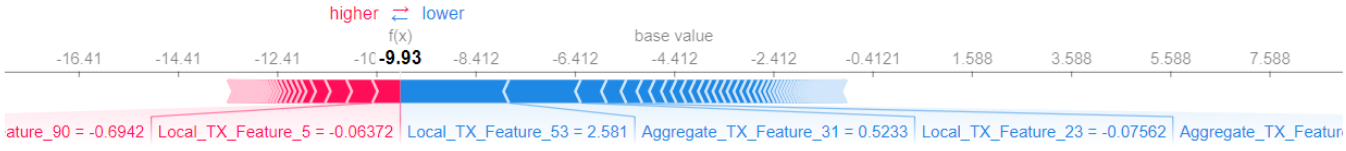

Figure 5.7: SHAP Local Explanations

5.2.2 ELI5 Results

As mentioned in the previous section, another way to see feature importance with permutations is via ELI5. In Figure 5.8, we share the results of the most important features and least important features (green and red respectively) for an observation. While there are some differences in comparison to SHAP, they are not significant and further reinforce that certain features in our dataset have higher (and lower) predictive powers relative to others.

Contribution[?]	Feature	Value
+4.079	<BIAS>	1.000
+1.753	Local_TX_Feature_59	0.151
+1.621	Local_TX_Feature_53	3.288
+0.832	Local_TX_Feature_89	2.358
+0.740	Local_TX_Feature_61	-0.032
+0.581	Local_TX_Feature_5	0.016
-0.276	Aggregate_TX_Feature_37	-0.156
-0.289	Local_TX_Feature_79	-0.165
-0.305	Aggregate_TX_Feature_7	-1.494
-0.471	Aggregate_TX_Feature_8	3.797
-0.489	Local_TX_Feature_34	-0.024
-0.665	Local_TX_Feature_72	-0.015

Figure 5.8: ELI5 Local Explanations

Likewise, we can also view the weights from a global perspective in Figure 5.9. This aggregation shows the weights that each feature carries. The higher the weight, the more critical the feature is in the model's scoring. Once again, this is strikingly similar to SHAP, which further reinforces our confidence when it comes to model interpretability as we see overlaps in both Local and Aggregate Feature rankings (out of 166 total features). ELI5 allows us to take what is an opaque tree model and achieve some interpretable results.

Weight	Feature
0.4815	Local_TX_Feature_5
0.1520	Local_TX_Feature_40
0.0426	Local_TX_Feature_53
0.0425	Local_TX_Feature_46
0.0216	Aggregate_TX_Feature_69
0.0148	Local_TX_Feature_90

Figure 5.9: ELI5 Global Explanations

5.3 Summary

We can that see that our top performing model (XGBoost with GridSearchCV) shows very encouraging results. We identify over 98% of all illicit transactions correctly. This means that any compliance efforts using such a model would not be drastically overburdened by the large amount of false positives or fraud flags for illicit transactions. This is ever so important as investigations on the blockchain can both be time consuming and laborious. Furthermore, by being able to capture 64% of all cases of known fraud, it shows that our model can be quite valuable in the fight against money laundering on the blockchain.

To date, there does not exist much work with regards to explaining and interpreting anomalies on the blockchain. Shapley values and ELI5 offer an opportunity for regulators

and blockchain investigators to be quite confident about the results of a model by showing which variables contribute to a certain prediction the most. Such explanations prevent the model from falling under regulatory scrutiny of being too opaque. By being able to see and assess the impact that each individual feature (or variable) has on the predicted outcome, it allows investigators to dive much more deeper into a potential fraudulent scheme. Moreover, by continuing research in this direction, it can be possible to prevent cryptocurrencies from being overly regulated - which would keep payments accessible for all of society.

Limitations

The dataset released by Elliptic Co is one of the largest and most comprehensive transaction labelled datasets released to the public. However, it is important to note that having access to more of such labelled data for the purposes of further model development is not always feasible as this requires a high manual effort by some individual or company. Thus, anomaly detection is generally an unsupervised machine learning task. Moreover, the aforementioned party must also be willing to release its data to the public.

Furthermore, while Elliptic has released the dataset to the public in hopes of encouraging research interest on novel anomaly detection methods, most of their data remains in an anonymous form. For instance, we saw in the previous sections which features were the most important for global and local interpretability (e.g., `Local_TX_Feature_53`), but we actually do not know what the underlying feature truly is as it is anonymized. Elliptic has given clues of what local features and aggregate features are (e.g., the number of inputs and outputs, the transferred amounts, payments to miners, etc), but we cannot make a certain 1-to-1 mapping of it from their dataset description. This is a limitation in the sense for compliance officers and those conducting certain investigations, but for model development, gauging performance remains unimpeded.

With that said, (a questionable practice, but important to acknowledge) there have already been efforts to deanonymize the Elliptic dataset. At the time of writing, it is possible to deanonymize the transactions by building a directed graph and calculating the incoming and outgoing degree of each transaction ID [41]. Then, using the elliptic time value, it is feasible to link the transactions in which the specified numbers occur only once in the Bitcoin blockchain's history. In this case, it means taking the random ID's Elliptic has assigned and finding the actual transaction hashes from the transaction list provided. It is then possible to overlay the other attributes (e.g., the number of inputs and outputs, the transferred amounts, payments to miners) to see most of the feature values. However, in the spirit of Elliptic's release of the dataset to the public sphere for anomaly detection research, the attributes provided are enough to gauge both model performance and interpretability. The actual transaction hashes and attributes would be most useful to those who are conducting an in-depth blockchain investigation.

Lastly, the nature of transactions on the blockchain are always changing. Places where illicit transaction tend to cluster around - darknet marketplaces, illegal gambling sites constantly open and shut-down. In fact, this concept was demonstrated by Mark Weber where

following a a large dark web marketplace shutdown, many anomaly detection algorithms were not working as expected and suffered performance wise [1]. What makes something anomolous today, may not be an anomaly (or may not even exist) on the blockchain tomorrow; as a result, it is always imperative to have new observations and transaction history - particularly if you are training a supervised machine learning algorithm.

Chapter 6

Summary and Conclusions

Cryptocurrencies today make up an irrefutably significant portion of financial transactions. While the vast majority of transactions are made for honest and legitimate services, a portion of them are comprised of illicit transactions. Rather than settle for strict AML laws or simply have regulators put curbs on cryptocurrencies, anomaly detection presents us with a way to both to identify and trace such illicit transactions on the blockchain. Furthermore, having sophisticated machine learning tools to detect anomalies not only allows us to identify illegal flows, but also serves as a major deterrent from criminals using the network for their own illicit gain in the first place.

The cross border component of many blockchain transactions means we need a novel and clever way to combat money laundering and organized crime within cryptocurrencies. The traditional way – where a centralized 3rd party was in charge of verifying transactions has been shown that it could be replaced by digital signatures and cryptographic proof. Although there appears to be less control in such a system, there is at the same time, a higher level of immutable data that can be viewed and verified by anyone at anytime. We showed that by leveraging the power of machine learning, it is possible to uncover illicit transactions at a strikingly accurate rate. By doing so, we can catch criminals and also drastically reduce the negative effects that strict AML/KYC laws have on society - especially those who are most financially vulnerable.

Furthermore, not only did we show that it is feasible to identify anomalous activity on the blockchain with a reasonably high accuracy, but the thesis also depicted that it is also possible to explain what drives the decision of the model. Explainable machine learning - in particular, the local interpretability factors are something that is crucial in the compliance world since it is not possible to simply rely on a black box machine learning model. Rather, we can take an opaque model and readily understand the decisions it makes. The progress made in recent years when it comes to improving the ability to explain the decisions of various machine learning models should not be ignored for anomaly detection on the blockchain.

The existence of illicit transactions on the blockchain impacts all realms of society - ranging from stolen funds from ordinary people, to ransomware attacks on corporations,

and to even actions that can destabilize entire governments. The oftentimes high paced nature of cryptocurrency and blockchain development make it somewhat difficult for regulators and law enforcement to adequately keep up. However, it is clear from the results of the machine learning algorithms and also the work done by other researchers, that even the most sophisticated ways to hide on the blockchain can ultimately be identified and tracked. While some level of regulation for the blockchain may be necessary, when working with vasts amounts of immutable data, human ingenuity can be an effective way to deal with illicit transactions on the blockchain.

Bibliography

[1] M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, "Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics," *arXiv preprint arXiv:1908.02591*, 2019.

[2] T. Pham and S. Lee, "Anomaly detection in bitcoin network using unsupervised learning methods," *arXiv preprint arXiv:1611.03941*, 2016.

[3] S. Nakamoto, "Re: Bitcoin p2p e-cash paper," *The Cryptography Mailing List*, 2008.

[4] B. Maurer, T. C. Nelms, and L. Swartz, ""when perhaps the real problem is money itself!": the practical materiality of bitcoin," *Social semiotics*, vol. 23, no. 2, pp. 261–277, 2013.

[5] "Deep dissection of bitcoin and ethernet, and comparison of utxo and acount models," *Develop Paper*, Jun 2019. [Online]. Available: https://developpaper.com/deep-dissection-of-bitcoin-and-ethernet-and-comparison-of-utxo-and-acount-models/

[6] B. Haslhofer, R. Karl, and E. Filtz, "O bitcoin where art thou? insight into large-scale transaction graphs." in *SEMANTiCS (Posters, Demos, SuCCESS)*, 2016.

[7] Y. Huang, *Using Crypto-currencies to Measure Financial Activities and Uncover Potential Identities of Actors Involved*. University of California, San Diego, 2017.

[8] S. Lee, C. Yoon, H. Kang, Y. Kim, Y. Kim, D. Han, S. Son, and S. Shin, "Cybercriminal minds: an investigative study of cryptocurrency abuses in the dark web," in *Network & Distributed System Security Symposium*. Internet Society, 2019, pp. 1–15.

[9] D. D. F. Maesa, A. Marino, and L. Ricci, "Uncovering the bitcoin blockchain: an analysis of the full users graph," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2016, pp. 537–546.

[10] M. Levi, "Money laundering and its regulation," *The Annals of the American Academy of Political and Social Science*, vol. 582, no. 1, pp. 181–194, 2002.

[11] U. Nations, "Overview," *United Nations : Office on Drugs and Crime*, 2018. [Online]. Available: https://www.unodc.org/unodc/en/money-laundering/overview.html

[12] A. Mikhaylov and R. Frank, "Cards, money and two hacking forums: An analysis of online money laundering schemes," in *2016 European intelligence and security informatics conference (EISIC)*. IEEE, 2016, pp. 80–83.

[13] E. E. Esoimeme, "Identifying and reducing the money laundering risks posed by individuals who have been unknowingly recruited as money mules," *Journal of Money Laundering Control*, 2020.

[14] S. Ovaska-Few, "Keeping an eye out for money mules," *Journal of Accountancy*, vol. 228, no. 3, pp. 44–47, 2019.

[15] G. Weimann, "Going dark: Terrorism on the dark web," *Studies in Conflict & Terrorism*, vol. 39, no. 3, pp. 195–206, 2016.

[16] M. Campbell-Verduyn and M. Goguen, "The mutual constitution of technology and global governance: Bitcoin, blockchains, and the international anti-money-laundering regime," in *Bitcoin and Beyond*. Routledge, 2017, pp. 69–87.

[17] W. B. Group, *Global financial development report 2014: Financial inclusion*. World Bank Publications, 2013, vol. 2.

[18] *Anti-Money Laundering in Bitcoin: Experiments with Graph Convolutional Networks*. YouTube, Aug 2019. [Online]. Available: https://www.youtube.com/watch?v=HtJeXPr_PBY

[19] O. Analytica, "Poly network attack underlines growing defi risks," *Emerald Expert Briefings*, no. oxan-es.

[20] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[21] C. Molnar, *Interpretable machine learning*. Lulu. com, 2020.

[22] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (xai): Toward medical xai," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[23] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st international conference on neural information processing systems*, 2017, pp. 4768–4777.

[24] A. Vij and P. Nanjundan, "Comparing strategies for post-hoc explanations in machine learning models," in *Mobile Computing and Sustainable Informatics*. Springer, 2022, pp. 585–592.

[25] S. Sayadi, S. B. Rejeb, and Z. Choukair, "Anomaly detection model over blockchain electronic transactions," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2019, pp. 895–900.

[26] J. Lorenz, M. I. Silva, D. Aparício, J. T. Ascensão, and P. Bizarro, "Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity," *arXiv preprint arXiv:2005.14635*, 2020.

[27] P. Monamo, V. Marivate, and B. Twala, "Unsupervised learning for robust bitcoin fraud detection," in *2016 Information Security for South Africa (ISSA)*. IEEE, 2016, pp. 129–134.

[28] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating user privacy in bitcoin," in *International conference on financial cryptography and data security*. Springer, 2013, pp. 34–51.

[29] P. L. Juhász, J. Stéger, D. Kondor, and G. Vattay, "A bayesian approach to identify bitcoin users," *PloS one*, vol. 13, no. 12, p. e0207000, 2018.

[30] M. A. Prado-Romero, C. Doerr, and A. Gago-Alonso, "Discovering bitcoin mixing using anomaly detection," in *Iberoamerican Congress on Pattern Recognition*. Springer, 2017, pp. 534–541.

[31] "Kaggle elliptic dataset," https://www.kaggle.com/ellipticco/elliptic-data-set.

[32] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[33] W. Zhang, C. Wu, H. Zhong, Y. Li, and L. Wang, "Prediction of undrained shear strength using extreme gradient boosting and random forest based on bayesian optimization," *Geoscience Frontiers*, vol. 12, no. 1, pp. 469–477, 2021.

[34] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

[35] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[36] L. S. Shapley, A. E. Roth *et al.*, *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

[37] M. Sundararajan and A. Najmi, "The many shapley values for model explanation," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9269–9278.

[38] E. Kalai and D. Samet, "On weighted shapley values," *International journal of game theory*, vol. 16, no. 3, pp. 205–222, 1987.

[39] K. Aas, M. Jullum, and A. Løland, "Explaining individual predictions when features are dependent: More accurate approximations to shapley values," *arXiv preprint arXiv:1903.10464*, 2019.

[40] A. Abu-Rmileh, "Be careful when interpreting your features importance in xgboost!" *Medium*, Sep 2021. [Online]. Available: https://towardsdatascience.com/be-careful-when-interpreting-your-features-importance-in-xgboost-6e16132588e7

[41] BenZz, "Deanonymizing the dataset of elliptic," Dec 2019. [Online]. Available: https://habr.com/ru/post/479178/

Abbreviations

AI	Artificial Intelligence
AML	Anti Money Laundering
ELI5	Explain Like I'm 5
GCN	Graph Convolutional Network
IP	Internet Protocol
KYC	Know Your Customer
ML	Machine Learning
SHAP	SHapley Additive exPlanations
SVM	Supervised Vector Machine
TX	Transaction
UI	User Interface
U-SVM	Unsupervised Vector Machine
UTXO	Unspent Transaction Output
XAI	Explainable Artificial Intelligence

Glossary

Anti Money Laundering - refers to the laws, regulations and procedures intended to prevent criminals from disguising illegally obtained funds through licit goods and transactions

Artificial Intelligence - a technology which enables a machine to simulate human behaviour.

Blockchain - a public distributed and decentralized append only ledger linked in a peer-to-peer fashion

Confusion Matrix - an error matrix which allows for easy visualization of a machine learning model's performance metrics

EDA - Exploratory Data Analysis

ELI5 - known colloquially as 'Explain Like I'm 5'

Graph Convolutional Network - it is a very powerful neural network architecture that is specifically designed for machine learning on graphs

Know Your Customer - a set of laws in which financial institutions and money provider services must adhere to - this usually includes asking for ID, proof of income, address. This law also is part of anti money laundering procedures

Machine Learning - a subset of artificial intelligence. Here, machines are able to learn to use data in order to find patterns and learn how humans would learn

Pseudonymous - using a fake or misleading name or identifier to prevent a real identity from being exposed

Unspent Transaction Output - a blockchain transaction that has not been spent, i.e. can be used as an input in a new transaction

XGBoost - eXtreme Gradient Boosting is a gradient boosting framework which aids in machine learning. It is based on decision tree ensemble learning methods.

List of Figures

List of Tables

Appendix A

Installation Guidelines

Any version of `Python 3` will suffice. You will also need to have `Jupyter Notebooks` installed to make following the code easier with the markdown and comments. To do this, I recommend installing Anaconda - the Python distribution which has already has many data science features and packages preinstalled for ease of use.

Anaconda download: `https://www.anaconda.com/products/individual`.

Next you will also need to packages to run the model. The list of packages can be found below, with the detailed descriptions of each found in the Implementation Chapter (Section 5.2). You can either do `conda install [package_name]` or `pip install [package_name]`. Note - some packages are already preinstalled if you chose to install via the Anaconda distribution route.

- `scikit-learn (sklearn)`
- `networkx`
- `numpy`
- `pandas`
- `matplotlib`
- `seaborn`
- `xgboost`
- `SHAP`
- `ELI5`

You may then use your Shell to start Jupyter Notebooks by typing `jupyter notebook` as the command in your shell. The notebook service will start locally to which you may

upload the `models.ipynb` file included in the project folder. You may then execute line-by-line the code (or choose simply choose the model you wish to run by the subheadings). Make sure you are reading the dataset files prior to running the model however - you can do this by ensuring that they are in the same directory (or point them to where you wish by editing the path accordingly in the code). It is also possible to have a view-only file by using `models.html`.

Appendix B

Contents of the CD

The CD contains the following items:

- `/main/models.ipynb` - a file for the source code (main file is models.ipynb)
- `elliptic_dataset.zip` - the Elliptic dataset files (three .csv files in total)
- `/main/models.html` - a file for the source code in html for easier viewing
- `thesis.pdf` - the final thesis in PDF format (thesis.pdf)
- `/main/tex.zip/` - a folder for the Latex source code for the thesis
- `midterm.pptx` - the midterm presentation in PowerPoint format
- `final.pptx` - the final presentation in PowerPoint format