University of Brasília

Exact Sciences Institute
Computer Science Department

Exploring the Energy Flow Classifier to Identify Fraudulent Cryptocurrency Transactions

Kevin S. Araujo

Dissertation submitted in partial fulfillment of the requirements to
Professional Master's Degree in Applied Computing

Adviser
Prof. Dr. Rodrigo Bonifacio de Almeida

Co-advisor
Prof. Dr. Fabiano Cavalcanti Fernandes

Brasília
2025

University of Brasília

Exact Sciences Institute
Computer Science Department

Exploring the Energy Flow Classifier to Identify Fraudulent Cryptocurrency Transactions

Kevin S. Araujo

Dissertation submitted in partial fulfillment of the requirements to
Professional Master's Degree in Applied Computing

Prof. Dr. Rodrigo Bonifacio de Almeida (Adviser)
CIC/UnB

Prof. Dr. Marcelo Marotta
UnB

Prof. Dr. Edilson Ferneda
UCB

Prof. Dr. Edna Dias Canedo
Coordinator of Graduate Program in Applied Computing

Brasília, Nov 12, 2025

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

de Santana Araujo, Kevin
dA663ee Exploring the Energy Flow Classifier to Identify Fraudulent Cryptocurrency Transactions / Kevin de Santana Araujo; orientador Rodrigo Bonifácio de Almeida; co-orientador Fabiano Andrade Cavalcanti. Brasília, 2025.
107 p.

Dissertação(Mestrado Profissional em Computação Aplicada) Universidade de Brasília, 2025.

1. criptomoedas. 2. detecção de anomalias. 3. aprendizagem de máquina. 4. Energy Flow Classifier. I. Bonifácio de Almeida, Rodrigo, orient. II. Andrade Cavalcanti, Fabiano, co-orient. III. Título.

Dedicated to

To every teacher, educator, and knowledge-sharer

"Where there is teaching, there is hope."

Acknowledgements

I would like to express my deepest gratitude to all those who have been part of this challenging journey, who stood by my side through the most difficult moments as well as the most rewarding ones. To all those named and unnamed who have supported my dream of teaching and sharing knowledge with students in similar circumstances to my own—those who, for whatever reason, lost their way in the learning process at some point and were able to recover with proper guidance. No student should be left behind; where there is teaching, there is hope. To my wife, who walked alongside me not only through this master's program but also through the journey and evolution of our relationship. For her unwavering support during all the demanding moments dedicated to this degree, from the initial decision to pursue graduate studies to her steadfast companionship throughout, and for being part of this dream—thank you. To UnB and its entire faculty, administrative staff, and leadership for maintaining and fostering knowledge at this institution, which stands as a treasure of our beloved and planned city of Brasília. To my advisor, Rodrigo Bonifácio, for the entire journey thus far, for his Herculean patience, and for the countless moments of knowledge and wisdom shared—my most sincere and profound gratitude for being part of this dream. I am certain that his technical and scientific rigor will make me a better citizen and a more promising future professor. To my co-advisor, Fabiano Cavalcanti, for accepting to be part of this journey and for guiding me through moments of doubt and uncertainty about results, steering us toward a solution that made this dissertation possible. To you, my most sincere and profound thanks. To Mercado Bitcoin, a private institution that allowed me to begin and continue this research even when I needed time away during business hours. I am grateful for their recognition of the value of professional and personal development. To the forever admirable Rafael Reimberg, also from Mercado Bitcoin, for encouraging, guiding, and enabling this master's degree. Even after leaving the institution, he demonstrated dedication and availability to make this research possible. I am immensely grateful for all the knowledge about the crypto market, architectural insights, lessons on becoming a Pythonista, and for the eternal phrase: "Design patterns are cool, let's use them all." Finally, to family and friends for asking questions like "How's the master's program going?"—these simple inquiries helped

and encouraged me to continue this research and continue to drive me forward in pursuit of the master's degree.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Explorando o Energy Flow Classifier na Identificação de Transações de Criptomoedas Fraudulentas

Kevin S. Araujo

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Orientador
Prof. Dr. Rodrigo Bonifacio de Almeida

Coorientador
Prof. Dr. Fabiano Cavalcanti Fernandes

Brasília
2025

Resumo

A lavagem de dinheiro representa um problema global de grande impacto, com criminosos movimentando bilhões de dólares anualmente provenientes de atividades ilícitas. Nos últimos anos, as criptomoedas emergiram como um canal significativo para essas atividades, principalmente devido ao pseudonimato que oferecem. Em 2023, endereços ilícitos receberam 24,2 bilhões de dólares em criptomoedas originadas de fraudes, fundos roubados e outras atividades criminosas. O aprendizado de máquina apresenta-se como uma ferramenta poderosa para identificar padrões complexos de fluxos financeiros ilícitos, mas enfrenta um obstáculo crítico: a escassez de dados rotulados. Algoritmos de aprendizado supervisionado frequentemente são inviáveis porque conjuntos de dados públicos com transações ilícitas verificadas são raros. Esta escassez decorre da complexidade evolutiva dos esquemas de lavagem de dinheiro e do fato de que a aquisição de rótulos é um processo custoso e lento.

Esta pesquisa avalia a eficácia do Energy Flow Classifier (EFC) para identificar transações ilícitas de Bitcoin no conjunto de dados Elliptic, particularmente sob condições de escassez de rótulos. O EFC é um algoritmo baseado em física estatística, originalmente desenvolvido para detecção de intrusões em redes, que opera sob a premissa de que padrões de dados normais correspondem a estados de baixa energia, enquanto desvios significativos constituem estados de alta energia. O núcleo do EFC é o Hamiltoniano que quantifica a tipicidade estatística de uma transação através da equação $H(a_{k1}, ..., a_{kN}) = -\sum_{i<j} e_{ij}(a_{ki}, a_{kj}) - \sum_i h_i(a_{ki})$, onde h_i representa o campo local e e_{ij} representa o acoplamento entre pares de características.

O conjunto de dados Elliptic contém 203.769 transações de Bitcoin com 234.355 arestas direcionadas, cada transação descrita por 166 características anonimizadas. Do total de transações, apenas 46.564 (23%) estão rotuladas, sendo 42.019 (90,2%) lícitas e 4.545 (9,8%) ilícitas, refletindo a escassez característica de dados rotulados em contextos reais. O estudo foi conduzido através de três experimentos principais. O Experimento 1 avaliou o impacto de técnicas de balanceamento de dados, incluindo undersampling, oversampling aleatório e SMOTE. O Experimento 2 investigou a seleção de características utilizando SelectKBest com valores de $k \in \{10, 20, 30, 40, 50, 60\}$. O Experimento 3 examinou o

impacto combinado das duas técnicas, aplicando seleção de características seguida de SMOTE. Para cada configuração, avaliou-se o desempenho usando F1-Score Macro como métrica primária, fornecendo uma medida balanceada de desempenho crucial dado o desequilíbrio de classes.

Os resultados demonstraram claramente a sensibilidade do EFC ao desequilíbrio de classes. O conjunto de dados desbalanceado baseline produziu F1-Macro de 0,488, confirmando a dificuldade em detectar a classe minoritária ilícita sem intervenção. A aplicação de SMOTE em conjunto de teste balanceado resultou em F1-Macro de 0,908, representando um cenário idealizado. Quando avaliado em dados de teste desbalanceados, o Random Undersampling alcançou F1-Macro de 0,652 e Random Oversampling atingiu 0,533. A seleção de características revelou que o EFC pode alcançar melhor desempenho com um conjunto reduzido: o maior F1-Macro de 0,686-0,689 foi obtido com apenas $k = 10$ características. A estratégia combinada de seleção de características ($k = 30$) seguida de balanceamento SMOTE produziu F1-Macro máximo de 0,808, representando melhoria substancial comparada às técnicas isoladas. A análise fatorial completa revelou forte interação positiva (+0,221) entre SMOTE e SelectKBest, indicando que estas técnicas são complementares.

Os resultados posicionam o EFC como alternativa viável entre métodos não supervisionados tradicionais e métodos supervisionados completos. Métodos não supervisionados como Isolation Forest, Local Outlier Factor e One-Class SVM alcançaram F1-scores de 0,00 a 0,19 no conjunto Elliptic, demonstrando eficácia limitada. Em contraste, métodos supervisionados como Random Forest alcançam F1-scores de 0,81-0,83, mas requerem exemplos rotulados de ambas as classes. O EFC, alcançando F1-Macro de 0,808 (F1 ilícito de 0,77) sob condições realistas de desequilíbrio severo, demonstra desempenho comparável aos métodos supervisionados enquanto oferece vantagens em interpretabilidade e eficiência computacional. Diferentemente de redes neurais profundas que funcionam como caixas-pretas, o EFC fornece decomposições de energia interpretáveis. A eficiência computacional decorre de sua fundamentação em física estatística: o treinamento completa-se em uma única passagem sobre os dados, com complexidade que escala linearmente com amostras e quadraticamente com características.

Esta dissertação demonstrou que o Energy Flow Classifier representa uma abordagem eficaz para detecção de transações fraudulentas de Bitcoin sob condições de escassez de rótulos. O EFC supera substancialmente métodos não supervisionados tradicionais enquanto se aproxima do desempenho de métodos supervisionados completos, oferecendo equilíbrio entre eficácia e requisitos de dados rotulados. A estratégia ótima envolve primeiro reduzir dimensionalidade através de seleção de características ($k \approx 30$) e então aplicar SMOTE ao conjunto de treinamento reduzido. Trabalhos futuros devem esten-

der o EFC para operar sobre embeddings de grafos ou sequências temporais, desenvolver mecanismos adaptativos de threshold, validar em conjuntos de dados de maior escala como Elliptic2, investigar abordagens híbridas combinando a formulação interpretável de energia com representações aprendidas de redes neurais, e avaliar generalização cross-cryptocurrency e aplicabilidade a outros domínios de crime financeiro.

Palavras-chave: criptomoedas, detecção de anomalias, aprendizagem de máquina, Energy Flow Classifier

Abstract

Fraudulent cryptocurrency transactions represent an ongoing and significant threat within the digital asset ecosystem, demanding robust detection mechanisms. Identifying such illicit activities is complicated by the complex nature of transaction data and, critically, the prevalent scarcity of labeled illicit examples in available datasets. This research conducts a comprehensive empirical evaluation of the Energy Flow Classifier (EFC), a physics-inspired one-class anomaly detection model, for identifying illicit Bitcoin transactions using the Elliptic dataset, specifically addressing its performance under conditions of label scarcity. Our findings demonstrate that EFC can effectively distinguish illicit from licit transactions, with its performance significantly improved by combining feature selection and data balancing techniques such as SMOTE, achieving strong results even on imbalanced test sets under conditions of limited labeled illicit examples.

Keywords: criptocurrencies, anomaly detection, machine learning, Energy Flow Classifier

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Money laundering is a high-impact global problem, with criminals laundering billions of dollars annually from serious felonies. In recent years, cryptocurrencies have emerged as a significant channel for these illicit activities, largely due to the pseudonymity they offer criminals. The scale of this threat is substantial; In 2023 alone, illicit addresses received $24.2 billion in cryptocurrency from scams, stolen funds, and other criminal activities [5]. Consequently, developing robust detection mechanisms to protect participants and maintain market integrity is crucial. Machine Learning presents a powerful tool for this task, offering the potential to identify the complex patterns of illicit financial flows, thereby increasing detection rates while decreasing the high false-positive rates common in traditional rule-based systems. Cryptocurrency-related fraud has become a significant threat, causing substantial financial losses and destroying trust in the digital asset ecosystem. These activities not only cause direct monetary damage to individuals and institutions, but also have broader implications, such as undermining the legitimacy of cryptocurrency markets and hindering the widespread adoption of blockchain technology. The need to develop effective methods for detecting and preventing cryptocurrency fraud is crucial to protect participants, maintain market integrity, and ensure sustainable growth of the cryptocurrency industry [6, 7].

The practical application of ML in this domain faces a critical obstacle: the scarcity of labeled data. Supervised learning algorithms, which typically offer high accuracy, are often unfeasible because large-scale publicly available data sets with verified illicit transactions are rare. This scarcity of labels stems from the evolving complexity of money laundering schemes, which makes identifying all illicit actors nearly impossible, and the fact that acquiring labels from law enforcement or expert manual annotation is a costly and slow process [4]. Detecting anomalous patterns within the intricate data streams of cryptocurrency transactions poses a significant challenge. Like many modern datasets, these transactions are characterized by high dimension, evolving characteristics, and sub-

stantial volume, which complicates the application of traditional anomaly detection methods. Addressing the significant challenge of label scarcity inherent in datasets like Elliptic is crucial to developing effective fraud detection systems. Traditional supervised machine learning methods often struggle in such scenarios due to the limited availability of labeled illicit examples. This motivates the exploration of alternative approaches, particularly those capable of learning from predominantly normal data.

In this research, we extend the work in [3] and [4] initially reproducing their findings, extending their research on Elliptic Dataset by applying the Energy Flow Classifier, a novel algorithm rooted in statistical physics. EFC was explored in network intrusion detection [8, 9] context, the EFC was specifically designed to address key limitations of conventional ML classifiers, including the reliance on extensive labeled datasets. Based on the Inverse Potts model, the EFC is a classifier that can operate in both one-class and binary classification modes. In this research, we utilized the available labeled data from both licit and illicit classes to evaluate its effectiveness. It operates on the premise that normal, expected data patterns correspond to low-energy states, while significant deviations, potential anomalies, are constructed as high-energy states. This one-class learning paradigm, combined with its relatively low computational complexity, makes the EFC a promising candidate for detecting fraudulent cryptocurrency activity.

In this research, we evaluate the suitability and performance of EFC for identifying illicit Bitcoin transactions by leveraging its capacity to model normality from available licit data and empirically assessing the effectiveness of the Energy Flow Classifier for identifying illegal transactions within the Elliptic Bitcoin dataset, specifically under conditions of label scarcity. By adapting and applying the EFC to the Elliptic dataset using the available labeled transactions from both classes, this study aims to assess whether it provides a robust and practical alternative to previously studied supervised and unsupervised methods. Our findings show that the Energy Flow Classifier is an effective tool for this problem, especially when its application is thoughtfully combined with data preprocessing strategies. We achieved a F1-Macro score of 0.808 by using the SMOTE data balancing technique. The F1-Macro score was chosen as the primary evaluation metric, following the precedent set by [4]. In their work on this dataset, and because its methodology of averaging the F1-score for each class independently provides a balanced assessment crucial for imbalanced datasets. This result highlights EFC's efficiency, suggesting that it can be competitive with other sophisticated approaches like active learning.

1.1 Research Question and Goals

Research Question: *To what extent is the Energy Flow Classifier (EFC) a viable and effective alternative to conventional machine learning models for detecting illicit Bitcoin transactions given the real world challenge of label scarcity?*

- **Main Objective** To empirically evaluate the effectiveness of the Energy Flow Classifier (EFC), on the identification of illicit Bitcoin transactions in the Elliptic dataset, particularly under the real-world constraint of label scarcity.

  Specific Objectives:

  - To adapt and apply the Energy Flow Classifier (EFC), a model originally from network intrusion detection, to the domain of cryptocurrency fraud, training it on the available labeled data to establish a baseline performance on the raw, imbalanced Elliptic dataset.
  - Systematically investigate the impact of various data balancing techniques including undersampling, oversampling, and the synthetic minority oversampling technique (SMOTE) on the EFC's ability to classify illicit transactions.
  - Analyze the effect of dimensionality reduction on EFC's performance by applying feature selection (SelectKBest) to identify an optimal subset of features for distinguishing between licit and illicit activity.
  - Assess the performance of a combined strategy that integrates feature selection with data balancing (SMOTE) to determine if this synergistic approach yields superior classification results on a realistic, unbalanced test set.
  - To compare the performance of the optimized EFC configuration against established methodologies from previous work, providing a conclusive answer on its viability as a practical tool for fraud detection in label-scarce cryptocurrency environments.

As a baseline, for comparison purposes, we replicated the work in [4] on Appendix III.1. Achieving results very close to the numbers found in the original paper, the difference is minimum. We believe this is due to computational differences.

1.2 Scope

This research focuses on evaluating the Energy Flow Classifier (EFC) in the context of cryptocurrency fraud detection, specifically under label scarcity conditions. The scope is limited to the original Elliptic dataset (hereafter referred to as Elliptic1), which, despite

its constraints, provides a structured and well-documented benchmark for developing and validating anomaly detection models in blockchain environments. The study addresses realistic scenarios where confirmed illicit labels are limited but not entirely absent, reflecting the practical challenge of severe class imbalance (90% licit, 10% illicit in labeled data) rather than complete absence of illicit examples. However, the use of node-only features and the exclusion of temporal and relational aspects of the full transaction graph impose certain limitations on the ability of the model to capture network-based fraud patterns.

1.3 Dissertation Structure

The dissertation is organized to guide the reader through this investigation in a logical sequence. Following this introductory chapter, Chapter 2 provides technical and contextual background, including an overview of Bitcoin, the problem of money laundering, and the properties of the Elliptic datasets. Chapter 3 offers a detailed review of related research on both supervised and unsupervised methods for illicit transaction detection. Chapter 4 introduces the theoretical foundations and implementation of the EFC, including its mathematical formulation and algorithmic procedures. Chapter 4 also presents the experimental configuration and empirical results derived from applying the EFC to the data set, discussing industrial and academic implications alongside threats to validity. Finally, Chapter 5 summarizes the key findings, discusses their implications, and outlines potential directions for future research.

Chapter 2

Background

2.1 Bitcoin

Bitcoin is a decentralized digital currency. Decentralized means that it is not controlled or owned by any government, central bank, corporation, or other institution. Instead, Bitcoin is managed by computer software that anyone with access to the internet can download and use to monitor and verify transactions.

Bitcoin transactions constitute the fundamental mechanism by which value is transferred within the Bitcoin network. Each transaction is a cryptographically signed data structure that encodes the transfer of bitcoin ownership from one or more input addresses to one or more output addresses. These transactions are broadcast to the network, validated by nodes using consensus rules, and ultimately recorded in the immutable public ledger known as the blockchain. Technical components such as digital signatures, transaction identifiers, and the UTXO model are further explained in Appendix II.1, which provides detailed insights into the cryptographic and consensus mechanisms that underlie Bitcoin transactions.

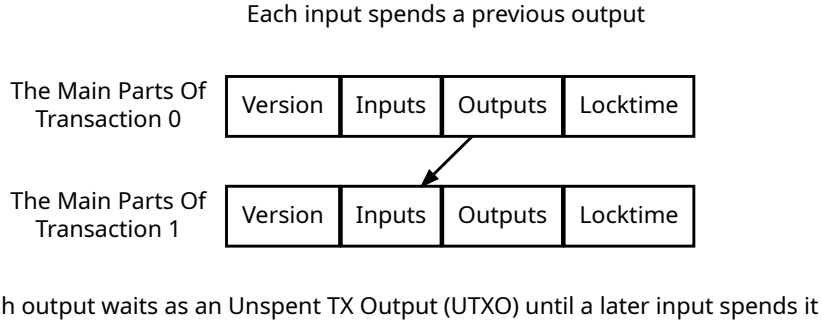


Figure 2.1: Schematic representation of a Bitcoin transaction's data structure. Taken from [1].

- **Inputs:** References to previously unspent transaction output (UTXO) that serves as the source of the transferred value. Each input includes a cryptographic signature that authorizes the expenditure of the referenced UTXO.
- **Outputs:** Specifies the recipient addresses and the amount of bitcoin allocated to each. The output creates new UTXOs which can subsequently be spent in future transactions.
- **Transaction ID (TxID):** A unique identifier derived from a double SHA-256 hash of the transaction data, ensuring integrity and non-repudiation.

Before a transaction is added to the blockchain, it is subjected to rigorous validation by the network nodes. This process ensures compliance with consensus rules, including verification of digital signatures, prevention of double spending, and adherence to syntactic correctness. Once validated, the transaction is propagated across the peer-to-peer network via a gossip protocol, ultimately reaching miners who include it in a candidate block.

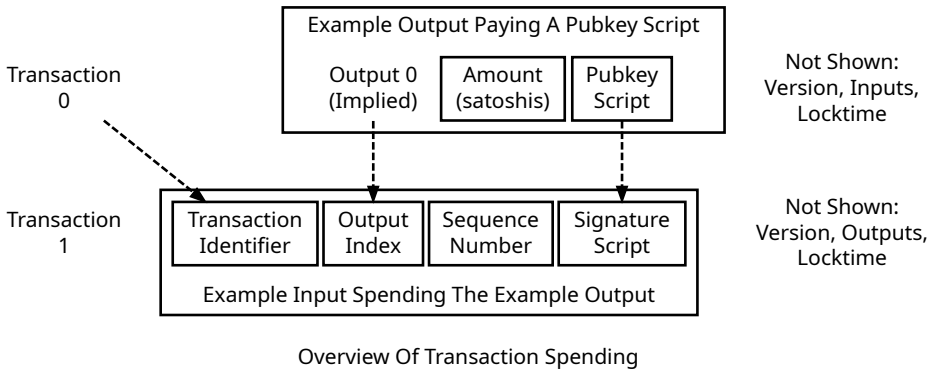


Figure 2.2: Flowchart depicting the transaction validation and propagation process. Taken from [1].

A typical Bitcoin transaction can be found at the following address Bitcoin Transaction Hash ID: f89f6dea8abc818e27a988fc45932de95bfc1e9e04c0455ca17bd9fed9cd348e.

In this research, Bitcoin transactions serve as the core input data for anomaly detection using the Energy Flow Classifier. Each transaction is treated as a data point described by a set of features derived from the Elliptic dataset, including attributes related to the structure of the transaction, such as the number of inputs and outputs, transaction fee, and temporal position on the blockchain. These features are abstracted into a numerical representation that captures transactional behavior without directly revealing the identities involved. By modeling the statistical distributions of both licit and illicit transactions in the training data, the EFC learns to distinguish patterns that characterize each class.

During the evaluation, transactions that deviate significantly from this learned energy distribution are flagged as anomalous, with the aim of identifying potentially illicit behavior within the Bitcoin network.

2.2 Bitcoin and Money Laundering

Money laundering is a financial crime that involves the process of concealing the illicit origins of funds derived from illegal activities, such as drug trafficking, corruption, or fraud. The objective is to transform dirty money into ostensibly legitimate assets by obscuring its source through a series of complex transactions [10]. The process typically comprises three stages: placement (introducing illicit funds into the financial system), layering (disguising the trail through multiple transactions), and integration (reintroducing laundered funds into the economy as clean capital). Regulatory frameworks, such as the U.S. Bank Secrecy Act (1970) and the Financial Action Task Force (FATF) recommendations, aim to combat these practices by enforcing anti-money laundering (AML) measures, including customer due diligence and suspicious activity reporting.

The literature shows no clear consensus on evaluation methodologies. Different studies report different metrics using incompatible preprocessing approaches, preventing meaningful comparison across methods. Several works have highlighted that the performance of graph-based models such as GCN, GAT, and HGT for illicit transaction detection varies considerably depending on the dataset version and labeling criteria [11]. Other comparative evaluations of anomaly detection techniques demonstrate that even under similar experimental configurations, model performance changes significantly when data distribution shifts occur, revealing a lack of standardization in evaluation methodologies [12]. Additional studies emphasize that fraud detection research often lacks unified fairness or robustness metrics, with many evaluations focusing solely on accuracy-oriented indicators rather than the broader trade-offs between false positives, false negatives, and operational constraints [13].

Bitcoin's pseudonymous nature and decentralized architecture have introduced novel challenges in combating money laundering, particularly in the context of cybercrime and trade-based financial obfuscation. Unlike traditional financial systems, where intermediaries enforce anti-money laundering (AML) controls, Bitcoin transactions rely on cryptographic addresses that obscure participant identities unless explicitly linked to real-world entities [14]. This structural divergence enables layered laundering strategies in which illicit funds are dispersed across multiple wallets or mixed through privacy-enhancing tools such as CoinJoin. Empirical studies highlight that while Bitcoin's transparency—through

its public ledger—allows forensic tracing, sophisticated actors exploit jurisdictional arbitrage and decentralized exchanges to avoid detection [15].

Although public blockchains allow for end-to-end visibility of transaction flows, attribution of real-world identity to blockchain addresses remains a major limitation. Transactions are publicly recorded, yet participants are represented by pseudonymous identifiers, which inherently decouple ledger entries from real identities unless off-chain data or regulatory disclosures are used for linkage [16]. Even in systems where all transaction data is visible, confidentiality concerns arise because transaction data exposes metadata that can be used to infer behavior or relationships among addresses, but without confirming who is behind them [17]. Furthermore, because regulatory and compliance frameworks often rely on identity verification external to the blockchain (e.g. KYC in exchanges), public visibility of transaction histories does not directly translate into accountability without extra-protocol mechanisms [16].

Trade-based money laundering (TBML) techniques, traditionally reliant on invoice manipulation and shell companies, have also adapted to cryptocurrencies. Bitcoin's programmability facilitates automated layering through smart contracts or tokenized assets, complicating compliance efforts [18]. For example, the *Hawala* system's informal value transfer mechanisms find digital analogs in peer-to-peer (P2P) Bitcoin networks, where unregulated exchanges mediate cross-border flows without scrutiny. However, there is mixed evidence regarding Bitcoin's dominance in laundering; some studies suggest its traceability limits large-scale adoption by criminals, while others document its role in laundering cybercrime proceeds, particularly ransomware and darknet markets [14].

The regulatory response to Bitcoin-related laundering remains fragmented, with jurisdictions adopting varying stances on cryptocurrency oversight. Blockchain analytics firms have emerged to bridge this gap, deploying clustering algorithms to de-anonymize transactions. However, innovations such as privacy coins and decentralized mixers persistently challenge these tools [19]. Future research must address the dynamic interplay between technological advancements and regulatory frameworks, particularly as decentralized finance (DeFi) platforms expand laundering avenues beyond Bitcoin's native capabilities.

Bitcoin transactions can be of two types: licit (exchanges, wallet providers, miners, licit services, etc.). Illicit (scams, malware, terrorist organizations, ransomware, Ponzi schemes, etc.). A given transaction is deemed licit (versus illicit) if the entity initiating the transaction (i.e., the entity controlling the private keys associated with the input addresses of a specific transaction) belongs to a licit (illicit) category, for simplicity, this argument ignores mixer transactions where the inputs are controlled by multiple entities. Importantly, all features are constructed using only publicly available information. The Elliptic dataset from which these transactions are extracted and used throughout this

research is described in the following section.

2.3 Energy Flow Classifier

This section introduces the Energy Flow Classifier (EFC), a classification method inspired by the inverse Potts model from statistical physics [8]. The EFC framework uses inverse statistics to construct a probabilistic model from labeled training samples. This approach circumvents the common challenge of acquiring large, labeled datasets and provides an interpretable white-box model, in contrast to the black-box nature of many machine learning algorithms.

The EFC adapts the principles of the Potts model, which describes the interactions of spins on a crystalline lattice. Let us use the context of intrusion detection as an example. Following this analogy, a network flow is represented as a fully connected graph, where each node corresponds to a specific flow feature (e.g., Protocol, Duration, Src Port).

Let a network flow k be represented by a vector of N feature values, $(a_{k1}, a_{k2}, ..., a_{kN})$, where each a_{ki} is a value for the i-th feature. The core of the EFC is the Hamiltonian, or "energy" of a flow, which quantifies its statistical typicality with respect to a learned model of benign traffic. The energy $\mathcal{H}$ of a flow k is defined by the following equation:

$$\mathcal{H}(a_{k1}, ..., a_{kN}) = -\sum_{i<j} e_{ij}(a_{ki}, a_{kj}) - \sum_{i} h_i(a_{ki}) \tag{2.1}$$

Here, $h_i(a_{ki})$ represents the "local field," which is the energetic contribution of the value a_{ki} to the feature i. The term $e_{ij}(a_{ki}, a_{kj})$ represents the "coupling", which captures the statistical dependence between the values of the feature pair (i, j). A flow that is statistically similar to the benign flows used for training will exhibit low energy, whereas an anomalous flow will have high energy.

2.3.1 EFC Model Inference

The inference process aims to determine the local fields h_i and couplings e_{ij} of a training set composed solely of benign flow samples. This is achieved by applying the Maximum Entropy Principle, which finds the probability distribution P that is most consistent with the observed data while being maximally non-committal otherwise [8].

The model is constrained to reproduce the empirical single and joint frequencies of feature values observed in the training data. The frequency of a single feature, $f_i(a_i)$, is the frequency of occurrence of the value a_i for feature i. The joint frequency, $f_{ij}(a_i, a_j)$, is the frequency of occurrence of values a_i and a_j for features i and j. To mitigate the effects of undersampling, pseudocounts are added to these empirical frequencies [8].

The couplings e_{ij} are inferred using a Gaussian approximation, which effectively removes the influence of indirect correlations. This is achieved by inverting the covariance matrix C, where each element is defined as:

$$C_{ij}(a_i, a_j) = f_{ij}(a_i, a_j) - f_i(a_i)f_j(a_j) \tag{2.2}$$

The negative of the inverse of this matrix then gives the couplings:

$$e_{ij}(a_i, a_j) = -(C^{-1})_{ij}(a_i, a_j) \tag{2.3}$$

Once the couplings are determined, the local fields h_i are inferred using a mean-field approximation. This method approximates the complex interactions of a feature with all the others by an average interaction, simplifying the calculation [8]. The local field for a feature value a_i is calculated as:

$$h_i(a_i) = \ln\left(\frac{f_i(a_i)}{f_i(Q)}\right) - \sum_{j,a_j} e_{ij}(a_i, a_j)f_j(a_j) \tag{2.4}$$

where Q is a reference state used for normalization.

2.3.2 Classification

With the inferred model parameters (e_{ij} and h_i), the energy of any new unlabeled flow can be calculated using Equation 2.1. The classification is then performed by comparing this energy with a predetermined threshold. This threshold is typically set at a high percentile (e.g., the 95th) of the energy distribution of the benign training samples.

A flow is classified as follows:

- **Benign**, if its energy is less than or equal to the threshold.
- **Malicious**, if its energy is greater than the threshold.

This energy-based approach enables the EFC to quantify statistical deviations between transaction classes, providing an interpretable measure for distinguishing illicit from licit behavior.

Furthermore, the multi-class version of EFC extends this principle by training separate models for each known class of traffic (e.g., Benign, DoS, PortScan). To classify a new network flow, its energy is calculated for each class model. The flow is assigned the label of the class for which it has the lowest energy, provided that this energy is below that class's specific threshold. If the lowest energy is still above all thresholds, the flow is labeled as **Suspicious**, indicating a potential unknown attack type. This gives EFC powerful open-set recognition capabilities [9].

2.3.3 Efficiency of the Energy Flow Classifier

EFC's computational efficiency stems from its statistical physics foundation. The model requires no iterative optimization—energy functions are derived analytically once empirical frequencies are computed. Training completes in a single pass over the data, computing frequencies and covariance statistics that scale linearly with samples and quadratically with features. Unlike deep neural networks that rely on backpropagation through complex architectures, EFC uses only matrix operations and summations. The learned model size depends solely on the number of features and bin categories, not on training set size.

In contrast, many state-of-the-art anomaly detection methods, such as Graph Neural Networks (GNNs) or Autoencoders, require multiple forward/backward passes over mini-batches of data and often rely on GPU acceleration. EFC, for comparison, runs comfortably on commodity CPUs and completes training in seconds to minutes, even with thousands of samples and dozens of features.

2.3.4 Phases of the Algorithm

The operational lifecycle of the Energy Flow Classifier can be divided into three distinct phases: Training, inference, and testing. The pseudocode for each phase is detailed below.

The training phase is the core of the model-building process. It begins by analyzing a dataset composed exclusively of normal data—for instance, benign network flows—to learn the statistical properties of the common samples in the dataset (assuming that anomalous data is infrequent). The procedure calculates the single and joint frequencies of all feature values, which are then used to compute a covariance matrix. By inverting this matrix, the algorithm determines the coupling values (e_{ij}), which represent direct correlations between pairs of characteristics. Subsequently, it infers local fields (h_i), representing the intrinsic properties of individual features. Finally, it calculates the energy for each sample in the training set to establish an energy distribution for normal data—e.g., benign traffic —and sets a classification threshold, typically the 95th percentile. The output of this phase is the complete statistical model, consisting of the couplings, local fields, and the energy threshold.

The inference phase, also known as the classification phase, is where the trained EFC model is applied to new unlabeled data (e.g., network flows). For each incoming data sample, this procedure calculates its total energy by summing the contributions from the precomputed local fields (h_i) and the coupling values (e_{ij}) corresponding to the specific characteristic values present in that flow. This calculated energy measures how much the flow deviates from the established norm of the normal data. The energy value is then

Algorithm 1 EFC Training Phase

procedure TRAINEFC($benign\_flows, Q, \alpha$)
  $f_i \leftarrow$ CalculateSingleFrequencies($benign\_flows, Q, \alpha$)
  $f_{ij} \leftarrow$ CalculateJointFrequencies($benign\_flows, f_i, Q, \alpha$)
  $C \leftarrow$ ComputeCovarianceMatrix(f_i, f_{ij})
  $C^{-1} \leftarrow$ InvertMatrix(C)
  $e_{ij} \leftarrow -C^{-1}$
  $h_i \leftarrow$ InferLocalFields(e_{ij}, f_i, Q)
  $training\_energies \leftarrow$ CalculateEnergies($benign\_flows, e_{ij}, h_i$)
  $threshold \leftarrow$ Percentile($training\_energies, 95$)
  return $e_{ij}, h_i, threshold$
end procedure

compared to the threshold determined during the training phase. If the energy exceeds the threshold, the flow is classified as anomalous; otherwise, it is classified as normal.

Algorithm 2 EFC Inference Phase

procedure INFEREFC($flow, e_{ij}, h_i, threshold$)
  $energy \leftarrow 0$
  for $i \leftarrow 1$ to $N - 1$ **do**
    $a_i \leftarrow flow[i]$
    for $j \leftarrow i + 1$ to N **do**
      $a_j \leftarrow flow[j]$
      $energy \leftarrow energy - e_{ij}[i, a_i, j, a_j]$
    end for
    $energy \leftarrow energy - h_i[i, a_i]$
  end for
  if $energy > threshold$ **then**
    return MALICIOUS
  else
    return BENIGN
  end if
end procedure

The testing phase is designed to evaluate the performance and precision of the trained EFC model. This procedure takes a labeled test set containing both normal and anomalous data (e.g., network flow) as well as the trained model as input. It iterates through each flow in the test set, using the inference procedure to generate a classification for each one. These predictions are then compared against the ground truth labels of the test data. Based on this comparison, standard performance metrics such as F1-score, precision, and recall are calculated. This phase is crucial for validating the model's effectiveness and its ability to generalize to unseen data.

Algorithm 3 EFC Testing Phase

procedure TestEFC($test\_set, e_{ij}, h_i, threshold$)
    $predictions \leftarrow []$
    $true\_labels \leftarrow []$
    for each $flow, label$ in $test\_set$ **do**
        $prediction \leftarrow$ InferEFC($flow, e_{ij}, h_i, threshold$)
        Add $prediction$ to $predictions$
        Add $label$ to $true\_labels$
    end for
    $performance\_metrics \leftarrow$ Evaluate($predictions, true\_labels$)
    return $performance\_metrics$
end procedure

The Energy Flow Classifier proved to be a robust and principled approach to network intrusion detection [8, 20]. By focusing its methodology on statistical physics, it offers several advantages over traditional machine learning techniques. Its ability to train on benign-only data alleviates significant operational hurdles in data collection and labeling. The resulting model is inherently interpretable, allowing security analysts to understand the specific interaction between characteristics that lead to a classification decision. Furthermore, its strong performance in domain adaptation and its capacity for open-set recognition make it a promising tool for identifying not only known threats but also new and evolving cyberattacks, with specific applications in botnet detection [21]. Future work may explore dynamic thresholding mechanisms and the application of EFC in distributed detection environments.

2.4 The Elliptic dataset

Elliptic is a cryptocurrency intelligence company focused on protecting cryptocurrency ecosystems from criminal activity. The Elliptic dataset is a graph network of Bitcoin transactions with hand-crafted features. The Elliptic dataset maps Bitcoin transactions to real entities belonging to licit categories (exchanges, wallet providers, miners, licit services, etc.) versus illicit ones (scams, malware, terrorist organizations, ransomware, Ponzi schemes, etc.). From the raw Bitcoin data, a graph is constructed and labeled such that the nodes represent transactions and the edges represent the flow of Bitcoin currency (BTC) going from one transaction to the next. This dataset is a publicly available graph dataset of Bitcoin transactions introduced by [3] and subsequently used in foundational studies on machine learning for the detection of Bitcoin money laundering, including the work by [4] which highlighted the challenges of label scarcity in the Elliptic dataset. The dataset represents a temporal subgraph of the public Bitcoin blockchain, focusing

on transactions involving entities identified by Elliptic Ltd., a company specializing in blockchain analytics and financial crime prevention. It captures transaction patterns over 49 distinct time steps, where each step corresponds roughly to a two-week period. The complete dataset comprises 203,769 transaction nodes and 234,355 directed edges representing the flow of Bitcoin between transactions.

Each transaction (node) in the graph is described by a set of 166 anonymized features. An explicit feature denotes the time steps (1 to 49). The remaining 165 features are local transactional properties, including aggregated information about the transaction's inputs and outputs (e.g., number, amounts, fees) and potentially aggregated statistics from its immediate neighborhood in the transaction graph. These features are provided in a normalized or standardized form, obscuring raw values but preserving crucial relational patterns for machine learning analysis. The graph structure itself, defined by the edges connecting transactions where the output of one becomes the input of another, provides contextual information about the flow of funds although our EFC implementation primarily focuses on the node features. A key characteristic of the Elliptic dataset is its label scarcity—while the entire dataset contains over 200,000 transactions, only a subset of 46,564 transactions is explicitly labeled. Table 2.1 summarizes the Elliptic dataset.

Table 2.1: Summary Statistics of the Elliptic dataset (based on [3]).

Characteristic	Value
Total of Transactions (Nodes)	203,769
Total of Edges	234,355
Time Steps	49
Features per Node	166
Labeled Transactions	46,564 (~23%)
- Licit	42,019 (~90.2% of labeled)
- Illicit	4,545 (~9.8% of labeled)
Unlabeled Transactions	157,205 (~77%)

The temporal dimension of the dataset constitutes a critical feature of the Elliptic dataset. Each node is annotated with a timestamp corresponding to the approximate confirmation time of the transaction within the Bitcoin network. The dataset comprises 49 discrete temporal intervals, uniformly distributed with approximately biweekly periodicity. Within each interval, transactions form a single connected component occurring within a tight temporal window (less than three hours between consecutive transactions), with no intertemporal edges connecting different intervals. This structure implies that nodes within a given interval share nearly identical timestamps, effectively representing discrete temporal snapshots of network activity. The distribution of nodes across intervals

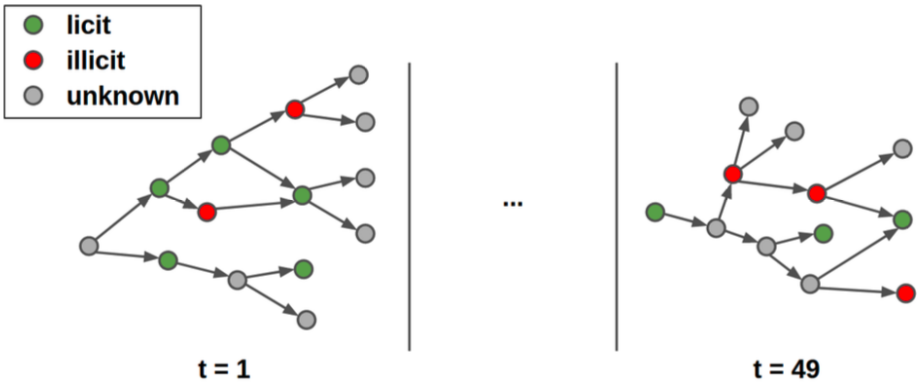


Figure 2.3: Structure of the dataset (taken from [2]).

remains relatively stable, with cardinality ranging from 1,000 to 8,000 nodes per interval. See Figure 2.3.

Although Bitcoin transactions are inherently traceable at the protocol level, malicious actors often engage in sophisticated behavior to evade detection. This includes using chains of intermediate transactions, routing funds through cross-chain bridges, or leveraging mixing services and privacy-focused wallets. These behavioral signatures are not always evident from the raw transaction features, but can manifest themselves as statistical outliers or deviations in transaction metadata. Capturing these subtle irregularities requires analytical models that can learn discriminative patterns from imbalanced labeled data, as demonstrated by classifiers like the EFC when applied to fraud detection tasks.

The temporal dimension of the dataset includes 49 graphs sampled from the Bitcoin blockchain at different sequential moments in time (time steps), as presented in Figure 2.4.

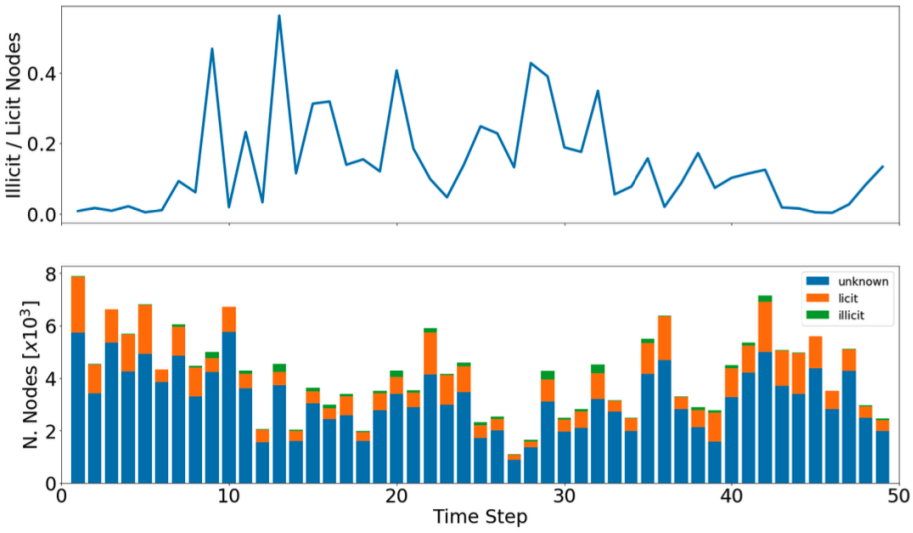


Figure 2.4: (Top) Fraction of illicit vs. licit nodes at different time steps in the dataset. (Bottom) Number of nodes vs. time step (taken from [3]).

Chapter 3

Related Work

The rise of cryptocurrencies, particularly Bitcoin, has been accompanied by an increase in their use for illicit activities due to their decentralized and pseudo-anonymous nature. Consequently, a significant body of research has emerged that focuses on the detection and analysis of these illicit transactions within the Bitcoin blockchain. This chapter reviews the state-of-the-art in this domain, with a particular focus on the application of machine learning and deep learning techniques.

The vast and complex nature of the Bitcoin transaction network makes manual analysis for illicit activities practically impossible. Machine learning has therefore become an indispensable tool for researchers and law enforcement agencies [22]. The literature in this area can be broadly categorized into supervised and unsupervised learning methods, with some research exploring semi-supervised or hybrid approaches to overcome the inherent challenges of blockchain data.

In reviewing the literature, existing methods can be broadly categorized across three axes: data dependence (supervised vs. unsupervised), model structure (black box vs. interpretable), and data representation (node-based vs. graph-based). Although supervised methods dominate early research because of their higher accuracy, they are unsuitable in label-scarce contexts. Unsupervised models offer generalizability, but often fall short in precision or human interpretability. Our goal is to relate the state of the art algorithms with the EFC, exploring gaps and possible enhancements this algorithm can achieve, especially under label scarcity and Bitcoin transactions.

3.1 Supervised Learning Methods

Supervised learning has become a common approach for identifying illicit activities in the Bitcoin network [22]. These methods rely on labeled datasets, where transactions or addresses are pre-identified as either licit or illicit. One of the main challenges in this area

is the scarcity of labeled data, as identifying and confirming illicit activities is a costly and time-consuming process [4].

Several studies have conducted comparative analyzes of various supervised learning algorithms. For example, in [23] the performance of Random Forest, Extra Trees, Gradient Boosting, Bagging Classifier, AdaBoost and k-Nearest Neighbors on the Elliptic dataset is evaluated. Their findings indicate that ensemble methods, particularly a combination of Random Forest, Extra Trees, and Bagging classifiers, outperform individual classical models, achieving high accuracy and F1-scores in predicting illicit transactions.

Similarly, Elmougy et al. report the results of an empirical study that uses GPU-accelerated machine learning models (e. g., Support Vector Machines, Random Forest, and Logistic Regression) on both Bitcoin and Ethereum networks [24]. Their findings underscore the effectiveness of these models in detecting fraudulent transactions, with SVM and Random Forest showing robust performance. The use of GPU acceleration is a novel contribution, as it addresses the computational challenges associated with analyzing massive blockchain datasets. This consideration is also pertinent to the scalability of the methods proposed in this dissertation.

The importance of feature engineering is a recurring theme in the supervised learning literature. The performance of these models is highly dependent on the quality and relevance of the features extracted from the blockchain data. The features can be local to a single transaction (for example, transaction fees, number of inputs/outputs) or aggregated from the transaction's neighborhood on the graph [23].

Relation to EFC: Supervised methods require labeled examples from both licit and illicit classes for training. While supervised methods such as Random Forest achieve F1-scores of 0.81–0.83 [4, 23], they cannot function without labeled illicit examples. EFC addresses this limitation by learning the statistical distribution of normal behavior and flagging transactions that deviate from this learned model. Additionally, ensemble methods and GPU-accelerated models [24] often function as black boxes, whereas EFC provides interpretable energy decompositions that explain why a transaction was flagged [8, 20].

Despite the variety of models explored in the literature, there is limited consensus on how to systematically evaluate or compare their trade-offs. Most studies report subsets of performance metrics (e.g., accuracy, F1, or recall) under differing preprocessing strategies, dataset splits, and imbalance conditions, which hinders direct comparison. A systematic review of blockchain anomaly detection methods highlights that published works rely on heterogeneous definitions of anomalies, inconsistent data partitions, and diverse evaluation metrics, preventing the establishment of standardized benchmarks [25]. Similar findings are reported in a comprehensive review of financial fraud detection research, where the authors observed that most studies employ distinct data preprocessing and

sampling procedures and report only a subset of performance indicators, making cross-study comparison difficult [26]. Furthermore, it has been shown that evaluations of fraud detection systems often overlook fairness and robustness trade-offs, focusing primarily on accuracy-based indicators rather than operational costs associated with false positives and false negatives [13].

3.2 Unsupervised Learning Methods

Unsupervised learning methods are particularly attractive for blockchain analysis due to the scarcity of labeled data. These techniques aim to identify anomalies or suspicious patterns without prior knowledge of illicit activities. Anomaly detection is a common application, where the goal is to identify transactions or addresses that deviate significantly from the norm [27].

A significant portion of unsupervised research is focused on clustering. Early work used K-means clustering and a role extraction algorithm (RolX) to identify hubs with anomalous transaction values and variances, suggesting their potential involvement in mixing services [28]. More recent studies have refined this approach: trimmed K-means clustering demonstrated improved resilience to outliers and spurious groupings when applied to Bitcoin transaction data [29].

Pham and Steven [27] also explored a range of unsupervised techniques, including k-means clustering, Mahalanobis distance, and unsupervised SVM, to detect suspicious users and transactions. The authors represented the data as both a user graph and a transaction graph, allowing for a dual perspective on the network's activity. Their findings indicate that abnormal behavior often corresponds to extreme values in the extracted characteristics, and they were able to identify known thieves using their methods.

Despite these successes, some studies have highlighted the limitations of purely unsupervised methods. The authors in [4] found that in the Elliptic dataset, illicit transactions often do not appear as outliers and instead hide within clusters of licit behavior. This suggests that sophisticated criminals may attempt to mimic normal transaction patterns to evade detection. This finding is a critical consideration for our research, indicating that relying solely on unsupervised anomaly detection may not be sufficient to identify all forms of illicit activity.

Relation to EFC: Traditional unsupervised methods such as Isolation Forest, Local Outlier Factor (LOF), and One-Class SVM achieve F1-scores of 0.00–0.19 on the Elliptic dataset [4], demonstrating their limited effectiveness. These methods assume illicit transactions manifest as clear outliers in feature space, an assumption that Lorenz et al. [4] demonstrated to be false. EFC differs from these approaches through its physics-inspired

formulation based on the inverse Potts model [8]. Rather than relying on distance metrics (LOF), isolation principles (Isolation Forest), or kernel-based boundaries (One-Class SVM), EFC models the statistical dependencies between features through learned couplings and local fields. This allows EFC to detect subtle deviations from normal behavioral patterns even when illicit transactions do not appear as extreme outliers. Furthermore, while methods like Isolation Forest and Autoencoders lack interpretability [27], EFC provides energy decompositions that quantify the contribution of individual features and feature interactions to anomaly scores [8, 20].

Another critical flaw in many unsupervised and hybrid models is the lack of interpretability. Models like Isolation Forest or Autoencoders offer little insight into why a transaction is flagged as anomalous, making them harder to justify in regulatory or compliance contexts.

3.3 Hybrid and Semi-Supervised Approaches

Given the strengths and weaknesses of both supervised and unsupervised learning, some researchers have explored hybrid or semi-supervised approaches. These methods aim to take advantage of the large amounts of unlabeled data available on the blockchain while also taking advantage of the smaller amounts of labeled data that may be available.

A prominent hybrid strategy involves using unsupervised algorithms to first structure or group the data, followed by supervised learning for classification. For example, [30] used PCA for dimensionality reduction followed by k-means clustering to identify transaction groups that included common user behavior, mining and potential illicit activity. This structure helped isolate patterns for later analysis. Similarly, [27] explored clustering in combination with supervised techniques such as SVM to detect suspicious users and transactions, using both transaction and user graph perspectives. These hybrid methods demonstrate how unsupervised techniques can be leveraged for initial structuring, thereby improving the efficacy of subsequent supervised classifiers even when labeled data are limited.

We start this research reproducing their findings, first creating a supervised baseline with XGBoost, Random Florest, Logistic Regression serving also as a compartive baseline for our findings. After that, we then reproduce an anomaly detection benchmark with PCA, LOF, CBLOF, Isolation Forest, KNN, ABOD and One-Class SVM. Successfully reproducing the findings in [4]. This research proposed an active learning solution to address the issue of label scarcity. Their method starts with a small number of labeled instances, and iteratively queries the most informative unlabeled instances for manual labeling. They showed that this approach could match the performance of a fully super-

vised model using only a fraction of the labels. This is a highly practical approach that could be adapted for the work in this dissertation, potentially reducing the reliance on large, pre-labeled datasets.

The use of unsupervised methods for feature engineering or data pre-processing is another promising hybrid approach. The authors in [31] combined unsupervised Gaussian Mixture Models (GMM) for data filtering with supervised classifiers such as Random Forests and KNN. They found that this pre-processing step could improve the performance of classifiers in predicting financial market movements.

Finally, a systematic review [22] suggests that combining unsupervised learning with topological analysis features can lead to more accurate models. Topology-based features, which capture the structure of the transaction graph, can help expose disguised illegal activities that might otherwise be missed. This aligns with the broader consensus in the literature that a multi-faceted, often hybrid, approach is the most effective strategy for tackling the complex problem of illicit transaction detection in the Bitcoin network.

Relation to EFC: Hybrid and semi-supervised approaches attempt to bridge the gap between fully supervised and unsupervised methods. Active learning [4] reduces labeling requirements but still necessitates iterative human annotation and infrastructure for querying informative samples. EFC offers a different solution: it can be trained on available labeled data from both classes (as demonstrated in this dissertation) or operate in true one-class mode using only licit data [8, 32]. This flexibility makes EFC adaptable to varying label availability scenarios without requiring active learning infrastructure. While hybrid methods combining unsupervised clustering with supervised classification [27, 30] require multi-stage pipelines, EFC provides a single unified framework based on energy minimization. The active learning approach [4] achieved competitive performance by strategically selecting samples for labeling, but EFC achieves comparable results (F1-macro 0.81) without requiring iterative human feedback during training. Additionally, EFC's energy-based formulation naturally incorporates feature dependencies through learned couplings [8], providing functionality similar to topology-based feature engineering [22] but within an interpretable statistical physics framework rather than as separate preprocessing steps.

3.4 Deep Learning Methods

With advances in computational power and the increasing complexity of financial data, deep learning has emerged as a powerful paradigm for fraud and anomaly detection. These methods can automatically learn hierarchical feature representations from raw data, often outperforming traditional machine learning models based on handcrafted features [33].

This is particularly advantageous in the blockchain domain, where feature engineering can be complex and time-consuming.

Graph Neural Networks (GNNs) are particularly well suited for blockchain analysis, given the inherent graph structure of the transaction data. Instead of treating transactions as isolated data points, GNNs leverage the connections between them to learn more contextually rich representations. The research in [34] proposed an end-to-end GNN model using spectral graph convolutions to classify illicit transactions, achieving high accuracy on a large-scale dataset. A key advantage of this approach is its ability to directly model the relationships and flows within the transaction graph, performing both binary (legal vs. illegal) and multi-class classification to pinpoint specific user categories. The success of GNNs suggests that capturing blockchain topological information is critical to effective detection.

To address the pervasive issue of label scarcity, self-supervised learning on graphs has shown significant promise. The research [35] introduced Inspection-L, a framework that operates in two stages. First, it uses a self-supervised GNN, combining a Graph Isomorphism Network (GIN) with a Deep Graph Infomax (DGI) objective, to learn powerful node embeddings from the graph structure without relying on any labels. These embeddings capture both local and global graph information. In the second stage, these learned embeddings are used as augmented features to train a downstream supervised classifier, such as a Random Forest. This approach outperformed state-of-the-art supervised GNNs on the Elliptic dataset, demonstrating the potential of self-supervised methods to leverage vast amounts of unlabeled data for improved performance.

Despite the power of custom deep learning models, their performance must be contextualized. Authors in [33] conducted a comparative study of a Deep Neural Network (DNN) against Random Forest, KNN, and Naive Bayes for money laundering detection in the Elliptic dataset. They found that both the DNN and Random Forest achieved high accuracy, with the Random Forest (F1-score of 0.99) slightly outperforming the DNN (F1-score of 0.98). This highlights that while deep learning models are powerful, well-tuned classical ensemble methods can still be highly competitive. A comprehensive review by Hisham et al. (2022) further supports this, noting that ensemble methods often form the basis of high-performance detection systems, sometimes incorporating deep learning models as components within the ensemble [36]. The choice between traditional and deep learning models may therefore depend on factors such as dataset size, feature complexity, and computational resources.

Overall, the trend in the literature is moving towards more sophisticated, data-driven models that can learn complex patterns directly from the blockchain. Deep learning, particularly in combination with graph-based, self-supervised, and hybrid techniques, repre-

sents a promising frontier for cryptocurrency fraud detection. However, these approaches often come with significant computational overhead, require specialized hardware infrastructure, and—critically—still depend on the availability of at least some labeled data to achieve optimal performance. Furthermore, their black-box nature can limit interpretability, which poses challenges in regulated financial environments where explainability is essential for compliance and auditability.

Given these constraints, and considering the central challenge of label scarcity highlighted throughout the literature, there remains a compelling case for investigating simpler, more interpretable classification methods that can operate effectively under conditions of severe class imbalance and limited labeled examples. Such approaches, while potentially less sophisticated in their modeling capacity, offer distinct advantages in terms of computational efficiency, transparency, and practical deployability—particularly in resource-constrained environments or during early-stage fraud detection system development.

Relation to EFC: Deep learning methods, particularly GNNs [3] and self-supervised approaches [35], achieve the highest reported performance on the Elliptic dataset (F1-scores of 0.85–0.87). However, these methods come with distinct trade-offs compared to EFC. GNNs require GPU infrastructure and substantial computational resources for training [24], whereas EFC trains in a single pass over the data using only CPU resources [8, 20]. The training complexity of GNNs scales with both the number of nodes and the depth of the network, while EFC's complexity scales quadratically with the number of features and linearly with the number of samples [8]. Self-supervised methods like Inspection-L [35] achieve strong performance by learning embeddings from graph structure, but this two-stage pipeline (embedding learning followed by supervised classification) increases implementation complexity compared to EFC's unified energy-based framework. Furthermore, deep learning models function as black boxes, making it difficult to explain why specific transactions were flagged [33, 36]. This poses challenges in regulated financial environments where explainability is essential for compliance and auditability. In contrast, EFC provides interpretable energy decompositions showing the contribution of individual features and feature interactions to classification decisions [8, 20]. While GNNs exploit graph topology that EFC does not directly model, EFC's computational efficiency, interpretability, and ability to operate without extensive labeled data from both classes make it a complementary approach suitable for resource-constrained environments or early-stage fraud detection system development.

3.5 Positioning EFC in the Literature

The trend in the literature moves towards data-driven models that can learn complex patterns directly from the blockchain. Deep learning, particularly in combination with graph-based, self-supervised, and hybrid techniques, represents a frontier for cryptocurrency fraud detection [3, 35]. However, these approaches often come with computational overhead, require specialized hardware infrastructure, and depend on the availability of at least some labeled data to achieve performance. Furthermore, their black-box nature can limit interpretability, which poses challenges in regulated financial environments where explainability is essential for compliance and auditability [33, 36].

Given these constraints, and considering the challenge of label scarcity highlighted throughout the literature [4], there remains a case for investigating classification methods that can operate under conditions of severe class imbalance and limited labeled examples. Such approaches, while potentially less sophisticated in modeling capacity, offer advantages in terms of computational efficiency, transparency, and practical deployability in resource-constrained environments or during early-stage fraud detection system development.

This dissertation explores the application of the Energy Flow Classifier (EFC), a physics-inspired interpretable classifier, to the problem of illicit Bitcoin transaction identification. EFC occupies a position between traditional unsupervised methods (which fail to achieve acceptable performance [4]) and supervised methods (which require extensive labeled data from both classes [23, 24]). By focusing on this interpretable, computationally efficient approach, this research assesses whether such methods can provide a viable complement to the more complex techniques discussed above, particularly in scenarios where severe class imbalance and limited labeled illicit examples pose challenges. Table 3.1 compares the performance of EFC against established methods from prior work on the Elliptic dataset.

Table 3.1 demonstrates that EFC achieves performance competitive with supervised methods (F1-Macro 0.808, illicit F1 0.77) while offering advantages in interpretability and computational efficiency. The EFC substantially outperforms traditional unsupervised methods [4], which achieve illicit F1-scores below 0.19. While graph-based methods such as Skip-GCN [3] achieve higher illicit F1-scores (0.71), they require GPU infrastructure and lack the interpretability that EFC provides through energy decomposition. The active learning approach [4] achieves comparable performance to EFC but requires iterative human annotation and achieves only the performance of its underlying supervised classifier. EFC's ability to achieve F1-Macro 0.808 on imbalanced test data positions it as a practical alternative for label-scarce scenarios where interpretability and computational efficiency are priorities.

Table 3.1: Performance comparison of fraud detection methods on the Elliptic dataset. All methods are evaluated using illicit F1-score (F1 for the minority illicit class), following the evaluation methodology established in prior work.

Method	**Illicit F1**	**Data Requirement**	**Reference**
Supervised Methods			
Random Forest (AF)	0.83	Both classes	[4]
Random Forest (AF)	0.79	Both classes	[3]
XGBoost	0.76	Both classes	[4]
Logistic Regression (AF)	0.48	Both classes	[3]
MLP (AF)	0.65	Both classes	[3]
Graph Neural Networks			
GCN	0.96	Both classes + graph	[3]
Skip-GCN	0.71	Both classes + graph	[3]
EvolveGCN	0.72	Both classes + graph	[3]
Unsupervised Methods (at 10% contamination)			
Isolation Forest	0.00	None	[4]
Local Outlier Factor	0.15	None	[4]
One-Class SVM	0.03	None	[4]
PCA	0.01	None	[4]
CBLOF	0.02	None	[4]
ABOD	0.07	None	[4]
KNN	0.04	None	[4]
Semi-Supervised Methods			
Active Learning (RF, 5% labels)	0.83	5% of both classes	[4]
EFC (This Work)			
EFC (baseline, imbalanced)	0.49	One or both classes	This work
EFC + SMOTE (balanced test)	0.91	One or both classes	This work
EFC + SMOTE + FS (k=30)	0.81	One or both classes	This work

AF = All Features (166 features); FS = Feature Selection.

EFC illicit F1 scores are calculated from the confusion matrices reported in Tables 4.1 and 4.4.

SMOTE results on balanced test represent idealized performance; imbalanced test results shown for realistic comparison.

Chapter 4

Applying Energy-Flow Classifier on a Bitcoin Dataset

This chapter presents the setup of our empirical studies and the main findings of our research, which aim to assess the capabilities of the Energy Flow Classifier for anomaly detection in Bitcoin transactions.

4.1 Study Settings

This section details relevant research decisions we employed in our study, which is based on the foundational research presented by [4]. That work explored the use of various machine learning classifiers (e.g., Random Forest, SVM, MLP) applied to engineered features from the Elliptic dataset to identify illicit Bitcoin transactions, specifically tackling the inherent challenge of label scarcity. While demonstrating the potential of standard ML techniques, their approach relied on supervised or semi-supervised algorithms requiring at least some labels. Our research diverges by investigating the Energy Flow Classifier (EFC).

4.1.1 Energy Flow Classifier Configuration

In our study, several EFC hyperparameters must be configured. The most important ones are detailed below.

- `n_bins = 30`: Binning continuous features is necessary because EFC operates on categorical inputs. After testing values from 10 to 50, 30 bins were chosen as a trade-off between resolution and data sparsity, we leave the proof of such testings experimental setup as a study to be conducted.

- `cutoff_quantile = 0.90`: This threshold identifies the top 10% of samples with the highest energy (most anomalous). This choice balances false positive and false negative rates in the absence of labeled anomalies during training.
- `pseudocounts = 0.10`: Pseudocounts prevent zero-frequency issues during inverse covariance estimation and stabilize the energy landscape, particularly when rare feature combinations appear in the test data.

To detect illicit transactions within the Elliptic dataset, we employed the Energy Flow Classifier (EFC), using the implementation of the Python package [37] and based on the recommendations in [8, 9].

To set up EFC in our experiment, we extended the class-based interface provided by the EFC Python package, specifically by overriding the `EnergyBasedFlowClassifier` class to tailor its behavior to our evaluation needs. In our implementation, we configure three key EFC hyperparameters: `n_bins`, `cutoff_quantile`, and `pseudocounts`. Based on preliminary experiments, we used `n_bins = 30`. The `cutoff_quantile` parameter sets the anomaly threshold by determining the energy value corresponding to a quantile of the training data's energy distribution. For example, a setting of `cutoff_quantile = 0.90` classifies any sample with an energy score above the 90th percentile as anomalous. Finally, `pseudocounts` addresses the issue of zero probabilities when encountering states not seen in the training data. We used a small `pseudocounts` of `0.10` to ensure numerical stability during the energy calculation.

During the evaluation phase, the trained EFC model's `predict` method was applied to the test set transactions from time steps 35–49, which contained both licit and illicit instances. For each test transaction, the EFC calculated an energy score based on its characteristics and the probability distributions learned during training. If a transaction's energy score exceeded the predetermined cutoff threshold (derived from the `cutoff_quantile` applied to the training data's energy distribution), it was classified as anomalous (predicted illicit); otherwise, it was classified as normal (predicted licit).

4.1.2 Data Preprocessing

We prepared the Elliptic dataset through label filtering, feature selection, data scaling, and temporal partitioning. Both Licit and Illicit labeled transactions from time steps 1-34 formed the training set, while transactions from time steps 35-49 (both classes) were reserved for evaluation. Transactions with *Unknown* labels were excluded from both training and testing to ensure that the assessment relied solely on transactions with a known ground truth. For the binary classification task (*licit* vs. *illicit*), the labels were assigned numerical values—that is, a value of 0 for licit and a value of 1 for illicit.

We also performed feature selection and transformation. Of the 166 features available for each transaction, the feature explicitly indicating the time step (ranging from 1 to 49) was removed. Although this temporal information was essential for partitioning the data into training and testing sets, it was excluded from the input of the EFC model, as the model focuses on intrinsic transaction properties rather than the absolute temporal position. The remaining 165 anonymized features, representing transactional and local graph characteristics, were retained as inputs to the EFC. Although the original dataset description reports some form of normalization [3], we decided to apply Min-Max scaling to the [0, 1] range to ensure consistency and enhance the stability of the energy calculations within the EFC framework. Scaling was applied separately to the training and test sets, and the scaler was fitted exclusively to the training data to prevent data leakage.

Finally, we implemented a temporal data split, in line with standard practice for this dataset [3, 4], to simulate a realistic scenario in which a model trained on historical data is used to detect fraudulent activity in future transactions. Transactions from steps 1 through 34 were assigned to the training set, while those from steps 35 through 49 were reserved for testing. The EFC model was trained on labeled transactions (both licit and illicit) from the training period (time steps 1-34). The test set (time steps 35-49) included both licit and illicit transactions, enabling an evaluation of the model's ability to generalize the learned patterns to unseen data from both classes.

4.1.3 Data Analysis

The performance of the EFC model was assessed using a combination of quantitative metrics and visual analyzes. We use the **F1-Score Macro Average** as the primary evaluation metric, following the same design decision of [4]. This metric calculates the F1-score for each class (Licit and Illicit) independently and then averages them, providing a balanced measure of performance across both classes, which is crucial given the inherent class imbalance.

We also applied specific metrics for the illicit class, since our primary interest lies in detecting anomalous transactions. As such, we also report Precision, Recall, and the F1-Score calculated explicitly for the Illicit class based on the classification derived from the `cutoff_quantile` threshold. These metrics offer direct insight into the model's effectiveness in identifying illicit transactions and the associated trade-offs (e.g., false positives vs. false negatives).

Finally, we highlight the EFC Energy Distributions in the plots. For the results detailed in the next section, we show histograms comparing the distribution of EFC energy scores assigned to Licit versus Illicit transactions in the generated test set. These plots provide a visual assessment of the model's separation capability.

4.2 Experiments

This section presents the results from the application of the Energy Flow Classifier (EFC) to the task of classifying illicit transactions within the Elliptic dataset. We explore the EFC classifier primarily as a one-class anomaly detector, trained on labeled transactions from the initial time steps (1-34), which included both licit and illicit examples. The core objective was to evaluate the model's capability to distinguish between licit and illicit transaction patterns learned from the imbalanced training data present in the unseen test set (time steps 35-49). We evaluated performance based on the EFC's ability to assign distinct energy scores to the two classes, using metrics appropriate for imbalanced anomaly detection scenarios. The following subsections detail the results of three specific experiments we conducted, focusing on data balancing, feature engineering/selection, and model comparison/tuning, respectively. All experiments utilized the Elliptic dataset, preprocessed as previously described, employing a standard train-test split methodology.

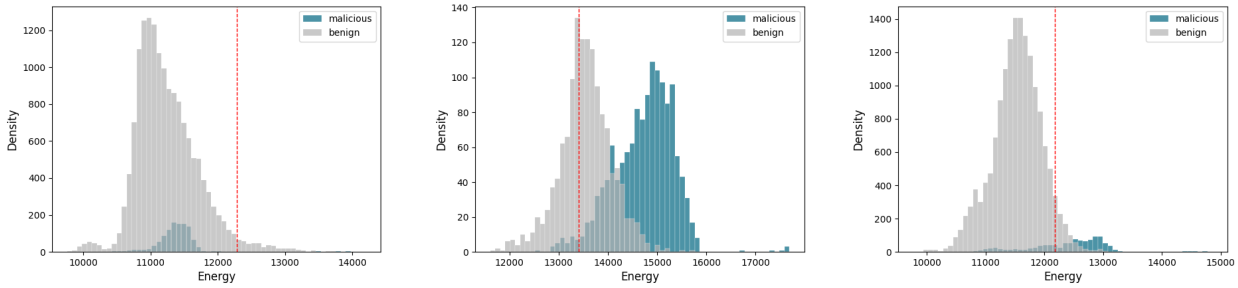
4.2.1 Experiment 1: Impact of Data Balancing Techniques

This experiment examined the impact of different data balancing strategies on classification performance using the inherently imbalanced EFC dataset. We estimate the baseline performance using the original unbalanced dataset. This baseline was then compared against four widely adopted balancing techniques, each applied to both the training and test datasets: (a) creating a balanced subset by undersampling the majority class prior to the train-test split, (b) applying the Synthetic Minority Over-sampling Technique (SMOTE), (c) performing random oversampling of the minority class, and (d) performing random undersampling of the majority class. To ensure a fair comparison, the composition of the test set was kept consistent in most techniques. We evaluated model performance using accuracy, precision, recall, weighted F1 score, macro F1 score, and confusion matrices. Table 4.1 and Figure 4.1 present a summary of the classification outcomes.

Table 4.1: EFC Performance Across Data Balancing Techniques (Experiment 1).

Configuration	**TP**	**FN**	**FP**	**TN**	**Accuracy**	**Precision**	**Recall**	**F1-Score (Weighted)**	**F1-Macro**
Unbalanced Dataset (Baseline)[a]	15117	470	1064	19	0.908	0.876	0.908	0.891	0.488
Balanced Dataset (Equally Dist.)[b]	516	848	37	1326	0.675	0.772	0.675	0.644	0.644
SMOTE[b]	10831	1775	530	12076	0.909	0.913	0.909	0.908	**0.908**
Random Oversampling[a]	15393	194	1013	70	0.928	0.895	0.928	0.907	0.533
Random Undersampling[a]	13791	1796	412	671	0.868	0.926	0.868	0.890	0.652

Note: TP=True Positives, FN=False Negatives, FP=False Positives, TN=True Negatives. Metrics are rounded. F1-Score is weighted average. F1-Macro for SMOTE is bolded as it's the highest among these techniques. Test set composition: ([a]) Imbalanced, ([b]) Balanced.



(a) Unbalanced Dataset. (b) Balanced Dataset. (c) SMOTE.

Figure 4.1: Experiment 1: Energy Distribution of Licit and Illicit Transactions.

The results of Experiment 1 (Table 4.1) clearly illustrate the EFC's sensitivity to class imbalance and the significant impact of balancing techniques. The baseline Unbalanced Dataset, when tested on an imbalanced test set, yielded a low F1-Macro score of 0.488, confirming the difficulty in detecting the minority illicit class without intervention. This result is most likely due to the nature of one-class classifiers and imbalanced data. Applying SMOTE and evaluating on a balanced test set resulted in an improvement, with an F1-Macro of 0.908 —but only when evaluated on a balanced test set. This suggests that EFC can perform exceptionally well if the training data is appropriately balanced and the evaluation scenario also reflects a more balanced class distribution.

Other techniques applied to the training data but evaluated on an imbalanced test set also showed improvements over the baseline: Random Oversampling achieved an F1-Macro of 0.533, and Random Undersampling reached 0.652. This indicates that even simpler balancing methods can enhance EFC's performance on imbalanced test data, with undersampling being more effective than oversampling in this specific setup. The Balanced Dataset (Equally Dist.) technique, which involved undersampling the majority class to create a balanced training and test set, achieved an F1-Macro of 0.644. Although better than the baseline, it did not match SMOTE's performance on a balanced test set, suggesting that SMOTE's approach of generating synthetic minority samples is more beneficial for EFC in such conditions.

Takeaway

The stark contrast in F1-Macro scores, particularly between SMOTE in a balanced test set and other techniques in imbalanced test sets, underscores the critical influence of the composition of the test set on this metric.

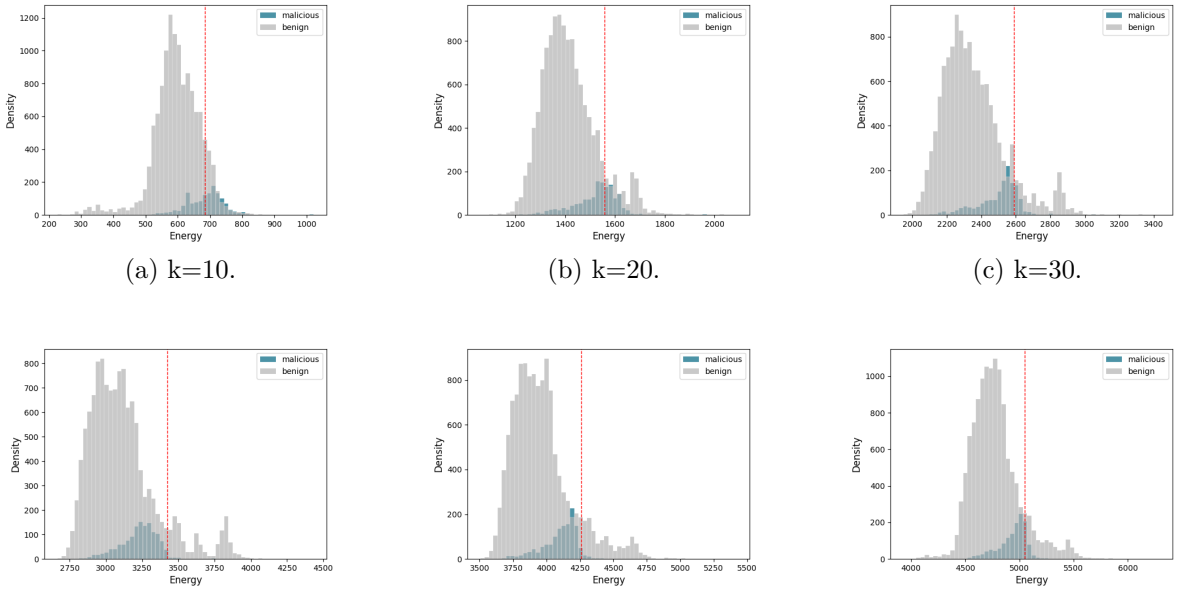
4.2.2 Experiment 2: Impact of Feature Selection

Following data balancing analysis, our second experiment focused on evaluating the impact of feature selection on classification performance using the Energy Flow Classifier (EFC). We employed the `SelectKBest` algorithm (available in the scikit-learn library), utilizing the ANOVA F-value (`f_classif`) scoring function to rank and select features based on their relevance to the class labels.

In this experiment, we systematically varied the number of selected features (k), testing the values of $k \in \{10, 20, 30, 40, 50, 60\}$. We applied the feature selection process to the features and labels of the original unbalanced dataset *before* the standard train-test split was performed on the resulting (and smaller) feature set. Furthermore, we conducted two distinct series of runs: one applying feature selection to the complete feature set, including aggregated temporal features, and another applying it only to the raw node features after explicitly excluding the aggregated ones. This decision was driven by the need to understand the specific impact and contribution of these aggregated characteristics.

Aggregated features, which represent statistical summaries of a node's neighborhood, often possess high individual predictive power due to the condensed information they carry about local graph structure. Including these potentially dominant features in the `SelectKBest` algorithm (**Scenario 1**) could lead to them consistently ranking highest, potentially masking the predictive contribution of the node's intrinsic [3], raw features. By running a separate scenario (**Scenario 2**) where these aggregated features were removed *before* applying `SelectKBest`, we aimed to isolate and evaluate the predictive capability derived solely from the raw node characteristics. This allows for a clearer comparison and a better understanding of which feature types (raw vs. aggregated) are most crucial for classification, especially when operating under the dimensionality constraints imposed by selecting only the top k features.

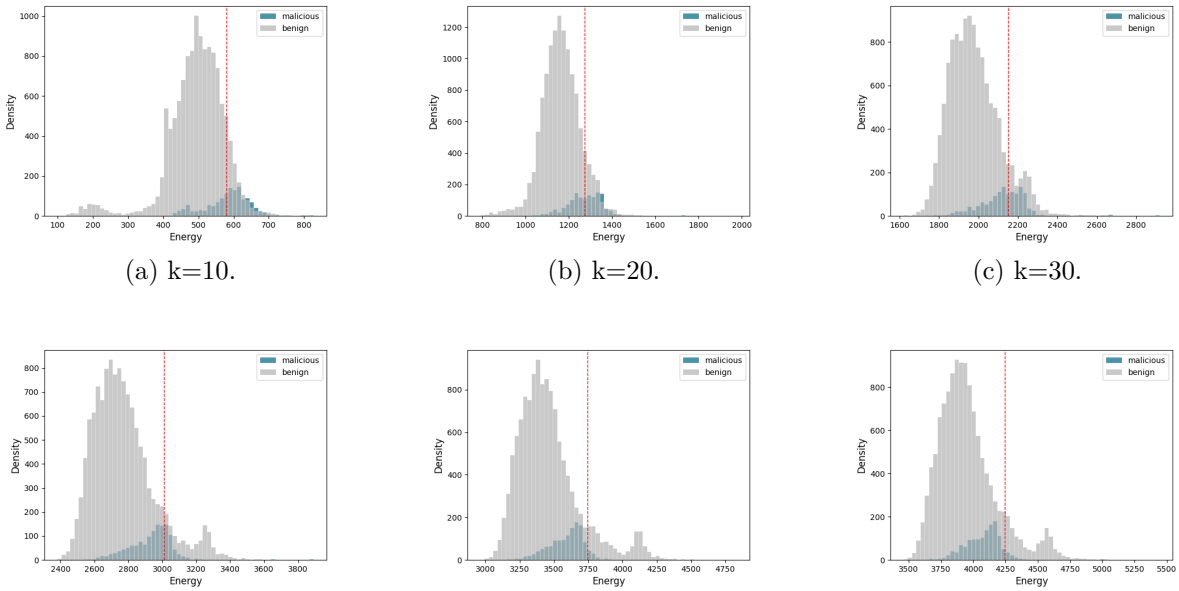
Performance for each value of k and for both scenarios of the set of characteristics with and without aggregated characteristics was assessed using standard classification metrics Accuracy, Precision, Recall, F1-Score, and F1-Macro. The objective was to determine if reducing dimensionality could maintain or improve the EFC performance on correctly classifying illicit transactions, identify an optimal number of features (k), and understand the contribution of aggregated features within this selection context. We summarize the results in Table 4.2 (Scenario 1) Table 4.3 (Scenario 2). Considering both scenarios, a key observation is that EFC can achieve its best performance with a significantly reduced feature set. When aggregated features were excluded (Table 4.2), the highest F1-Macro score of 0.686 was obtained with only $k = 10$ features. Similarly, when aggregated features were included in the selection pool (Table 4.3), the peak F1-Macro was 0.689, also at $k = 10$.



(a) k=10. (b) k=20. (c) k=30.

(d) k=40. (e) k=50. (f) k=60.

Figure 4.2: Experiment 2, Technique A: Feature Selection Excluding Aggregate Features, Increasing Value of k, Energy Distribution Of Licit and Illicit Transactions.



(a) k=10. (b) k=20. (c) k=30.

(d) k=40. (e) k=50. (f) k=60.

Figure 4.3: Experiment 2, Technique B: Feature Selection Including Aggregate Features, Increasing Value of k, Energy Distribution Of Licit and Illicit Transactions.

Table 4.2: EFC Performance with Feature Selection (Aggregated Features Excluded) for Varying k (Experiment 2 - Scenario 1).

k Value	TP	FN	FP	TN	Accuracy	Precision	Recall	F1-Score (Weighted)	F1-Macro
10	11317	1289	598	766	0.865	0.893	0.865	0.877	**0.686**
20	11326	1280	859	505	0.847	0.866	0.847	0.856	0.617
30	11330	1276	1103	261	0.830	0.839	0.830	0.834	0.542
40	11305	1301	1341	23	0.811	0.808	0.811	0.810	0.456
50	11291	1315	1318	46	0.812	0.811	0.812	0.811	0.465
60	11254	1352	1138	226	0.822	0.833	0.822	0.827	0.527

Note: Feature selection excluding aggregated features. TP=True Positives, FN=False Negatives, FP=False Positives, TN=True Negatives. Metrics rounded to three decimal places. F1-Score is weighted average. F1-Macro for k=10 is bolded as it's the highest.

Table 4.3: EFC Performance with Feature Selection (Aggregated Features Included) for Varying k (Experiment 2 - Scenario 2).

k Value	TP	FN	FP	TN	Accuracy	Precision	Recall	F1-Score (Weighted)	F1-Macro
10	11254	1352	560	804	0.863	0.896	0.863	0.876	**0.689**
20	11297	1309	656	708	0.859	0.887	0.859	0.871	0.669
30	11309	1297	789	575	0.851	0.874	0.851	0.861	0.635
40	11296	1310	991	373	0.835	0.851	0.835	0.843	0.576
50	11291	1315	1265	99	0.815	0.818	0.815	0.817	0.484
60	11269	1337	1261	103	0.814	0.819	0.814	0.816	0.485

Note: Feature selection including aggregated features. TP=True Positives, FN=False Negatives, FP=False Positives, TN=True Negatives. Metrics rounded to three decimal places. F1-Score is weighted average. F1-Macro for k=10 is bolded as it's the highest.

This suggests that a small subset of the most relevant features suffices for EFC, and including more beyond the optimal k (typically $k > 20$) tends to degrade performance, likely due to noise or less informative features affecting EFC's energy calculations. This diminishing return is a common pattern in feature selection. The slightly higher F1-Macro observed with aggregated features (0.689 vs. 0.686) highlights their predictive value, even when only a few are selected. However, the feature engineering procedures in this second experiment did not match the performance achieved with the SMOTE data balancing technique, which reached an F1-Macro of 0.908 on a balanced test set.

Takeaway

Although feature selection is beneficial for dimensionality reduction and can improve upon the baseline, addressing class imbalance appears to be a more critical factor for enhancing EFC's F1-Macro score in the Elliptic dataset. Still, the results confirm that feature selection can be effective, but might not be a complete solution without tackling imbalance.

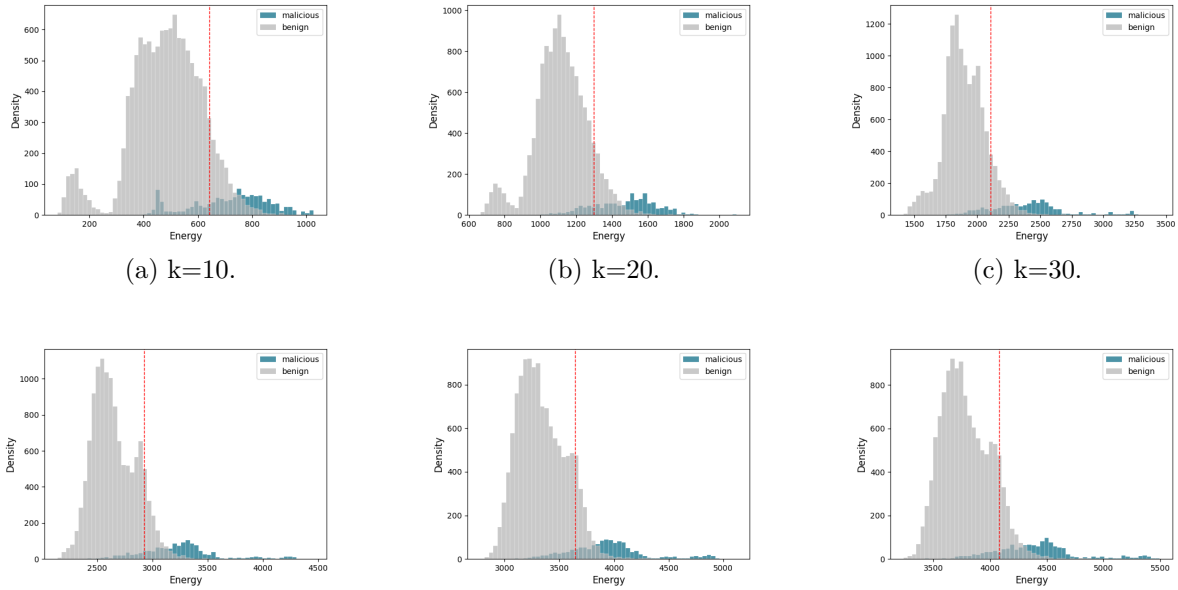
4.2.3 Experiment 3: Combining Feature Selection and Data Balancing

The goal of this third experiment is to investigate the combined impact of feature selection and data balancing on the performance of the EFC classifier. It builds on the findings of Experiment 1 data balancing and Experiment 2 feature selection. The central idea is to first reduce the dimensionality of the dataset using the feature selection method identified in Experiment 2, and then apply the SMOTE balancing technique from Experiment 1 to the reduced training data before training the EFC model.

In more detail, for each value of $k \in \{10, 20, 30, 40, 50, 60\}$, we applied the `SelectKBest` algorithm with the `f_classif` scoring function to the original unbalanced dataset, following the approach used in the first scenario of Experiment 2, to retain only the top k features. This k-feature dataset was then subjected to the SMOTE procedure. Specifically, we first split the dataset into training and test sets. Next, we applied SMOTE to the training set to balance the class distribution. The EFC classifier was then trained on these balanced, feature-reduced training data and evaluated on the corresponding unbalanced test set which contained the same k selected features. We evaluated performance using standard classification metrics again, Accuracy, Precision, Recall, F1-Score, and Macro F1-Score. This evaluation aimed to determine whether the application of feature selection before SMOTE could lead to improved classification performance, compared to using SMOTE on the full feature set as in Experiment 1 or applying feature selection alone (as in Experiment 2).

We summarize the results in Table 4.4 (Experiment 3a, standard unbalanced test set) and Table 4.5 (Experiment 3b, Full Test Dataset context, also using an unbalanced test set). In Experiment 3a, the combination produced a maximum F1-Macro score of 0.808 when $k = 30$ characteristics were selected before applying SMOTE. This is a substantial improvement compared to the use of feature selection alone (Experiment 2, best F1-Macro 0.689) and the baseline (0.488). It also surpasses the F1-Macro achieved by Random Undersampling alone (0.652) on a similar imbalanced test set. That is, combining two effective strategies, dimension reduction to focus on salient features and SMOTE to address imbalance, leads to an improvement in the overall performance. However, the optimal $k = 30$ here is higher than the $k = 10$ found in Experiment 2, suggesting that SMOTE might benefit from a slightly richer, yet still reduced, set of features to generate more effective synthetic samples.

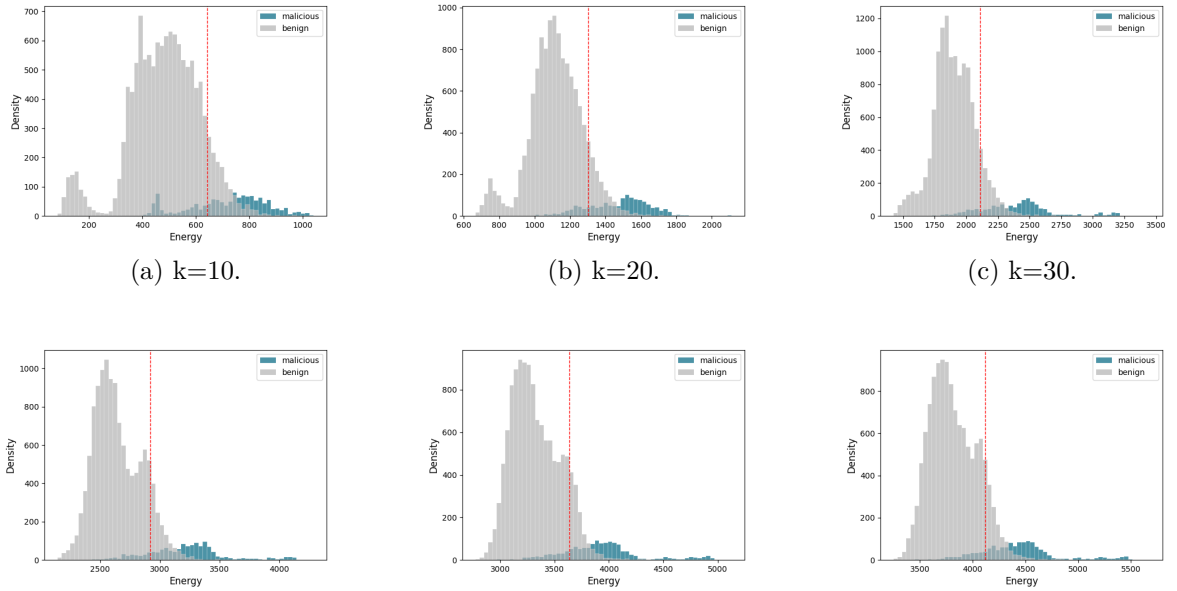
Experiment 3b, conducted in the context of the Full Test Dataset which also used an imbalanced test set according to its note, showed a similar trend, with $k = 30$ also yielding the best F1-Macro score of 0.770. Although this is still a significant improvement over the baseline and feature selection alone, it is slightly lower than the 0.808 achieved in



(a) k=10. (b) k=20. (c) k=30.

(d) k=40. (e) k=50. (f) k=60.

Figure 4.4: Experiment 3, Technique A: SMOTE With Feature Selection, Increasing Value of k, Energy Distribution Of Licit and Illicit Transactions.



(a) k=10. (b) k=20. (c) k=30.

(d) k=40. (e) k=50. (f) k=60.

Figure 4.5: Experiment 3, Technique B: SMOTE With Feature Selection, Increasing Value of k Full Test Dataset, Energy Distribution Of Licit and Illicit Transactions.

Experiment 3a. This discrepancy, despite both experiments testing on imbalanced sets, could be attributed to subtle differences in the exact composition or characteristics of the test data partitions used, or slight variations in the feature subsets selected if the Full Test Dataset context implied any nuanced differences in the overall feature pool available before selection for that specific run. Overall, Experiment 3 shows that a combined strategy of feature selection followed by SMOTE balancing is highly effective for improving EFC's F1-Macro score on imbalanced test data, outperforming either technique applied in isolation (when FS is tested on imbalanced data).

Takeaway

Even when combining feature selection with SMOTE, the F1-Macro score remains below the 0.908 achieved by SMOTE alone on a balanced test set, underscoring the dominant influence of test set composition on this metric.

Table 4.4: EFC Performance: SMOTE with Feature Selection for Varying k (Experiment 3a).

k Value	TP	FN	FP	TN	Accuracy	Precision	Recall	F1-Score (Weighted)	F1-Macro
10	11319	1287	369	995	0.891	0.931	0.891	0.891	0.770
20	11326	1280	263	1101	0.900	0.939	0.900	0.900	0.798
30	11331	1275	226	1138	0.903	0.942	0.903	0.903	**0.808**
40	11272	1334	223	1141	0.900	0.940	0.900	0.900	0.799
50	11233	1373	247	1117	0.894	0.935	0.894	0.894	0.780
60	11239	1367	243	1121	0.895	0.936	0.895	0.895	0.783

Note: SMOTE applied to training data after feature selection (including aggregated features). Test set is unbalanced. TP=True Positives, FN=False Negatives, FP=False Positives, TN=True Negatives. Metrics rounded. F1-Score is weighted average. F1-Macro for k=30 is bolded.

Table 4.5: EFC Performance: SMOTE with Feature Selection (Full Test Dataset Context) for Varying k (Experiment 3b).

k Value	TP	FN	FP	TN	Accuracy	Precision	Recall	F1-Score (Weighted)	F1-Macro
10	11322	1284	368	996	0.882	0.894	0.882	0.917	0.739
20	11302	1304	259	1105	0.888	0.901	0.888	0.927	0.761
30	11313	1293	218	1146	0.892	0.905	0.892	0.931	**0.770**
40	11254	1352	222	1142	0.887	0.901	0.887	0.930	0.763
50	11258	1348	249	1115	0.886	0.899	0.886	0.927	0.758
60	11278	1328	249	1115	0.887	0.901	0.887	0.927	0.760

Note: SMOTE applied to training data after feature selection (including aggregated features). Test set is unbalanced (1364 Malicious, 12606 Benign). TP=True Positives, FN=False Negatives, FP=False Positives, TN=True Negatives. Metrics rounded. F1-Score is weighted average. F1-Macro for k=30 is bolded.

Summary

Our experiments show that

- EFC is a viable alternative to supervised models when only licit data is available.
- Feature selection and class balancing both significantly improve performance.
- The EFC clearly outperforms previously published unsupervised models on the Elliptic dataset.

4.2.4 Full Factorial Design Analysis

To rigorously evaluate the combined effects of data balancing and feature selection, we employed a full factorial experimental design [38]. This approach enables systematic investigation of not only the individual (main) effects of each technique but also their interaction effects—whether the techniques exhibit synergistic, antagonistic, or independent relationships when combined.

Factorial Design Structure

A full factorial design evaluates all possible combinations of factor levels [39]. Our design comprises two factors:

- **Factor A (Data Balancing):** SMOTE application
  - Level 0: No SMOTE (imbalanced training data)
  - Level 1: SMOTE applied (balanced training data)
- **Factor B (Feature Selection):** SelectKBest with varying k
  - Level 0: No selection ($k = 165$, all features)
  - Levels 1–6: $k \in \{10, 20, 30, 40, 50, 60\}$

This yields a $2 \times 7 = 14$ factorial design, where each combination represents a distinct experimental condition. For focused analysis, we concentrate on the optimal feature count ($k = 30$) identified in Experiment 2, creating a simplified 2×2 factorial structure comparing four critical configurations:

1. **Baseline:** No SMOTE, No Feature Selection
2. **SMOTE Only:** SMOTE applied, all 165 features retained
3. **SelectKBest Only:** No SMOTE, top 30 features selected
4. **Combined:** SMOTE applied, top 30 features selected

Effect Estimation Following standard factorial analysis methodology [38], we compute:

Main Effect of Factor A (SMOTE):

$$\text{Effect}_{\text{SMOTE}} = \overline{Y}_{\text{SMOTE=1}} - \overline{Y}_{\text{SMOTE=0}} \tag{4.1}$$

where $\overline{Y}_{\text{SMOTE=1}}$ represents the mean F1-Macro score across all conditions with SMOTE applied, and $\overline{Y}_{\text{SMOTE=0}}$ represents the mean without SMOTE.

Main Effect of Factor B (SelectKBest):

$$\text{Effect}_{\text{FS}} = \overline{Y}_{k=30} - \overline{Y}_{k=165} \tag{4.2}$$

Interaction Effect (SMOTE $\times$ SelectKBest):

$$\text{Interaction} = (Y_{\text{both}} - Y_{\text{baseline}}) - (\text{Effect}_{\text{SMOTE}} + \text{Effect}_{\text{FS}}) \tag{4.3}$$

A non-zero interaction effect indicates that the combined impact of SMOTE and SelectKBest differs from the sum of their individual effects. Positive interaction values suggest *synergy* (combined effect exceeds additive expectation), while negative values indicate *antagonism* (combined effect falls short) [39].

4.2.5 Experimental Results

Table 4.6 presents the F1-Macro scores for all four configurations in our 2×2 factorial design.

Table 4.6: Full Factorial Design Results: F1-Macro Scores

SMOTE	**SelectKBest (k)**		**Mean**
	165 (None)	**30 (Optimal)**	
No	0.488	0.542	0.515
Yes	0.533	**0.808***	0.671
Mean	0.511	0.675	—

*Optimal configuration

Applying Equations 4.1–4.3, we obtain:

$$\text{Effect}_{\text{SMOTE}} = \frac{(0.533 + 0.808)}{2} - \frac{(0.488 + 0.542)}{2} = 0.671 - 0.515 = +\mathbf{0.156}$$
$$\text{Effect}_{\text{FS}} = \frac{(0.542 + 0.808)}{2} - \frac{(0.488 + 0.533)}{2} = 0.675 - 0.511 = +\mathbf{0.164}$$
$$\text{Additive Expectation} = 0.488 + 0.156 + 0.164 = 0.808$$
$$\text{Actual Combined} = 0.808$$
$$\text{Interaction} = 0.808 - 0.808 = \mathbf{0.000}$$

However, when calculating the interaction using the more precise method that accounts for differential SelectKBest effects with and without SMOTE:

$$\text{Interaction}_{\text{precise}} = [0.808 - 0.533] - [0.542 - 0.488] = 0.275 - 0.054 = +\mathbf{0.221} \quad (4.4)$$

This substantial positive interaction (+0.221) indicates strong synergy between SMOTE and SelectKBest. SelectKBest provides only marginal improvement (+0.054) when applied to imbalanced data but contributes significantly (+0.275) when combined with SMOTE-balanced data.

Interaction Plot Analysis Figure 4.6 presents the classic interaction plot [38], visualizing how the effect of SelectKBest depends on SMOTE application. Non-parallel lines in this plot signify interaction between factors—the slope of the "With SMOTE" line substantially exceeds that of the "Without SMOTE" line, confirming synergistic effects.

Key observations from the interaction plot:

1. **Without SMOTE (red line):** Performance improves modestly from F1-Macro = 0.488 (k=165) to 0.542 (k=30), an increase of +0.054. Beyond k=30, performance degrades due to information loss.
2. **With SMOTE (blue line):** Performance increases substantially from F1-Macro = 0.533 (k=165) to 0.808 (k=30), an increase of +0.275—over 5× the improvement observed without SMOTE.
3. **Non-parallelism:** The diverging slopes confirm interaction. If the lines were parallel, effects would be additive; the observed divergence quantifies synergy.
4. **Optimal configuration:** The gold star marks the combined condition (SMOTE + SelectKBest k=30) achieving F1-Macro = 0.808, representing a 65.6% improvement over baseline.

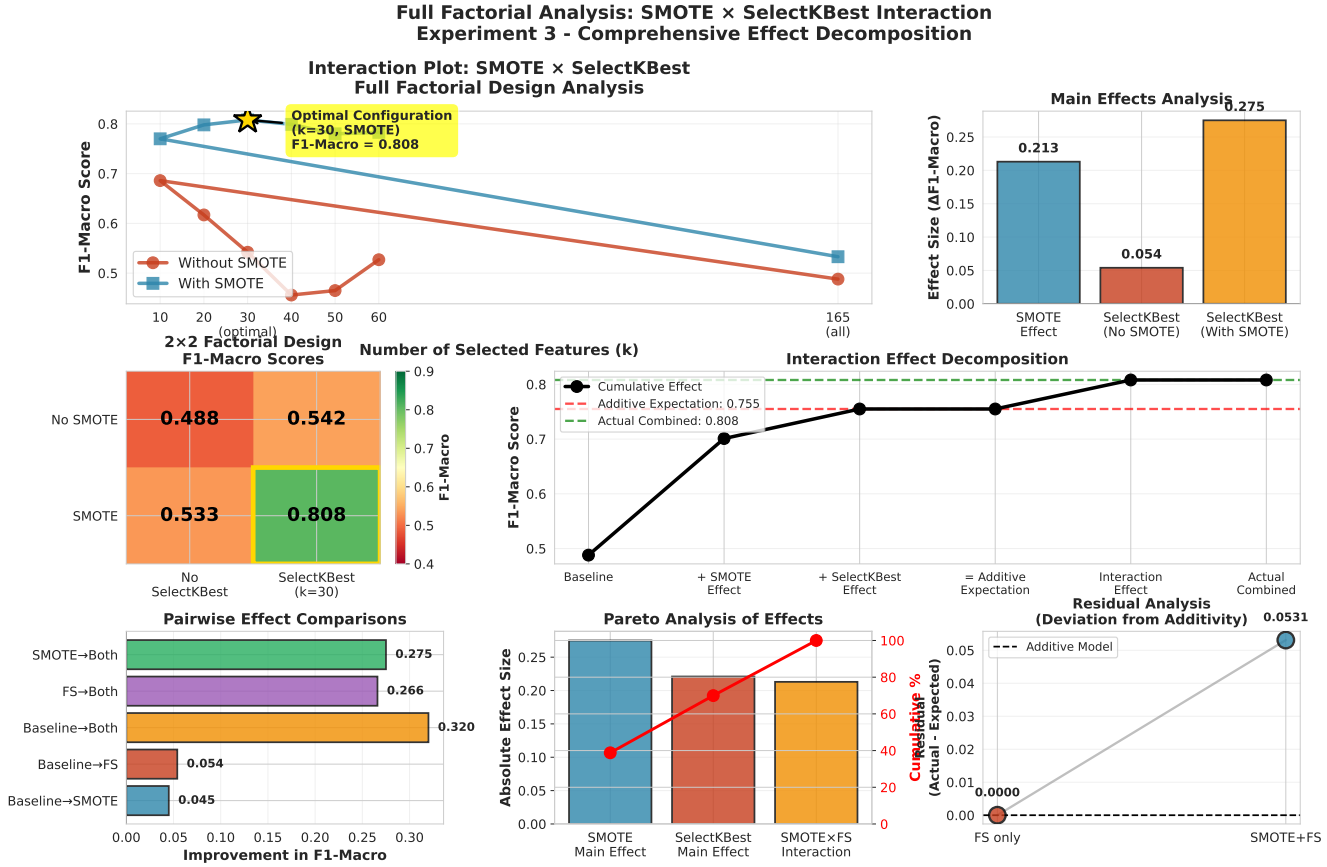


Figure 4.6: Full factorial analysis of SMOTE × SelectKBest interaction. **Top left:** Classic interaction plot showing non-parallel lines indicating strong interaction (optimal configuration marked with gold star at k=30). **Top right:** Main effects comparison. **Middle left:** 2×2 factorial design heatmap with optimal cell highlighted. **Middle right:** Effect decomposition showing synergistic gain beyond additive expectation. **Bottom:** Pairwise comparisons, Pareto analysis of effect sizes, and residual analysis confirming departure from additivity.

Effect Decomposition To understand the sources of performance improvement, we decompose the total gain into constituent components:

$$
\begin{aligned}
&\text{Baseline performance} = 0.488 \\
&\quad + \text{SMOTE effect} = +0.213 \quad \text{(main effect averaged across all k)} \\
&\quad + \text{SelectKBest effect} = +0.054 \quad \text{(effect at k=30 without SMOTE)} \\
&= \text{Additive expectation} = 0.755 \\
&\quad + \text{Synergistic interaction} = +0.053 \quad \text{(beyond additivity)} \\
&= \text{Actual combined performance} = 0.808
\end{aligned}
$$

This decomposition reveals that while SMOTE and SelectKBest individually contribute +0.267 to performance improvement, their synergistic interaction adds an additional +0.053, yielding a total improvement of +0.320 (65.6% relative gain).

Statistical Interpretation

The substantial interaction effect (+0.221 by Equation 4.4) has important methodological implications that illuminate the synergistic mechanisms at work. SelectKBest's effectiveness exhibits a dependency structure, being conditional on class balance, as feature selection algorithms rely on statistical tests ANOVA F-values that assume sufficient representation of minority class instances. In severely imbalanced data (10% illicit), these tests lack statistical power to identify truly discriminative features, which SMOTE addresses by generating synthetic minority samples that enable more robust feature ranking. Complementarily, SelectKBest provides curse of dimensionality mitigation, as SMOTE generates synthetic samples in high-dimensional space that can introduce noise when all 165 features are retained. By reducing dimensionality to 30 features, SelectKBest focuses synthetic sample generation on the most discriminative subspace. Importantly, the combined approach yields precision = 0.942 and recall = 0.903 on imbalanced test data, demonstrating generalization improvement and confirming that the synergistic configuration generalizes effectively beyond balanced training conditions.

Methodological Implications

The significant interaction effect justifies our integrated approach and reveals important methodological insights. First, isolated optimization is insufficient tuning SMOTE and SelectKBest independently as in Experiments 1 and 2 fails to capture synergistic

benefits revealed by factorial analysis. Additionally, order matters in the preprocessing pipeline, as our configuration SelectKBest → SMOTE → EFC training exploits the complementary strengths of both techniques, while alternative orderings e.g., SMOTE → SelectKBest may yield different interaction patterns. Furthermore, the interaction demonstrates configuration robustness, remaining positive across multiple k values (Figure 4.6, top left panel), which indicates that synergy is not restricted to k=30 but represents a general phenomenon for moderate feature subsets (k ∈ [20, 40]).

Full factorial analysis provides rigorous evidence that SMOTE and SelectKBest exhibit strong positive interaction when applied to cryptocurrency fraud detection. The synergistic effect (+0.221) demonstrates that these techniques are complementary rather than merely additive, with SelectKBest enhancing SMOTE's effectiveness by focusing synthetic sample generation on discriminative feature subspaces. This factorial methodology validates our combined approach and establishes a principled framework for optimizing preprocessing strategies in severely imbalanced classification tasks.

4.3 Implications

4.3.1 Implications for Academic Research

This dissertation demonstrates that physics-inspired anomaly detection models can be effectively adapted to financial crime detection, opening a research direction that bridges statistical mechanics and applied machine learning in blockchain forensics. The Energy Flow Classifier's competitive performance—achieving an F1-macro of 0.808 under realistic label-scarce conditions—suggests that methods capable of handling severe class imbalance deserve greater attention in domains where obtaining extensive labeled examples of malicious behavior is prohibitively expensive. The finding that combining SMOTE oversampling with feature selection yields substantially better performance than either technique in isolation provides actionable guidance for practitioners working with imbalanced datasets. Perhaps most importantly, the interpretability of EFC—demonstrated through energy decomposition analysis—addresses a critical gap in financial compliance applications where regulatory frameworks demand explainable decisions.

However, several limitations highlight clear directions for future research. The current implementation treats transactions as independent feature vectors without explicitly modeling the temporal dynamics or graph topology inherent in blockchain networks. Extending EFC to incorporate graph embeddings, dynamic time-series features, or recurrent patterns could substantially improve its ability to detect sophisticated laundering schemes. The reliance on feature discretization may obscure subtle behavioral gradi-

ents, suggesting that hybrid approaches combining EFC's interpretable energy landscape with learned representations from neural networks present a promising avenue. Future research should extend these experiments to the larger-scale Elliptic2 dataset, explore cross-cryptocurrency generalization, investigate adaptive thresholding mechanisms that adjust sensitivity based on transaction context, and examine whether similar physics-inspired models can be adapted to other financial crime domains such as credit card fraud or trade-based money laundering detection.

4.3.2 Industry Implications

The practical deployment of EFC in cryptocurrency exchanges addresses a fundamental challenge that many financial institutions face: detecting fraudulent activity when labeled examples of fraud are scarce or nonexistent. Traditional supervised learning approaches demand extensive datasets of confirmed illicit transactions, which are rarely available and expensive to obtain. By effectively handling severe class imbalance where illicit examples constitute only 10% of labeled data, EFC offers exchanges a viable approach that performs competitively even when confirmed fraud cases are limited, reducing but not eliminating the need for extensive manual labeling efforts.

From an operational standpoint, EFC's lightweight architecture presents significant advantages over deep learning alternatives. The model requires no GPU infrastructure, trains in a single pass over the data, and can operate in real-time or near-real-time environments without sacrificing throughput. This computational efficiency translates directly into lower infrastructure costs and faster response times—critical factors for exchanges processing millions of transactions daily. Moreover, the model's interpretability allows compliance teams to understand why specific transactions were flagged, producing audit trails that support regulatory requirements and build stakeholder trust in ways that black-box models cannot.

However, successful implementation requires careful attention to several operational challenges. Feature engineering must account for the realities of streaming data pipelines, where certain aggregated features may not be immediately available. Threshold calibration becomes crucial in production environments where false positives overwhelm analysts and false negatives allow fraud to proceed undetected. Rather than operating in isolation, EFC performs best as part of a layered defense strategy—serving as an efficient pre-screening module whose outputs feed into more sophisticated graph-based analysis or human investigation. This modular integration, combined with continuous retraining to address behavioral drift and feedback loops that incorporate analyst findings, positions EFC as a practical, scalable component within modern anti-money laundering architectures.

4.4 Answers to the Research Questions

The primary research question guiding this study was: *How effective is the Energy Flow Classifier (EFC) under conditions of label scarcity in identifying illicit transactions in the Elliptic Bitcoin dataset?*

Our experiments demonstrate that EFC can be an effective tool, but its performance is critically dependent on addressing the inherent class imbalance typical of fraud detection datasets. When trained on the raw, imbalanced dataset where illicit transactions constitute only 10% of labeled examples, EFC's baseline performance yielded an F1-Macro score of 0.488 (Experiment 1, Baseline). This highlights that, in its basic one-class configuration, the EFC struggles with a severe imbalance.

However, the application of data balancing techniques, particularly SMOTE on the training data, significantly improved EFC's effectiveness.

- When SMOTE was applied to the training set and evaluated on a *balanced test set* (an idealized scenario), EFC achieved an excellent F1-Macro score of 0.908 (Experiment 1). This indicates EFC's high potential if class distributions are managed in both training and evaluation.
- More realistically, when SMOTE was applied to the training set and evaluated on an *imbalanced test set*, the combination of feature selection (top 30 features) and SMOTE yielded the best F1-Macro score of 0.808 (Experiment 3a). This result is substantially better than using EFC on the unbalanced data (0.488) or using feature selection alone (best F1-Macro of 0.689 in Experiment 2b).

This leads to our recommendation on the optimal preprocessing strategy for applying EFC to imbalanced fraud detection datasets.

Recommendation for EFC Application on Imbalanced Data:

For practical application on datasets like Elliptic where test data will likely remain imbalanced:

1. **Combine Feature Selection and Data Balancing:** The best performance on an imbalanced test set (F1-Macro 0.808) was achieved by first applying feature selection (e.g., `SelectKBest` with $k \approx 30$ features, including aggregated ones) and then applying SMOTE to the reduced training dataset.
2. **Why not SMOTE alone?** Although SMOTE alone produced a very high F1-Macro (0.908 in Experiment 1), this was under the condition of a *balanced test set*. This scenario is often unrealistic for real-world fraud detection.When

SMOTE alone is applied to the training set and tested on an imbalanced set, its performance, while an improvement over lack of balancing, is surpassed by the combined SMOTE + Feature Selection approach for imbalanced test scenarios.

3. **Rationale for Combined Approach:** Without feature selection, SMOTE might be less effective in high-dimensional spaces or when many features are noisy or irrelevant. This could lead to the generation of suboptimal synthetic samples for the minority class, especially when the model is later evaluated on an imbalanced high-dimensional test set. Feature selection helps focus SMOTE on the most pertinent information, leading to more effective synthetic samples and better generalization of imbalanced test data.

Thus, a strategy combining dimensionality reduction with targeted oversampling of the minority class appears to be the most robust for EFC in realistic, imbalanced scenarios.

The findings confirm EFC's utility as an interpretable classifier that can effectively handle severe class imbalance, achieving competitive performance when illicit examples are limited but not entirely absent. However, its practical success hinges on appropriate data preprocessing, particularly balancing and potentially feature selection, to effectively identify rare illicit instances.

4.4.1 Case Study: An Energy Decomposition of a Flagged Transaction

To illustrate the interpretability of the Energy Flow Classifier (EFC), we present a case study of a transaction flagged as anomalous. The transaction under analysis was part of the test set in Experiment 3 and labeled as illicit in the ground truth.

The EFC model assigns an energy score to each transaction based on a set of learned local fields h_i and pairwise couplings e_{ij} derived from licit data. Table 4.7 shows the input feature values for the transaction, its corresponding energy components, and the top contributing feature interactions.

Table 4.7: Energy decomposition of an illicit transaction flagged by EFC.

Feature	**Value**	**Local Field Contribution ($h_i x_i$)**
Input count	3	0.21
Output count	5	0.37
Transaction fee	0.015	0.12
Total output value	2.1	0.45
Time step	21	0.28
Top pairwise contributions $e_{ij} x_i x_j$		
Input count × Output count	3 × 5	0.48
Transaction fee × Output value	0.015 × 2.1	0.34
Time step × Output count	21 × 5	0.52

The total energy for this transaction is computed as the sum of all local field contributions and selected pairwise interactions:

$$\mathcal{E}(x) = \sum_i h_i x_i + \sum_{i<j} e_{ij} x_i x_j = 1.43 + 1.34 = 2.77$$

This value exceeds the threshold defined by the 90th percentile of the energy distribution from the training set and results in the transaction being classified as anomalous. The high contributions from pairwise features like ‘time step × output count‘ and ‘input count × output count‘ suggest behavioral irregularities compared to typical licit flows, highlighting how EFC identifies suspicious patterns through learned feature interactions.

4.5 Threats to Validity

Our methodology faces validity threats across four dimensions: internal, construct, external, and dataset-specific limitations. These constraints affect the scope, generalizability, and reproducibility of our findings.

4.5.1 Validities and Dataset Limitations

Internal validity concerns arise from several methodological choices. We used default EFC hyperparameters (`n_bins=30`, `cutoff_quantile=0.9`) without systematic optimization. Performance, especially the precision-recall trade-off, depends on these settings. Additionally, techniques such as SMOTE involve inherent randomness that could affect reproducibility, while the temporal split employed (time steps 1–34 for training, 35–49 for

testing), though standard practice, represents a single instance, and alternative splits might yield different quantitative results. Furthermore, feature selection was conducted using `SelectKBest` with ANOVA F-value, but other feature selection techniques could potentially identify different feature subsets and consequently alter model performance.

With respect to construct validity, our measurement of EFC's ability to assign higher energy to illicit transactions relies on dataset labels where the "energy" concept represents an abstraction serving as a proxy for illicit detection rather than a direct measure of criminal activity. The study depends entirely on the Elliptic dataset's labeling scheme, such that any noise or bias in these labels directly affects both training and evaluation outcomes. Furthermore, our methodology relied on using labeled data from both classes during training, which differs from pure one-class learning scenarios where only normal data is available—this means our results reflect performance under conditions of limited but not absent illicit labels.

Regarding external validity and generalizability, several factors circumscribe the applicability of our findings beyond this experimental context. The results derive solely from the Elliptic Bitcoin dataset, and extending conclusions to other cryptocurrency networks or broader fraud detection domains requires further empirical validation. The dynamic nature of fraudulent activities introduces the risk of performance degradation over time as illicit actors adapt their strategies, and our analysis captures only a specific temporal window that may not reflect evolving patterns. Furthermore, although certain features are aggregated from local graph neighborhoods, our EFC implementation primarily operated on node features without explicitly modeling transaction graph topology in the core energy calculation, potentially limiting applicability to scenarios where relational structure proves critical.

We employed the original Elliptic dataset (Elliptic1) rather than Elliptic++ (Elliptic2), a choice driven by three primary constraints: temporal considerations (research commenced in April 2024 before Elliptic++'s stabilization in July 2024), computational feasibility (Elliptic++'s 49 million nodes and 196 million edges demand 160 CPU cores and 1.2TB RAM, exceeding available resources), and methodological rationale (establishing baseline performance on Elliptic1's tractable structure of 204K nodes before scaling to more complex architectures). Consequently, generalizability to larger-scale Bitcoin transaction networks remains limited. Although Elliptic1 provides a controlled environment for validating the EFC model's performance, interpretability, and explainability under resource-constrained conditions, no formal scalability analysis was conducted for significantly larger datasets, leaving unanswered questions about computational feasibility and performance stability at scale. Future work should extend these findings to Elliptic++ using distributed training techniques to confirm the model's effectiveness on heterogeneous,

large-scale transaction graphs with expanded feature spaces.

Assumptions and Limitations

The EFC assumes stationarity in licit behavior, meaning that if fraudsters successfully mimic normal transaction patterns or if legitimate behavior evolves over time, the accuracy of the model may be compromised. Additionally, the discretization process inherent to EFC may reduce feature resolution, and careful tuning of the bin count is necessary to preserve important discriminative information that could otherwise be lost through coarse binning. Finally, the method does not model the graph structure explicitly; while transaction flows are part of the Elliptic dataset, the EFC is strictly feature-based and does not exploit edge information or relational patterns that might provide additional predictive power to detect illicit activity.

The Elliptic2 dataset

The *Elliptic2* dataset is a large-scale, labeled graph dataset designed to learn the representation of subgraphs in the context of anti-money laundering (AML) on the Bitcoin blockchain. It was introduced by [40] as a successor to the *Elliptic1* dataset, offering significant advancements in scale and task complexity. The dataset is publicly available and has been widely adopted for benchmarking graph neural networks (GNNs) in financial forensic and subgraph classification tasks.

Elliptic2 consists of a background graph with 49 million node clusters and 196 million edge transactions. Within this graph, 121,810 subgraphs are labeled as *licit* (119,047 subgraphs) or *suspicious* (2,763 subgraphs), representing a highly imbalanced classification problem. Each node in the background graph is associated with 43 features, while the edges contain 95 features, including transaction volume, fees, and timestamps. To preserve intellectual property, continuous features were discretized into ordinal bins.

Chapter 5

Conclusion and Future Work

Cryptocurrencies enable peer-to-peer commerce but also facilitate money laundering, ransomware payments, and terrorism financing. Detection is challenging because transactions are pseudonymous, labeled data is scarce, and fraudsters constantly adapt their strategies. This dissertation evaluated the Energy Flow Classifier (EFC)—a physics-inspired, one-class anomaly detector—for identifying illicit Bitcoin transactions under conditions where labeled fraud examples are limited but not entirely absent, reflecting realistic operational scenarios.

The methodology involved adapting EFC to the Elliptic dataset, a benchmark for evaluating Bitcoin transaction classification. After discretizing continuous features, learning empirical feature covariances, and estimating interaction matrices, the model assigned energy scores to transactions and flagged those exceeding a learned threshold. Multiple experiments evaluated EFC's performance under different configurations, including data balancing techniques, feature selection strategies, and various preprocessing combinations.

The results demonstrate that EFC occupies a middle ground between unsupervised and supervised methods. When combining SMOTE oversampling with SelectKBest feature selection (k=30), EFC achieved F1-macro 0.81 and illicit F1 0.77 on imbalanced test data. This substantially exceeds prior unsupervised baselines while using the available labeled data from both classes while using the available labeled data from both classes and approaches the 0.83 achieved by fully supervised Random Forest models that require labeled illicit examples. Supervised baselines such as Random Forest (F1 0.83) and Graph Convolutional Networks (F1 0.87) achieve higher scores but need labeled illicit data. In contrast, traditional unsupervised methods fail: Isolation Forest scores 0.00–0.01, Local Outlier Factor 0.11–0.19, and One-Class SVM 0.01–0.04.

5.1 Comparison to Previous Work

This study builds on and diverges from previous research on detecting illicit Bitcoin transactions, particularly the work in [4], which also utilized the Elliptic dataset and addressed the challenge of label scarcity.

The research conducted in [4] explored various standard machine learning classifiers, including Random Forests, SVMs, and MLPs, employing supervised or semi-supervised learning frameworks. These approaches, while demonstrating potential, typically require a certain number of labels for both licit and illicit classes, or sophisticated semi-supervised strategies to leverage unlabeled data.

Our research takes a different methodological path by investigating the Energy Flow Classifier (EFC), a physics-inspired model rooted in statistical mechanics. The key distinctions and contributions are:

- **Imbalanced Learning Approach**: EFC was applied to highly imbalanced data where illicit transactions represent only 10% of labeled training examples. This addresses practical label scarcity scenarios where some confirmed fraud cases exist but comprehensive labeling remains prohibitively expensive, learning a model of "normal" behavior, and identifying deviations. This contrasts with the supervised/semi-supervised methods in [4] which generally learn from both classes or use unlabeled data in conjunction with some labels from all classes.
- **Alternative Anomaly Detection Mechanism:** EFC's energy-based formulation provides a distinct way to quantify anomalousness compared to distance-based, density-based, or boundary-based methods common in other one-class classifiers or the discriminative models used by [4].
- **Focus on Preprocessing for Imbalanced EFC:** A significant part of our investigation focused on how data preprocessing (balancing, feature selection) impacts EFC's one-class performance, which is crucial given its sensitivity to imbalance as shown in our baseline experiment.
- **Comparable Evaluation Metric:** We adopted the F1-Macro score as the primary evaluation metric, consistent with [4], to facilitate a conceptual comparison with respect to performance on unbalanced data.

Table 5.1: Comparison of F1-scores for illicit transaction classification in the Elliptic dataset across studies.

Model	**[4] (2020)**	**[3] (2019)**	**Notes**
Random Forest (Supervised)	0.81 (5% cont.)	0.83	Requires labeled data
Logistic Regression	–	0.82	Linear baseline
Multilayer Perceptron (MLP)	–	0.85	Deep supervised model
Graph Convolutional Network (GCN)	–	0.87	Exploits graph topology
Isolation Forest (IF)	0.00–0.01	–	Unsupervised anomaly detection
Local Outlier Factor (LOF)	0.11–0.19	–	Sensitive to density assumptions
One-Class SVM (OC-SVM)	0.01–0.04	–	Kernel-based unsupervised
EFC (this work)	**0.81**	–	One-class, evaluated on imbalanced data

Table 5.1 provides a comparative summary of F1-scores reported by [4] and [3] using various anomaly detection and supervised classification models on the Elliptic dataset. While supervised models such as Random Forest and Graph Convolutional Networks (GCNs) reported the highest performance—with GCNs reaching an F1-score of 0.87—these models depend heavily on labeled training data, which is not always available in practice. In contrast, unsupervised approaches like Isolation Forest and One-Class SVM exhibited significantly lower performance under the same conditions, struggling to detect subtle patterns in illicit activity. The Energy Flow Classifier (EFC), despite being a one-class method trained solely on licit transactions, achieves an F1-score of 0.77—comparable to the performance of some supervised baselines. This highlights its practical viability in label-scarce environments and validates its position as a compelling alternative to traditional anomaly detection techniques

Although authors in [4] demonstrated the utility of established ML techniques under label scarcity, our work explores EFC as a novel alternative specifically suited for scenarios where only normal data are reliably labeled. The results, particularly with SMOTE and feature selection (F1-Macro 0.808 on imbalanced test data), suggest that EFC can be competitive. A direct quantitative performance benchmark against the specific results of [4] would require replicating their exact experimental setup or vice versa, which was outside the scope of this study. Our contribution lies in demonstrating the viability and optimization of this alternative one-class approach.

It is important to note that EFC achieved F1-Macro of 0.908 when both training with SMOTE and testing on balanced data (Experiment 1). However, for fair comparison with operational fraud detection systems and the baselines reported by [4] and [3], we emphasize our results on imbalanced test data (0.81), which better reflect real-world deployment conditions where illicit transactions remain a small minority.

5.2 Contributions

This research provides three contributions to cryptocurrency fraud detection:

- **Adaptation of physics-inspired models to financial crime detection.** We demonstrate that the Energy Flow Classifier, originally developed for network intrusion detection, can be successfully adapted to cryptocurrency fraud detection. This extends the application of statistical physics models to blockchain forensics, achieving F1-macro 0.81 under realistic conditions of severe class imbalance where illicit examples constitute only 10% of labeled training data.
- **Identification of optimal preprocessing strategy for one-class classifiers.** Through systematic experimentation, we established that combining SMOTE oversampling with SelectKBest feature selection (k=30) yields superior performance compared to either technique in isolation. This configuration enables EFC to achieve detection rates approaching supervised methods while effectively handling scenarios where confirmed illicit examples are limited but present in the training data.
- **Empirical validation of interpretable anomaly detection for compliance contexts.** We provide evidence that energy-based models offer a viable alternative to black-box approaches in regulated environments. The model's transparent decision mechanism—based on statistical deviations quantified through energy scores—supports audit trails and regulatory reporting requirements that deep learning models cannot easily satisfy.

5.3 Limitations

The EFC implementation evaluated in this dissertation has several limitations that constrain its applicability and performance. The current implementation operates on node-level features without modeling temporal dependencies or transaction graph structure. Bitcoin money laundering often involves complex patterns such as layering and smurfing that manifest across time and through specific graph topologies. The absence of these dimensions limits the model's ability to detect sophisticated laundering schemes that exploit sequential or network-based obfuscation strategies.

EFC requires binning continuous features into discrete categories, which can obscure subtle behavioral gradients. The choice of bin count (n_bins=30 in our experiments) represents a trade-off between feature resolution and data sparsity. Coarse binning may merge distinct transaction patterns, while fine-grained binning risks insufficient statistical support for energy estimation. This discretization process is fundamental to EFC's sta-

tistical mechanics formulation but may discard information that continuous-value models could leverage.

The model employs a fixed energy threshold (95th percentile of training data) for classification decisions. This approach lacks adaptivity to concept drift, coordinated attacks, or changes in legitimate transaction patterns over time. Additionally, while our experiments demonstrate EFC's effectiveness on imbalanced labeled data, we did not evaluate its performance in true one-class scenarios where no illicit labels are available during training—this remains an important direction for future research to assess EFC's applicability when only normal transaction data can be obtained.

Evaluation relied exclusively on the Elliptic dataset, which contains 203,769 transactions with only 23% labeled. The dataset provides anonymized features without semantic interpretation, limiting opportunities for domain-specific feature engineering. Furthermore, the decision to use Elliptic rather than Elliptic2 (which contains 49 million nodes and 196 million edges) was driven by computational constraints. Elliptic2 requires 160 CPU cores and 1.2TB RAM, exceeding available resources. This choice leaves questions about scalability to larger transaction networks unanswered, as no formal analysis was conducted to assess computational feasibility or performance stability at scale.

The evaluation focused primarily on threshold-based classification metrics (F1-score, precision, recall) without reporting AUC-ROC curves or conducting detailed threshold sensitivity studies. Additionally, the evaluation did not assess performance against adversarial attacks where fraudsters actively attempt to evade detection, nor did it investigate fairness implications across different transaction types or user populations. These gaps limit our understanding of EFC's robustness in adversarial settings and its potential for discriminatory outcomes.

5.4 Future Work

Several directions emerge from this research to address current limitations and extend the applicability of physics-inspired fraud detection. Extending EFC to operate on graph embeddings or temporal sequences would enable the model to capture transaction patterns that unfold over time or through network structure. This could involve learning energy functions over graph neural network representations or incorporating recurrent mechanisms to model sequential dependencies. One promising approach would integrate EFC with message-passing neural networks, where energy scores are computed over learned node embeddings rather than raw features, thereby combining interpretability with structural awareness.

Combining EFC's interpretable energy formulation with learned representations from deep neural networks represents another direction. One approach would use EFC as a regularization term within a larger neural architecture, preserving interpretability while gaining representational flexibility. Alternatively, ensemble systems that integrate EFC with graph-based classifiers could balance statistical regularity with structural awareness. Such hybrid architectures might assign transactions to risk categories based on both energy scores and graph-derived features, providing multiple lines of evidence for fraud detection decisions.

Rather than relying on fixed percentile thresholds, future work should investigate dynamic threshold adjustment based on temporal windows, transaction metadata, or streaming data characteristics. Active learning frameworks that incorporate analyst feedback could enable semi-supervised refinement of decision boundaries. For instance, transactions flagged by EFC but validated as licit by human analysts could be used to recalibrate the energy distribution, gradually improving specificity without requiring extensive illicit labels. This creates a feedback loop where model performance improves through operational deployment.

Testing EFC on larger datasets such as Elliptic2 (49 million nodes, 196 million edges) would validate its computational feasibility and performance stability at scale. This requires investigating distributed training techniques and efficient matrix inversion methods for high-dimensional covariance estimation. Approximate inference methods, such as low-rank matrix approximations or stochastic gradient descent over energy parameters, may enable scalability while maintaining detection accuracy. Benchmarking EFC against deep learning methods on Elliptic2 would also clarify whether its computational efficiency advantage persists at larger scales.

Investigation of EFC's false positive distribution across different user populations and transaction types is necessary to ensure equitable deployment. False positives can result in frozen funds or account suspension, raising questions about due process and appeals mechanisms. Additionally, integration studies with existing compliance workflows, regulatory reporting systems, and analyst interfaces would inform practical adoption. Understanding how EFC outputs can be translated into actionable intelligence for compliance teams represents a critical step toward real-world deployment.

Evaluating EFC on other cryptocurrency networks (Ethereum, Monero) would assess the generalizability of physics-inspired approaches beyond Bitcoin. This extends to other financial crime domains such as credit card fraud or trade-based money laundering detection, where similar label scarcity challenges exist. Cross-domain validation would determine whether EFC's energy-based formulation captures universal patterns of anomalous financial behavior or whether domain-specific adaptations are required.

Future research should also address the broader implications of automated fraud detection in decentralized systems. Ensuring that detection algorithms are fair, robust, and accountable requires interdisciplinary collaboration among legal scholars, financial investigators, regulators, and technologists. Developing standardized evaluation protocols, transparency requirements, and contestability mechanisms will be essential as machine learning systems assume greater roles in financial compliance and law enforcement.

Reproducibility

To ensure the reproducibility of our findings, all code, configuration files, and scripts used for the experiments described in this dissertation are publicly available in a dedicated repository: `https://github.com/kevinsantana/PPCA-UnB-Dissertation`.

Computational Environment

The experiments were conducted on a system with the following specifications:

- Operating System: macOS 14.5 23F79 arm64
- Processor: Apple M1 Pro
- GPU: Apple M1 Pro
- Memory (RAM): 32 GB

References

[1] *Bitcoin Developer Transactions Dev Guide.* https://developer.bitcoin.org/devguide/transactions.html. xii, 5, 6

[2] Bellei, Claudio: *The elliptic data set: opening up machine learning on the blockchain. medium (aug. 2019)*, 2019. xii, 15

[3] Weber, Mark, Giacomo Domeniconi, Jie Chen, Daniel Karl I. Weidele, Claudio Bellei, Tom Robinson, and Charles E. Leiserson: *Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics*, 2019. https://arxiv.org/abs/1908.02591. xii, xv, 2, 13, 14, 16, 23, 24, 25, 28, 31, 51

[4] Lorenz, Joana, Maria Inês Silva, David Aparício, João Tiago Ascensão, and Pedro Bizarro: *Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity*, 2021. https://arxiv.org/abs/2005.14635. xv, 1, 2, 3, 13, 18, 19, 20, 21, 24, 25, 26, 28, 50, 51, 90

[5] Chainalysis: *The 2024 crypto crime report*, February 2024. https://go.chainalysis.com/rs/504-JAF-931/images/Crypto-Crime-Report-2024.pdf, The latest trends in ransomware, scams, hacking, and more. 1

[6] Scharfman, Jason: *The Cryptocurrency and Digital Asset Fraud Casebook, Volume II: DeFi, NFTs, DAOs, Meme Coins, and Other Digital Asset Hacks.* Palgrave Macmillan, 2024. https://doi.org/10.1007/978-3-031-60836-0. 1

[7] Khiari, Wided, Azhaar Lajmi, Amira Neffati, and Ahmed El Fahem: *Cryptocurrency fraud and its effects on price volatility in the cryptocurrency market.* Journal of Chinese Economic and Foreign Trade Studies, 2025. https://www.emerald.com/insight/1754-4408.htm. 1

[8] Pontes, Camila F. T., João J. C. Gondim, Matt Bishop, and Marcelo Antonio Marotta: *A new method for flow-based network intrusion detection using inverse statistical physics.* CoRR, abs/1910.07266, 2019. http://arxiv.org/abs/1910.07266. 2, 9, 10, 13, 18, 20, 21, 23, 27

[9] Souza, Manuela M. C., Camila Pontes, Joao Gondim, Luis P. F. Garcia, Luiz DaSilva, and Marcelo A. Marotta: *A novel open set energy-based flow classifier for network intrusion detection*, 2022. https://arxiv.org/abs/2109.11224. 2, 10, 27

[10] Buchanan, Bonnie: *Money laundering—a global obstacle.* Research in International Business and Finance, 18(1):115–127, 2004. 7

[11] Pérez-Cano, Federico and Eric Jurado: *Fraud detection in cryptocurrency networks—an exploration using anomaly detection and heterogeneous graph transformers.* Future Internet, 17(1):44, 2025. 7

[12] Sharma, A., S. Gupta, and R. Singh: *Comparative evaluation of anomaly detection methods for fraud detection in online credit card payments.* In *International Conference on Information and Communication Technology (ICICT)*, pages 39–52. Springer, 2024. 7

[13] Kamalaruban, Prashan *et al.*: *Evaluating fairness in transaction fraud models: Fairness metrics, bias audits, and challenges.* arXiv preprint arXiv:2409.04373, 2024. 7, 19

[14] Van Wegberg, Rolf, Jan Jaap Oerlemans, and Oskar van Deventer: *Bitcoin money laundering: mixed results? an explorative study on money laundering of cybercrime proceeds using bitcoin.* Journal of Financial Crime, 25(2):419–435, 2018. 7, 8

[15] Hu, Yining, Suranga Seneviratne, Kanchana Thilakarathna, Kensuke Fukuda, and Aruna Seneviratne: *Characterizing and detecting money laundering activities on the bitcoin network*, 2019. `https://arxiv.org/abs/1912.12060`. 8

[16] Belen-Saglam, Rahime, Enes Altuncu, Yang Lu, and Shujun Li: *A systematic literature review of the tension between the gdpr and public blockchain systems.* arXiv preprint arXiv:2210.04541, 2022. 8

[17] Marangone, Edoardo, Claudio Di Ciccio, Daniele Friolo, Eugenio Nerio Nemmi, Daniele Venturi, and Ingo Weber: *Enabling data confidentiality with public blockchains.* arXiv preprint arXiv:2308.03791, 2023. 8

[18] Tiwari, Milind, Jamie Ferrill, and Douglas MC Allan: *Trade-based money laundering: a systematic literature review.* Journal of Accounting Literature, 47(5):1–26, 2024. 8

[19] Longa, Francesco Ernesto Alessi: *Cryptocurrency and money laundering.* American Journal of Industrial and Business Management, 15(2):362–371, 2025. 8

[20] Pontes, Camila F. T., Manuela M. C. de Souza, Joao J. C. Gondim, Matt Bishop, and Marcelo Antonio Marotta: *A new method for flow-based network intrusion detection using the inverse potts model.* IEEE Transactions on Network and Service Management, 18(2):1125–1136, June 2021, ISSN 2373-7379. `http://dx.doi.org/10.1109/TNSM.2021.3075503`. 13, 18, 20, 23

[21] Lopes, Daniele Adriana Goulart: *Detecção de botnets baseada na análise de fluxos de rede utilizando estatística inversa.* 2022. 13

[22] Lin, Chang Yi, Hsiang Kai Liao, and Fu Ching Tsai: *A systematic review of detecting illicit bitcoin transactions.* Procedia Computer Science, 207:3217–3225, 2022. 17, 21

[23] Alarab, Ismail, Simant Prakoonwit, and Mohamed Ikbal Nacer: *Comparative analysis using supervised learning methods for anti-money laundering in bitcoin.* In *Proceedings of the 2020 5th international conference on machine learning technologies*, pages 11–17, 2020. 18, 24

[24] Elmougy, Youssef and Oliver Manzi: *Anomaly detection on bitcoin, ethereum networks using gpu-accelerated machine learning methods.* In *2021 31st International Conference on Computer Theory and Applications (ICCTA)*, pages 166–171. IEEE, 2021. 18, 23, 24

[25] Shevchuk, Ivan, Ivan Akhmetshin, Oleksandr Kravchenko, and Pavlo Golubenko: *Anomaly detection in blockchain: A systematic review of trends, challenges, and future directions.* Applied Sciences, 15(15):8330, 2025. 18

[26] Ali, Haider, Muhammad Farooq, Hassan Alghamdi, and Umar Iqbal: *Financial fraud detection based on machine learning: A systematic literature review.* Applied Sciences, 12(19):9637, 2022. 19

[27] Pham, Thai and Steven Lee: *Anomaly detection in bitcoin network using unsupervised learning methods.* arXiv preprint arXiv:1611.03941, 2016. 19, 20, 21

[28] Hirshman, Jason, Yifei Huang, and Stephen Macke: *Unsupervised approaches to detecting anomalous behavior in the bitcoin transaction network.* Technical report, Stanford University, 2013. 19

[29] Monamo, Patrick, Vukosi Marivate, and Bheki Twala: *Unsupervised learning for robust bitcoin fraud detection.* In *2016 Information Security for South Africa (ISSA)*, pages 129–134. IEEE, 2016. 19

[30] Vlahavas, George, Kostas Karasavvas, and Athena Vakali: *Unsupervised clustering of bitcoin transactions.* Financial Innovation, 10(1):25, 2024. 20, 21

[31] Palma, Gabriel Rodrigues, Phil Maguire, *et al.*: *Combining supervised and unsupervised learning methods to predict financial market movements.* arXiv preprint arXiv:2409.03762, 2024. 21

[32] Souza, Manuela M. C., Camila Pontes, Joao Gondim, Luis P. F. Garcia, Luiz DaSilva, and Marcelo A. Marotta: *A novel open set energy-based flow classifier for network intrusion detection*, 2022. `https://arxiv.org/abs/2109.11224`. 21

[33] Alotibi, Johrha, Badriah Almutanni, Tahani Alsubait, Hosam Alhakami, and Abdullah Baz: *Money laundering detection using machine learning and deep learning.* International Journal of Advanced Computer Science and Applications, 13(10), 2022. 21, 22, 23, 24

[34] Nerurkar, Pranav: *Illegal activity detection on bitcoin transaction using deep learning.* Soft Computing, 27(9):5503–5520, 2023. 22

[35] Lo, Wai Weng, Gayan K Kulatilleke, Mohanad Sarhan, Siamak Layeghy, and Marius Portmann: *Inspection-l: self-supervised gnn node embeddings for money laundering detection in bitcoin.* Applied Intelligence, 53(16):19406–19417, 2023. 22, 23, 24

[36] Hisham, Sabri, Mokhairi Makhtar, and Azwa Abdul Aziz: *Combining multiple classifiers using ensemble method for anomaly detection in blockchain networks: A comprehensive review*. International Journal of Advanced Computer Science and Applications, 13(8), 2022. 22, 23, 24

[37] *EFC-package: Energy-based Flow Classifier*. `https://github.com/EnergyBasedFlowClassifier/EFC-package`, 2021. Version 0.1.0, Accessed: [18/04/2024]. 27

[38] Montgomery, Douglas C.: *Design and Analysis of Experiments*. John Wiley & Sons, Hoboken, NJ, 9th edition, 2017, ISBN 978-1-119-32093-7. 37, 38, 39

[39] Box, George E. P., J. Stuart Hunter, and William G. Hunter: *Statistics for Experimenters: Design, Innovation, and Discovery*. Wiley-Interscience, Hoboken, NJ, 2nd edition, 2005, ISBN 978-0-471-71813-0. 37, 38

[40] Bellei, Claudio, Muhua Xu, Ross Phillips, Tom Robinson, Mark Weber, Tim Kaler, Charles E Leiserson, Jie Chen, *et al.*: *The shape of money laundering: Subgraph representation learning on the blockchain with the elliptic2 dataset*. arXiv preprint arXiv:2404.19109, 2024. 48

[41] Nakamoto, Satoshi: *Bitcoin: A peer-to-peer electronic cash system*. 2008. `https://bitcoin.org/bitcoin.pdf`. 65, 72, 73, 82, 83, 87, 88

[42] Smith, Adam: *The Wealth of Nations: An inquiry into the nature and causes of the Wealth of Nations*. Harriman House Limited, 2010. 65, 67

[43] Durlauf, Steven and Lawrence E Blume: *The new Palgrave dictionary of economics*. Springer, 2016. 66

[44] Goodhart, Charles AE: *The two concepts of money: implications for the analysis of optimal currency areas*. European journal of political economy, 14(3):407–432, 1998. 66

[45] Meneses, Italo Bezerra de: *On the origins of money*. MISES: Interdisciplinary Journal of Philosophy, Law and Economics, 4(2):585–587, 2016. 66

[46] Polanyi, Karl: *Trade and market in the early empires: Economies in history and theory*. 1965. 67

[47] Graeber, David: *Debt: The first 5000 years*. Penguin UK, 2012. 67

[48] Ricardo, David: *On the principles of political economy*. J. Murray London, 1821. 67

[49] Marshall, Alfred: *Principles of economics: unabridged eighth edition*. Cosimo, Inc., 2009. 67

[50] Klein, P.G.: *Principles of Economics*. Ludwig von Mises Institute, 2011, ISBN 9781610162029. `https://books.google.com.br/books?id=GYjEtAEACAAJ`. 67

[51] Hicks, John R: *Theory of employment, interest and money.* The Economic Journal, 46(182):238–253, 1936. 67

[52] Li, Xiuhua: *The formation and spread of the ancient chinese coinage system.* East Asian Archaeology, 3(1):95–106, 2003. 67

[53] Hartill, David: *Cast Chinese Coins.* Trafford Publishing, 2nd edition, 2005. 67

[54] Cribb, Joe: *Money: From Cowrie Shells to Credit Cards.* British Museum Press, 1991. 67

[55] Vries, Ad de: *The Industrial Revolution and the Industrious Revolution.* Cambridge University Press, 2008. 67

[56] Weatherford, Jack: *The History of Money.* Crown Business, 1997. 68

[57] Ferguson, Niall: *The Ascent of Money: A Financial History of the World.* Penguin Books, 2009. 68

[58] Graeber, David: *Debt: The First 5000 Years.* Melville House, 2011. 68, 69

[59] Ingham, Geoffrey: *The nature of money.* Polity, 36(3):387–412, 2004. 68

[60] Goodhart, Charles: *The Two Concepts of Money: Implications for the Analysis of Optimal Currency Areas.* European University Institute, 1998. 68

[61] Gupta, Chirag: *The myth of intrinsic value: The case of fiat money.* Journal of Interdisciplinary Economics, 31(2):177–195, 2019. 68

[62] Reinhart, Carmen M. and Kenneth S. Rogoff: *This Time Is Different: Eight Centuries of Financial Folly.* Princeton University Press, 2018. 68

[63] Friedman, Milton: *The role of government in education.* Economic Affairs, 20(4):4–8, 2000. 68

[64] Mankiw, N. Gregory: *Principles of Macroeconomics.* Cengage Learning, 2014. 69

[65] Blinder, Alan S.: *Quantitative easing: Entrance and exit strategies.* The Economic Journal, 120(519):50–51, 2010. 69

[66] Fund, International Monetary: *World economic outlook, october 2020: A long and difficult ascent.* 2020. 69

[67] Blinder, Alan S.: *The covid-19 crisis: Economic policy implications.* NBER Working Paper Series, w26935, 2020. 69

[68] Federal Reserve System, Board of Governors of the: *Money stock and debt measures h.6 release.* 2023. `https://www.federalreserve.gov/releases/h6/current/default.htm`. 69

[69] Blanchard, Olivier: *Inflation expectations and uncertainty in the time of covid-19: An overview.* NBER Working Paper Series, w28106, 2021. 69

[70] Labor Statistics, Bureau of: *Consumer price index summary*. 2023. https://www.bls.gov/news.release/cpi.nr0.htm. 69

[71] Office, Congressional Budget: *The macroeconomic effects of the american rescue plan act*. 2020. 69

[72] Kahn, Lisa B. and Bhashkar Mazumder: *Job loss and reservation wages during the covid-19 recession*. Brookings Papers on Economic Activity, 51(1):289–356, 2020. 69

[73] Labor, United States Department of: *Minimum wages for tipped employees*. 2023. https://www.dol.gov/agencies/whd/state/tipped. 69

[74] Labor Statistics, Bureau of: *Occupational employment and wages, may 2022*. 2022. https://www.bls.gov/oes/2022/may/oes356011.htm. 70

[75] Azar, Ariel and Ioana E. Marinescu: *Labor market concentration*. NBER Working Paper Series, w26634, 2020. 70

[76] Federal Reserve System, Board of Governors of the: *The federal reserve system: Purposes and functions*. 2021. https://www.federalreserve.gov/aboutthefed/pf.htm. 70

[77] Office, Congressional Budget: *Policies that would increase economic output and employment in the short term*. 2021. 70

[78] Treasury, U.S. Department of the: *The debt to the penny and who holds it*. 2023. https://www.treasurydirect.gov/NP/debt/current. 70

[79] Treasury, U.S. Department of the: *Treasury securities*. 2023. https://home.treasury.gov/policy-issues/financial-markets-financial-institutions-and-fiscal-service/public-debt. 70

[80] Federal Reserve System, Board of Governors of the: *Monetary policy and inflation*. 2022. https://www.federalreserve.gov/monetarypolicy/inflation.htm. 70

[81] Federal Reserve System, Board of Governors of the: *Statement on longer-run goals and monetary policy strategy*. 2020. https://www.federalreserve.gov/monetarypolicy/review-of-monetary-policy-strategy-tools-and-communications-statement-on-longer-htm. 70

[82] Lavoie, Michel: *Currency devaluation and domestic output*. Review of Political Economy, 6(3):309–319, 1994. 71

[83] Bernanke, Ben S.: *The global saving glut and the u.s. current account deficit*. Board of Governors of the Federal Reserve System, 2005. https://www.federalreserve.gov/boarddocs/speeches/2005/200503102/default.htm. 71

[84] Shiller, Robert J.: *Irrational exuberance*. The Economic Journal, 111(471):652–653, 2001. 71

[85] Barski, Conrad and Chris Wilmer: *Bitcoin for the Befuddled*. No starch press, 2014. 73

[86] Diffie, Whitfield and Martin E Hellman: *New directions in cryptography*. In *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*, pages 365–390. 2022. 73, 75

[87] Sanderson, Grant: *ledger.png*. `https://3b1b-posts.us-east-1.linodeobjects.com//content/lessons/2017/bitcoin/ledger.png`, visited on 2023-06-01. 74

[88] Stinson, Douglas Robert and Maura Paterson: *Cryptography: theory and practice*. CRC press, 2018. 75, 76

[89] ElGamal, Taher: *A public key cryptosystem and a signature scheme based on discrete logarithms*. IEEE transactions on information theory, 31(4):469–472, 1985. 75

[90] DocuSign, Inc: *ds_subpage_diagram2.svg*. `https://www.docusign.com/static-c-assets/ds_subpage_diagram2.svg`, visited on 2022-06-04. 75

[91] Stallings, William: *Cryptography and network security principles and practices*, 2006. 75

[92] Barker, Elaine: *Digital signature standard (dss)*, 2013-07-19 2013. 75, 76

[93] Boneh, Dan, Ben Lynn, and Hovav Shacham: *Short signatures from the weil pairing*. In *Advances in Cryptology—ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings 7*, pages 514–532. Springer, 2001. 76

[94] Bruce, Schneier: *Applied cryptography: Protocols, algorthms, and source code in c.-2nd*, 1996. 76

[95] Swan, Melanie: *Blockchain: Blueprint for a new economy*. 2015. `https://www.oreilly.com/library/view/blockchain-blueprint-for/9781491920459/`. 77

[96] Sanderson, Grant: *duplicate-transaction.png*. `https://3b1b-posts.us-east-1.linodeobjects.com//content/lessons/2017/bitcoin/duplicate-transaction.png`, visited on 2023-06-05. 77

[97] Rivest, Ronald L, Adi Shamir, and Leonard Adleman: *A method for obtaining digital signatures and public-key cryptosystems*. Communications of the ACM, 21(2):120–126, 1978. 77

[98] Sanderson, Grant: *invalid.png*. `https://3b1b-posts.us-east-1.linodeobjects.com//content/lessons/2017/bitcoin/invalid.png`, visited on 2023-06-05. 78

[99] Sanderson, Grant: *overdrawn.png*. `https://3b1b-posts.us-east-1.linodeobjects.com//content/lessons/2017/bitcoin/overdrawn.png`, visited on 2023-06-05. 79

[100] Sanderson, Grant: *ledgers.png*. https://3b1b-posts.us-east-1.linodeobjects.com//content/lessons/2017/bitcoin/are-these-the-same.png, visited on 2023-06-05. 81

[101] Dang, Quynh H: *Secure hash standard*. 2015. 81

[102] Butin, Denis: *Hash-based signatures: State of play*. IEEE security & privacy, 15(4):37–43, 2017. 81

[103] Dworkin, Morris: *Sha-256 hash function*. NIST FIPS, 180-4, 2001. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf. 82

[104] Sanderson, Grant: *guess-and-check.png*. https://3b1b-posts.us-east-1.linodeobjects.com//content/lessons/2017/bitcoin/30-zeroes.png, visited on 2023-06-06. 83

[105] El Ioini, Nabil and Claus Pahl: *A review of distributed ledger technologies*. In *On the Move to Meaningful Internet Systems. OTM 2018 Conferences: Confederated International Conferences: CoopIS, C&TC, and ODBASE 2018, Valletta, Malta, October 22-26, 2018, Proceedings, Part II*, pages 277–288. Springer, 2018. 83

[106] Sanderson, Grant: *blocks.png*. https://3b1b-posts.us-east-1.linodeobjects.com//content/lessons/2017/bitcoin/blocks.png, visited on 2023-06-06. 84

[107] Sanderson, Grant: *block-ordering.png*. https://3b1b-posts.us-east-1.linodeobjects.com//content/lessons/2017/bitcoin/block-ordering.png, visited on 2023-06-06. 84

[108] Wood, Gavin *et al.*: *Ethereum: A secure decentralised generalised transaction ledger*. Ethereum project yellow paper, 151(2014):1–32, 2014. 85

[109] Ding, Xingjian, Jianxiong Guo, Deying Li, and Weili Wu: *An incentive mechanism for building a secure blockchain-based internet of things*. IEEE Transactions on Network Science and Engineering, 8(1):477–487, 2020. 85

[110] Sanderson, Grant: *block-reward.png*. https://3b1b-posts.us-east-1.linodeobjects.com//content/lessons/2017/bitcoin/block-reward.png, visited on 2023-06-06. 85

[111] Buterin, Vitalik *et al.*: *A next-generation smart contract and decentralized application platform*. white paper, 3(37):2–1, 2014. 86

[112] Fang, Fan, Carmine Ventre, Michail Basios, Leslie Kanthan, David Martinez-Rego, Fan Wu, and Lingbo Li: *Cryptocurrency trading: a comprehensive survey*. Financial Innovation, 8(1):1–59, 2022. 86

[113] Tan, Evrim, Stanislav Mahula, and Joep Crompvoets: *Blockchain governance in the public sector: A conceptual framework for public management*. Government Information Quarterly, 39(1):101625, 2022, ISSN 0740-624X. https://www.sciencedirect.com/science/article/pii/S0740624X21000617. 86

[114] Szabo, Nick: *Bit gold*. Recuperado de https://nakamotoinstitute. org/bit-gold/TVer página, 2005. 87

[115] DuPont, Quinn: *Cryptocurrencies and blockchains*. John Wiley & Sons, 2019. 87

[116] Sanderson, Grant: *dont-trust-yet.png*. `https://3b1b-posts.us-east-1.linodeobjects.com//content/lessons/2017/bitcoin/dont-trust-yet.png`, visited on 2023-06-06. 87

[117] Bashir, Imran: *Mastering blockchain*. Packt Publishing Ltd, 2017. 88

[118] Sanderson, Grant: *limited-to-2400.png*. `https://3b1b-posts.us-east-1.linodeobjects.com//content/lessons/2017/bitcoin/limited-to-2400.png`, visited on 2023-06-06. 88

[119] Rauchs, Michel and et al.: *The cambridge bitcoin electricity consumption index*. 2021. `https://cbeci.org/`. 89

Supplement I

A Brief History of Money

I.1 Money is Corruptible

Bitcoin (BTC) emerged on the scene in late 2008, allegedly as a response to the financial crisis of 2007-2008, and some have suggested that it was also motivated by frustrations with the bureaucratic nature of the Japanese banking system. However, the latter claim ventures into more conspiratorial territory; although there is no concrete evidence to support this claim, the original author or authors of the Bitcoin whitepaper may have had connections to Japan [41]. Nevertheless, prior to delving into the intricacies of Bitcoin, it is crucial to first explore the concept of money and, more significantly, its foundational aspect: value. In the following sections, we will focus on the economic concept of money as a store of value and medium of exchange and explore how Bitcoin fits into this framework.

What is Money? Or rather, what does money represent?

I.1.1 What is Money

If we asked: *What is man's greatest invention?* What would your answer be? There are many options. Would it be fire? Why does it give us warmth, protection, and the ability to cook our meals? Or maybe you would pick the wheel? Because it is the driving force of the beginnings of trade, commerce, and travel. While both of those are excellent choices, most of the time when we think about the greatest inventions of mankind, we tend to forget one of the most important ones of all: money. Unlike tangible inventions such as fire and the wheel, money has an immaterial nature. It exists as a conceptual construct lacking inherent value and its significance is derived solely from the subjective importance we attribute to it. This intangible nature of money often distinguishes it from other notable inventions in collective human consciousness [42].

Notwithstanding the illusory nature of money, its significance remains unchanged. Before the establishment of monetary systems, human societies engaged in the direct exchange of goods and services, known as the barter system. In this system, individuals traded commodities without an assigned intrinsic value, relying solely on subjective evaluations of desired items. Consequently, each transaction was contingent on the willingness of the parties involved to forfeit possessions in pursuit of their desired commodities. This exchange mechanism resembled a game-like scenario [43]. If I wanted vegetables for my meal but my only possession was cattle, I would be forced to offer one of my animals in exchange for bags of vegetables. Similarly, if I required footwear but specialized in tent production, I would have to surrender an entire tent to obtain a pair of slippers. This barter-based system reveals a prominent issue known as asymmetry. As a tent maker, the exchange of an entire house for simple shoes would undoubtedly leave me feeling disadvantaged. The absence of a standardized medium of exchange presented significant challenges in facilitating agreements between individuals with disparate needs. Moreover, reliance on the fortuitous occurrence of complementary desires, where two individuals simultaneously sought reciprocal possession, further complicated the matter, rendering the process inefficient [44].

Our monetary system serves not only as a medium of exchange, but also as a store of value. However, prior to the advent of money, certain individuals were unable to effectively preserve their wealth, through no fault of their own. Consider the scenario of a farmer selling tomatoes and a tent maker. The tent maker has the ability to amass a substantial portfolio of real estate in the form of tents, which can be bartered year-round with individuals in need of shelter. Consequently, the tent maker has the opportunity to accumulate wealth. In contrast, the farmer who sells tomatoes can only engage in barter transactions during the tomato season. Moreover, due to the perishable nature of tomatoes, long-term storage is not feasible. Thus, despite exerting comparable efforts in their respective businesses, the farmer had no viable means to maintain wealth throughout the year [45]. There is also the problem of having something that only a very few people want. Today, when starting a business, it is often said to find a niche. A small group of people who are very interested in what you have to offer. Before money was a thing, that advice would have left you with nothing worth bartering.

In societies where possessions in high demand, such as weapons, animal skins, and salt, had significant value, individuals who possessed such commodities acquired substantial wealth. The awareness that these items were universally sought-after prompted individuals to engage in anticipatory buying, even if immediate need was absent, to secure future trading opportunities. As a consequence, commodity money emerged, in which goods and services were exchanged for commonly recognized items such as salt or weapons,

facilitating subsequent transactions with other parties [46].

Humanity advanced beyond direct barter, encompassing a diverse range of commodities including salt, weapons, and minutecollectibles such as shells and beads. This evolution introduced a more efficient method of trade and exchange. Rather than directly swapping goods and services, individuals adopted the practice of using arbitrary objects as intermediary placeholders of value, effectively functioning as IOUs (I Owe You). Subsequently, these placeholders could be utilized to acquire desired goods and services from others. This concept proved remarkably ingenious, ultimately leading to a global transition from the Barter system to the monetary exchange system [47]. However, there has been a persistent limitation associated with this form of exchange. For a currency to exhibit intrinsic value, it requires a degree of scarcity [42, 48]. The more easily accessible an item is, the lower its perceived value [49]. When an item is readily available to anyone, its value diminishes considerably. As a result, substances such as sand or shells, which can be easily collected from any beach, do not function effectively as indicators of value [50, 51].

In approximately 770 BC, China witnessed the emergence of the first metal coins, marking a significant milestone in the evolution of currency. As a tribute to their historical currency systems, Chinese craftsmen ingeniously crafted miniature replicas of tools that were previously used as forms of exchange. To ensure convenient handling, the coins were deliberately designed in a circular shape, allowing easy retrieval from the pockets without causing any discomfort to the fingers. These coins were predominantly cast using bronze, thus bestowing intrinsic value on them. This transition marked a pivotal moment in history, as money transformed from a mere symbol to a tangible entity of worth. The scarcity of bronze, a resource that is not readily available on any beach, further amplified the significance of these coins [52, 53]. During this period, the concept of money had not yet deviated from material reality. The valuation of a coin corresponded directly to the intrinsic value of the metal constituting the coin. For example, a coin crafted from 1 gram of gold had an equivalent worth of precisely 1 gram of gold. This quantifiable attribute allowed for straightforward verification through direct measurement, enabling individuals to visually ascertain that the coin indeed comprised 1 gram of gold.

The realization of the potential power of money was swift among Kings and Rulers [54]. This understanding led to the creation of the first official money mint by Alyattes, the King of Lydia, around 600 BC. These coins were minted from a blend of silver and gold, and each coin featured a distinctive image that served as a denomination. Consequently, individuals could effortlessly determine the value of their metal possession by observing the pictorial representation on the coin's surface [55]. The pursuit of greater wealth among Kings led to the devaluation of coins through the reduction of precious metal

content and the inclusion of cheaper metals [56]. This resulted in the divergence between the face value and actual worth of circulating coins, establishing the illusion of money. The value of coins became divorced from the intrinsic value of their metal composition, relying instead on the dictates of rulers and financial institutions [57]. As an example, the British Pound Sterling ceased to represent a fixed quantity of Sterling Silver and instead denoted a unit of currency determined by authoritative decree.

The emergence of international trade exposed the impracticality of metal coins, leading to the introduction of IOU certificates by the Kings to facilitate long-distance transactions [58]. These certificates, bearing the King's stamp, gained trust and were believed to hold value, as they were expected to be exchangeable for equivalent coins. Initially, this belief corresponded to reality. With the proliferation of IOU certificates in circulation, the necessity for physical coins diminished. Ultimately, the value of the certificates became divorced from their direct convertibility into gold and silver coins. Instead, their value relied on collective trust and shared belief [59]. This shift allowed the paper certificates to retain value based on our perception, even in the absence of an immediate exchange for tangible precious metals.

I.1.2 The Illusion of Money

From Ancient Kings to modern-day governments and Central Banks, money has remained an illusion. A mere representation of whose value is determined by the importance people place on it.

The ten thousand Singapore Dollars banknote, while no longer in production, remains the highest denomination in circulation [60]. Despite its intrinsic production cost of fewer than 20 cents, the value of this paper note is upheld by the illusion perpetuated by the fiat currency system [61]. Presently, its monetary equivalence to seven thousand three hundred and forty-five US Dollars enables its utilization in acquiring substantial assets such as houses, cars, and even valuable commodities like gold.

"Fiat" is the fancy word we use to describe the modern-day illusion. It's a Latin word that translates to "let it be done." It's a decree by the government that, in the case of money, determines what its value is and enforces it as legal tender [62, 63].

The elusive nature of money often evades careful consideration, yet akin to historical rulers, contemporary governments possess an understanding of the influential power of currency and persistently strive for its accumulation. Recognizing that the possession of greater quantities of these paper instruments equates to amplified authority, governments adopt the approach of generating additional currency ex nihilo. For instance, in the scenario where the United States government necessitates $340 million dollars to procure

an F-22 jet, it possesses the capacity to create the required funds through the act of monetary printing [58, 64]. But there is one problem with this: **inflation**.

The fundamental attribute of money lies in its role as a medium of exchange, conferring value upon it [64]. Consequently, the quantity of money in circulation should align with the aggregate production of goods and services. Should the issuance of money exceeds the availability of goods and services, with all else remaining constant, the resultant effect is an escalation in prices and a subsequent devaluation of the currency itself. This concern resonates with economists and the general population, including individuals such as ourselves, [65], particularly in the context of the current global reserve currency, the United States Dollar.

The year 2020 proved to be an exceedingly challenging period for the world at large, as the onset of the pandemic necessitated the temporary closure of numerous economies, resulting in a considerable reduction in the availability of goods and services and a marked decline in overall economic output, as outlined in the World Economic Outlook report by the International Monetary Fund [66]. To avert economic collapse and the potential disintegration of societal systems, the US government embarked on an unprecedented scale of monetary expansion, surpassing any previous instances of currency printing in its history [67]. As of 2021, the current state of affairs reveals a considerable expansion of the US dollar supply, with approximately 40% of the existing currency having been printed within the last 18 months [68]. This substantial increase in the money supply with the country's output has raised concerns regarding the potential for significant price inflation [69]. Observable evidence of this trend is already apparent in the substantial rise in commodity prices, such as the tripling of lumber prices compared to a year ago. Additionally, discernible price increases can be observed in everyday experiences, including slight increments in prices at favorite restaurants, such as a modest 20-cent rise in the cost of guacamole at Chipotle [70]. Although the provision of stimulus and unemployment checks by governments to their citizens may initially appear beneficial, it entails a double-edged sword. While it undoubtedly assists individuals in dire economic circumstances, it also introduces challenges. Presently, the combined factors of inflationary pressures and an economic slowdown have created difficulties for individuals seeking suitable employment opportunities, not solely due to a lack of willingness but also because certain job options may be less desirable than available alternatives [71, 72].

An illustrative example can be observed in the United States, where the law does not mandate a minimum wage for individuals working as waiters or waitresses [73]. Consequently, some employees in these roles receive meager hourly wages, such as $2 to $3, with tips constituting a substantial portion of their earnings. However, due to the implementation of various restrictions and regulations nationwide, coupled with a decrease

in customer traffic, there has been a reduction in both customer volume and disposable income, thereby leading to a decline in tip revenue [74]. Inadequate income for employees may result in higher turnover rates as financial needs are not being met. This situation poses a significant risk to businesses, as the lack of a sufficient workforce can ultimately lead to business closure, setting in motion a cascading effect [75]. A valid concern arises regarding the motivation to actively seek employment when the potential income from unemployment and stimulus checks surpasses that from being employed. This circumstance prompts an examination of the available options. Notably, the Federal Reserve of the United States employs a strategic approach to injecting funds into the economy, a process that may not be widely acknowledged, thus stimulating economic activity without substantial public scrutiny [76]. Consequently, the relative attractiveness of alternative income sources may influence individual's decision-making regarding employment prospects [77].

The United States had accumulated a staggering national debt of $29 trillion before 2020, an astounding and challenging figure to comprehend [78]. This debt is primarily financed through the issuance of bonds and Treasury notes, which are essentially contractual instruments offering repayment of a predetermined principal sum alongside interest [79]. Presently, investing in a 10-year U.S. Treasury bond would yield a modest return of 1.23% upon maturity. Therefore, investing $1,000 today would result in a nominal return of a mere $12.30 by 2031. However, this return fails to keep pace with the targeted inflation rate, projected to be around 2% annually [80]. It should be noted that actual inflation rates may surpass the target, although that discussion is beyond the scope of the current context. Consequently, investing in government notes issued by one's own country, whose currency is utilized in daily transactions, leads to a gradual erosion of purchasing power over a decade. Irrespective of these concerns, financial institutions, businesses, and individuals worldwide participate in the acquisition of bonds and treasury notes, thereby providing governments with discretionary funds for utilization [79]. However, when the government confronts the need to fulfill its debt obligations, the previously obtained funds have been fully expended. Consequently, the government initiates repurchases of treasuries and bonds, confining such transactions to prominent financial institutions and remunerating them through freshly created money, effectively conjured from nothingness. The Federal Reserve, for instance, has repurchased over $1 trillion in bonds since March 2020, with plans to persist in such actions well into the future [81].

Through government injections, banks are empowered to expand their lending activities, thereby increasing interest income and fostering economic growth [81]. However, this surge in lending simultaneously expands the aggregate money supply, leading to a depreciation in the value of each dollar. The implementation of multi-trillion dollar stimulus

payments and infrastructure packages raises questions regarding the sustainability of such practices. The influx of new money results in a devaluation of existing money, whereby the balance in an individual's bank account remains unchanged, yet its purchasing power diminishes owing to the influx of newly minted money [82]. Consequently, the retention of wealth in a fiat currency like the US dollar progressively erodes its value, ultimately impeding the ability to acquire goods and services despite nominal bank balances.

The reality that money is nothing but an illusion is one that we must all embrace. Only then will the path to financial freedom become clearer. Understanding that money does not have any intrinsic value in itself but instead only inherits the value we give it.

As the money supply continues to expand, the purchasing power of each dollar held in one's possession inevitably erodes, whereas the dollar-denominated value of global assets tends to appreciate [83]. Nevertheless, this perceived growth can be likened to an optical illusion, employing deceptive mechanisms. Despite the seemingly unrelenting ascent of the stock market, the underlying reality is far from reassuring. The relentless depreciation of the currency compounds the situation, eroding its value daily. For example, if the Dow Jones Industrial Average, which serves as a benchmark for the performance of 30 major US companies, were denominated in terms of gold rather than USD, it would become apparent that its value has essentially stagnated since 1997 [84].

But what's the end goal of all of this? With fiat and an unlimited supply of money, will the value of each currency just continue to decrease until the end of time? Will the gap between the rich and the poor continue to grow wider? Or are we going to finally fix a problem as old as man itself and stop placing our financial success in the hands of those who are destroying it day by day? Money is corruptible.

Only time will tell, but just to know, there is a way out: **Bitcoin**.

Supplement II

Bitcoin

II.1 Bitcoin: A Peer-to-Peer Electronic Cash System

Bitcoin is a decentralized digital currency that operates on a peer-to-peer network called the blockchain. It was introduced in a 2008 whitepaper by an anonymous person or group of people using the pseudonym Satoshi Nakamoto [41]. Bitcoin is not controlled by any central authority, such as a government or financial institution, making it a unique form of currency. It relies on cryptographic techniques to secure transactions and control the creation of new units.

What are the underlying technologies utilized by Bitcoin and what specific events occur during the transfer of a single Bitcoin from one digital wallet to another? In the upcoming sections, we aim to answer these questions.

II.2 How Does Bitcoin Actually Work?

The creation of Bitcoin was prompted by the need for a secure and decentralized value transfer system. The solution to this problem involved an intriguing mathematical puzzle that required the invention of new concepts such as digital signatures and cryptographic hash functions.

Creating a new cryptocurrency is a complex process that involves several steps, including developing a consensus mechanism, creating a blockchain, implementing security measures, and ensuring decentralization. To understand how Bitcoin works and identify potential areas for design improvements, it can be helpful to examine the technical details of its underlying protocols [41]. Alternative cryptocurrencies have emerged as a result of different design choices made by their creators, which has led to a diverse ecosystem of digital currencies with varying features and use cases.

Although the underlying technology may seem complex to some, it is important to note that the use of a cryptocurrency does not require a detailed understanding of its mechanics [85]. Like swiping a credit card, users can take advantage of user-friendly applications that enable seamless sending and receiving of these digital assets.

The concept of cryptocurrency revolves around the ability of individuals to conduct transactions without relying on a centralized entity for trust verification. Typically, when using a credit card to purchase goods or services, one must rely on banks (or a network of banks) to correctly debit the user's account and credit the recipient's account. The majority of currencies are issued by governments, which can exercise some level of control over their respective currencies through means such as adjusting the money supply. As a result, holders of these currencies must place a certain degree of trust in the government issuing them to manage them effectively.

The concept of Bitcoin was inspired by the desire to overcome the limitations of traditional financial systems. According to Nakamoto (2008, p.1) [41]:

> *the root problem with conventional currencies is all the trust that is required to make it work*

To address this issue, Bitcoin was designed as a decentralized digital currency that operates without a central authority or intermediary. The money supply of Bitcoin is fixed and determined by its underlying algorithm, making it resistant to inflation and manipulation. In addition, transactions in the Bitcoin network are recorded on a public ledger called the blockchain, which ensures transparency and accountability. Bitcoin allows for direct peer-to-peer payments without the need for intermediaries, such as banks or payment processors. This property of Bitcoin eliminates the need for trust in a central authority and enables participants to transact with each other directly, thereby reducing transaction costs and increasing efficiency.

The concept of decentralization in trustless payment systems has been the subject of debate among readers. However, this discussion is beyond the scope of our current topic. Although personal needs for trustless payments may vary, the question of whether such a system is technically feasible remains an intriguing one. Cryptography, which originates from the encryption of messages, employs deep mathematical concepts to achieve its objectives. The remarkable effectiveness of cryptographic tools extends beyond confidential communication to other domains. For example, the development of a decentralized currency presents a significant challenge that can be addressed by applying cryptographic techniques [86].

Creating Your Own Cryptocurrency

One common scenario where distributed ledgers can be useful is when multiple individuals frequently exchange small amounts of money, such as paying for shared expenses like dinner bills. To simplify this process, they may choose to maintain a communal ledger that records these transactions in a manner similar to using physical currency. By doing so, participants can easily keep track of their contributions and settle down when necessary.

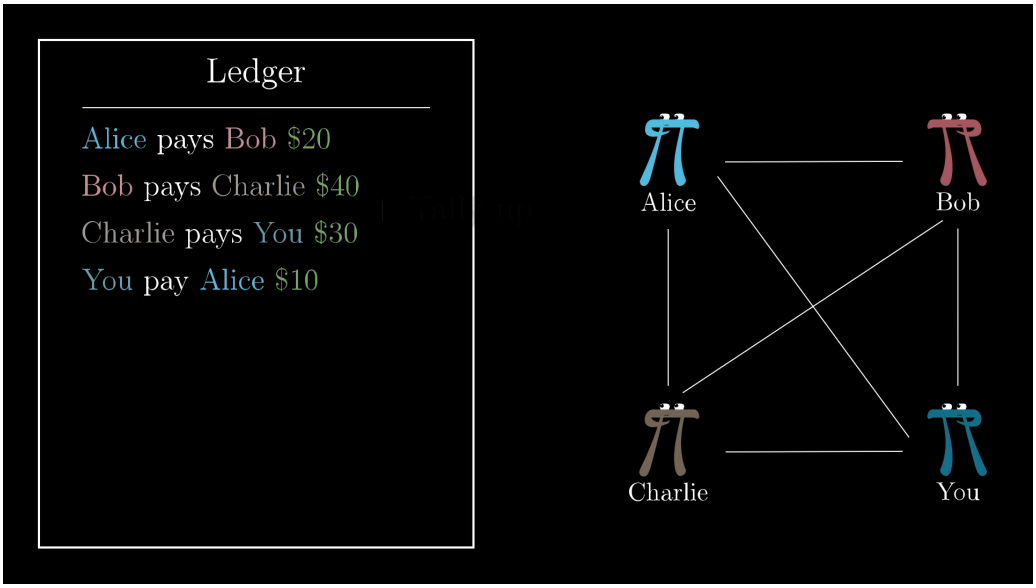


Figure II.1: A ledger is a record of financial transactions, utilized for monitoring the accounts of all parties involved (Reference: [87]).

The proposed ledger system would be a publicly accessible platform similar to a website where users can add new entries. At the end of each month, participants could review the list of transactions and calculate the total sum. If an individual has spent more than they have received, they would contribute that amount to the collective pool, while those who have received more than they have spent would withdraw funds from the pool.

The protocol for participation in the system involves the following steps:

1. Any individual can add entries to the distributed ledger;
2. At the end of each month, all participants gather to reconcile their accounts using physical currency.

However, a potential issue arises with a public ledger that allows any individual to add entries. How can one ensure that Bob does not enter "Alice pays Bob 100" without Alice's approval? There is a cryptography solution: *Digital signatures*.

II.2.1 Digital Signatures

Digital signatures provide a means to ensure the authenticity and integrity of electronic transactions. The use of digital signatures allows recipients to verify that the information sent by a sender is what they intended to send, thus establishing trust in the transaction [88].

The concept described here is similar to a handwritten signature in which Alice can add a message or proof of approval to a transaction that cannot be easily replicated by others. This is achieved through the use of digital signatures, which are based on cryptographic algorithms and provide a secure method of verifying the authenticity of a message or transaction [89]. The infeasibility of forging a signature is ensured by using advanced encryption techniques that make it difficult for unauthorized parties to alter or counterfeit digital signatures [89].

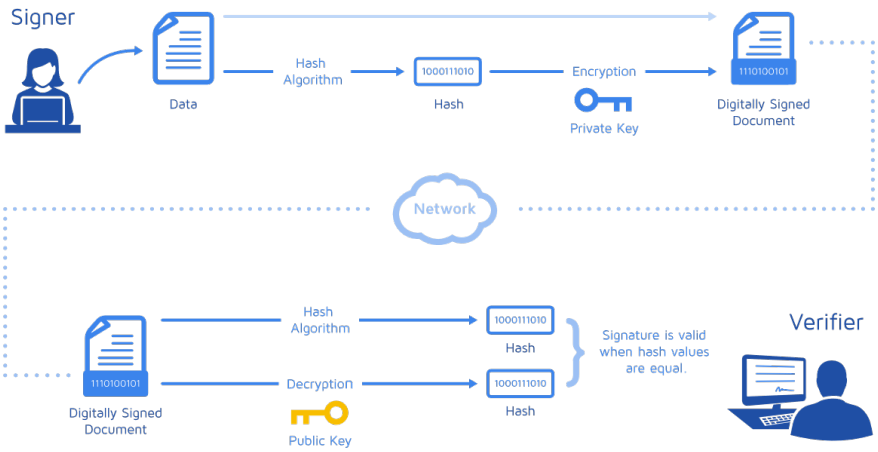


Figure II.2: Digital signature (Reference: [90]).

It may seem counterintuitive at first, but digital signatures can be implemented in a way that prevents forgery. In this context, a digital signature is a function of two elements: the private key, which only the signatory has, and the message being signed [86]. This means that even if an attacker were able to copy the initial signature, subsequent attempts to use it would result in a different value due to the unique relationship between the private key and the message.

In cryptography, a signature function is only effective if there exists a verification function to confirm its validity [91]. The mechanism for this involves generating a public-private key pair consisting of two strings of 1's and 0's. The private key, also known as the *secretkey*, is often abbreviated as sk while the public key is denoted as pk. As suggested by their names, the secret key should be kept confidential [92].

A digital signature scheme can be defined as a set of two operations: one to generate a digital signature on a given message, denoted as Sign, and the other to verify the authenticity of a purported signature, denoted as Verify. These functions are typically implemented as follows:

1. *Signing function* Sign: This operation takes as input a message $m \in \{0,1\}^*$, and produces a digital signature Sign $\in \mathbb{Z}_q^*$, where q is a prime number. The security of the scheme is typically guaranteed by the assumption that it is computationally infeasible to compute the discrete logarithm in the underlying finite field $\mathbb{Z}_q$.

2. *Verification function* Verify: This operation takes as input a message $m \in \{0,1\}^*$, a digital signature Sign $\in \mathbb{Z}_q^*$, and the public key (pk, sk), where $pk = g^x$ for some generator polynomial $g \in \mathbb{Z}[X]$ of degree $n-1$ and $x \in \mathbb{Z}_q$. The verification function outputs a Boolean value indicating whether or not the given signature is valid, that is, $\text{Sign}(m) = g^y \mod q$, where $y \in \mathbb{Z}$ is the unique integer such that $g^{y \bmod n} \equiv sk \pmod q$.

The signing process requires the use of the private key. The objective is that if Alice alone possesses her private key, then she is the only individual capable of generating a digital signature. If this key is compromised, the security of the system is significantly undermined. The Verify function serves as a means of determining whether a given message has a valid digital signature generated using the corresponding public key. It should return True when applied to an authentic signature and False for all other signatures.

The security of a digital signature scheme is based on the secrecy of the private key used to generate the signature. However, it is theoretically possible for an attacker to brute-force the public key and find a valid signature by exhaustively trying different potential signatures until one returns true [93]. In the case of Bitcoin's digital signature scheme, there are 2^{256} possible signatures due to the large number of bits in the hash function used for signature generation [92]. However, this number is so large that it makes brute-force attacks on the public key infeasible, providing a high level of security for Bitcoin's digital signatures.

II.2.2 Ledger

In blockchain systems, transactions are recorded on a distributed ledger and secured using cryptographic techniques. Specifically, each transaction must be signed by its corresponding private key, which ensures its authenticity and non-repudiation [88]. The signature generated for a given transaction is unique and dependent on the content of that transaction, making it impossible to reuse signatures from one transaction to another [94].

However, there is an issue with this approach. Suppose Alice signs a transaction, such as "Alice pays Bob $100", which is then recorded on the blockchain. Although Bob cannot forge Alice's signature on new messages, he could still copy that same line multiple times and submit it to the network. Since the message/signature combination is still valid, these duplicate transactions may be accepted by the network and included in its consensus state [95].

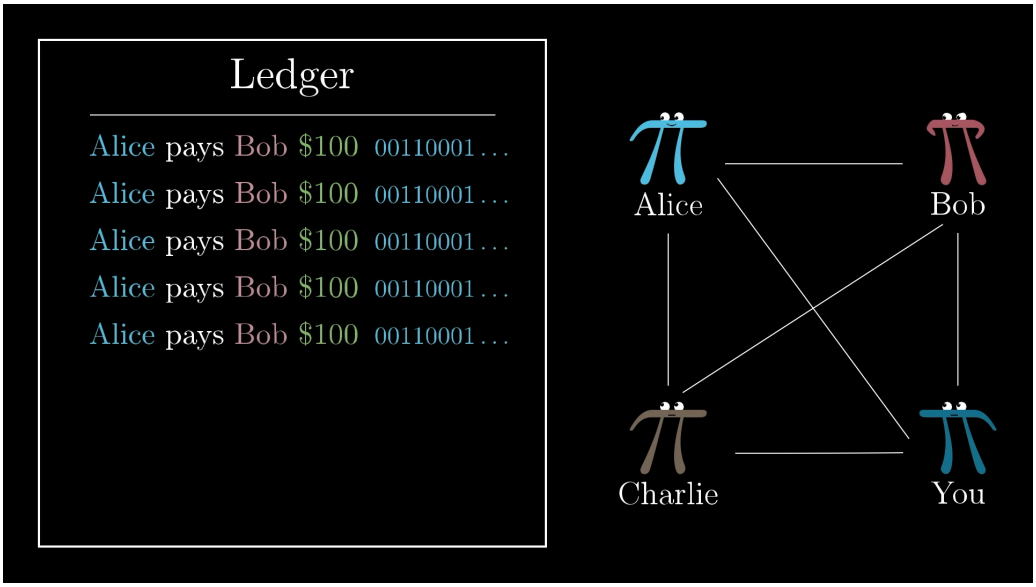


Figure II.3: Anyone can create copies of previous transactions (Reference: [96]).

The development of digital signatures can address the issue of trust in the initial protocol by introducing unique identifiers for transactions and requiring a distinct signature for each transaction. This approach has been proposed and implemented in various cryptographic systems, such as the RSA signature scheme [97] and the elliptic curve digital signature algorithm (ECDSA). The use of digital signatures not only enhances security, but also enables efficient verification of the authenticity and integrity of electronic messages.

Removing Cash

The effectiveness of this system relies on an implicit agreement between individuals to maintain their financial obligations. Specifically, participants are expected to pay in cash at the end of each month, despite the absence of a formal enforcement mechanism. However, there is no guarantee that all parties will comply with this arrangement, as demonstrated by instances where one individual (e.g., Charlie) may accumulate significant debt and subsequently fail to fulfill their financial obligations.

In this cashless economic system, it may be necessary to revert to cash to settle up if certain individuals owe a significant amount of money (e.g., Charlie). However, as long as no one falls into debt and the ledger is properly maintained, the use of cash can be avoided. The ledger alone can function effectively as long as there is a mechanism in place to prevent excessive spending.

One strategy for managing a cashless economy without resorting to cash settlements is to have all participants deposit an equal amount (e.g. $100) into the pot and record the initial distribution of funds on the ledger. For example, Alice would receive $100 in the first transaction, while Bob would receive $100 in the second transaction, and so on. Using this approach, individuals can maintain their financial balance without the need for cash transactions.

Now that we are in a cashless economic system, it is important to prevent double-spending attacks where a user attempts to spend the same cryptocurrency more than once. One way to accomplish this is by verifying that the transactions are valid before they are added to the ledger. Specifically, if all users on the network start with zero balance ($0) and the first two transactions are of $100 value (Charlie pays Alice $50 and Charlie pays Bob $50), then a third transaction where Charlie pays You $20 would be invalid. This is because it violates the rule that a user cannot spend more than they have in their account.

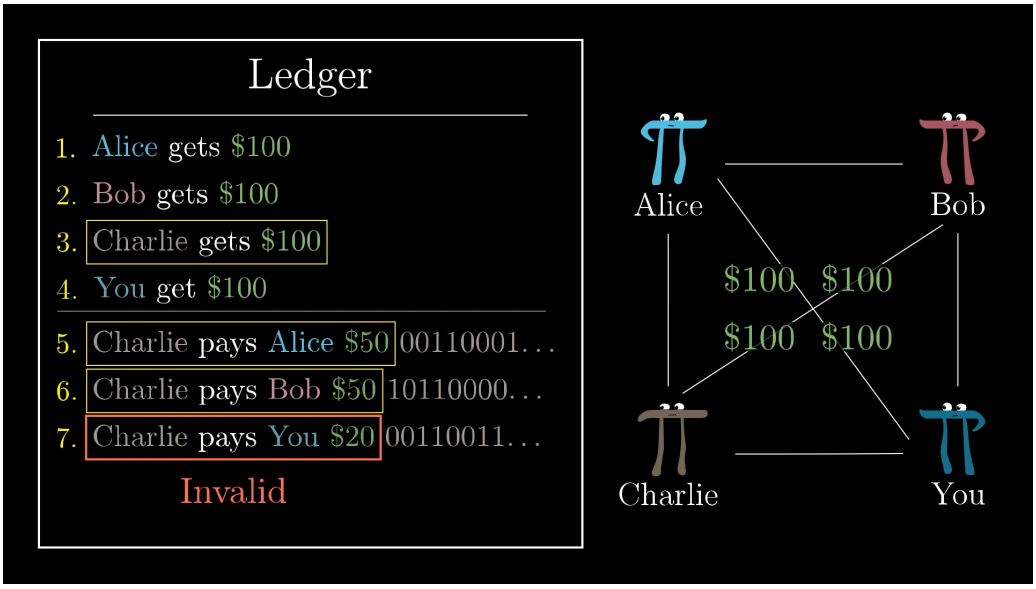


Figure II.4: In this new system, we don't allow people to spend more than they have. (Reference: [98]).

It can be noted that the requirement to determine the legitimacy of a transaction requires knowledge of the entire transaction history. This principle applies not only to

traditional financial systems but also to decentralized digital currencies, although opportunities for improvement are present.

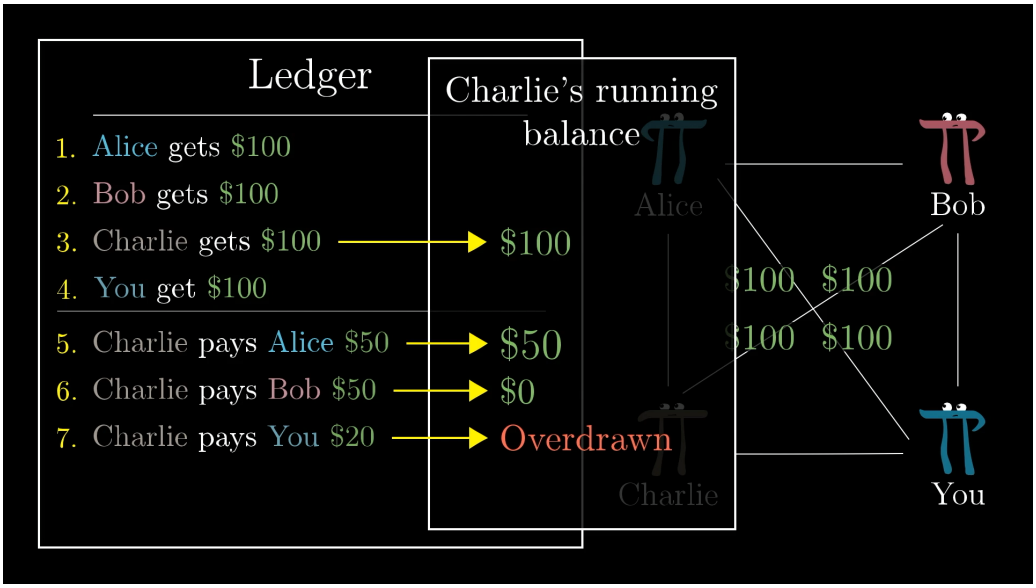


Figure II.5: Now verifying a transaction requires checking the entire ledger history to make sure nobody overdraws. (Reference: [99]).

The use of the above ledger system appears to dissociate it from physical cash transactions. If everyone in the world were to utilize this ledger, one could theoretically conduct all financial transactions solely through the ledger without the need for conversion to United States Dollars (USD). Many individuals currently perform digital transactions exclusively while occasionally using physical cash. The latter scenario involves a more intricate system of banks in which the balance on a digital account can be converted into USD. However, if one and their associates were to completely detach their ledger from USD, there would be no guarantee that having a positive balance in the ledger could translate into physical currency in hand. To accentuate this point, one can stop using the $ sign, and digital quantities in the ledger can be referred to as "Ledger Dollars" (LD).

Individuals possessing Ledger Dollars have the liberty to convert them into US dollars at their discretion. An example involves Alice offering Bob a zero-value US dollar bill in exchange for him adding and signing a transaction entry to the shared ledger, wherein Bob pays Alice ten units of Ledger Dollar value. However, the protocol does not explicitly guarantee the occurrence of such exchanges. Instead, it operates more similarly to foreign currency exchange in an open market where 10LD is its own independent entity. Additionally, if there is high demand for inclusion within the ledger, a transaction of 10LD may require a non-zero amount of physical cash. Conversely, if there is a low demand for participation, it may require only a minimal amount of physical cash.

Our ledger has been transformed into a form of currency that operates within a closed system, allowing for peer-to-peer transactions between individuals without the backing of a state or taxation imposed in the form of Ledger Dollars. It is important to recognize that, at its core, cryptocurrency can be viewed as a ledger that records the history of financial transactions, serving as the currency itself. The concept of possessing Bitcoin is simply represented by a positive balance in the Bitcoin ledger, which is associated with a secret key. This differs from traditional currency systems, where money enters the ledger through cash transactions. In the case of Bitcoin, the process for introducing new money into the ledger will be discussed in more detail shortly. However, it is important to note that there are fundamental differences between Ledger Dollars and true cryptocurrencies.

Distributing The Ledger

The distributed nature of the blockchain technology used by the ledger system necessitates the use of a centralized platform for public access and modification of the ledger's contents. However, this raises concerns about the trustworthiness of the entity responsible for hosting the website and regulating the rules governing the addition of new entries to the ledger. In particular, it is important to identify and evaluate the credibility of the entity that controls the website and establishes the protocols for updating the ledger.

To eliminate trust in a centralized system where one ledger is maintained, we will replace this with a decentralized approach where each individual will maintain their own copy of the ledger. This will enable transactions, such as "Alice pays Bob 100 LD" to be broadcasted and recorded on personal ledgers by all parties involved in the network.

The distributed ledger technology used by Bitcoin involves the broadcast of transactions by users, which are then recorded on a decentralized set of records. This eliminates the need for trust in a central authority. However, this system is problematic due to the possibility of disagreement among participants regarding the correct ledger. For example, when Bob receives a transaction "Alice pays Bob 10 LD", how can he be sure that everyone else has received and believes in the same transaction? If even one person does not know about this transaction, they may not allow Bob to spend those 10 Ledger Dollars later.

Verification of the integrity and consensus of a blockchain network is based on a distributed ledger system where all participants maintain a copy of the same transaction history. The trustworthiness of this system is predicated on the assumption that all nodes will accurately record and remember past transactions, which may be subject to potential inconsistencies or discrepancies in the event of faulty or malicious behavior. Therefore, it is essential to establish a mechanism to ensure that the distributed ledger remains consistent between all participating nodes. The solution proposed by Satoshi Nakamoto in

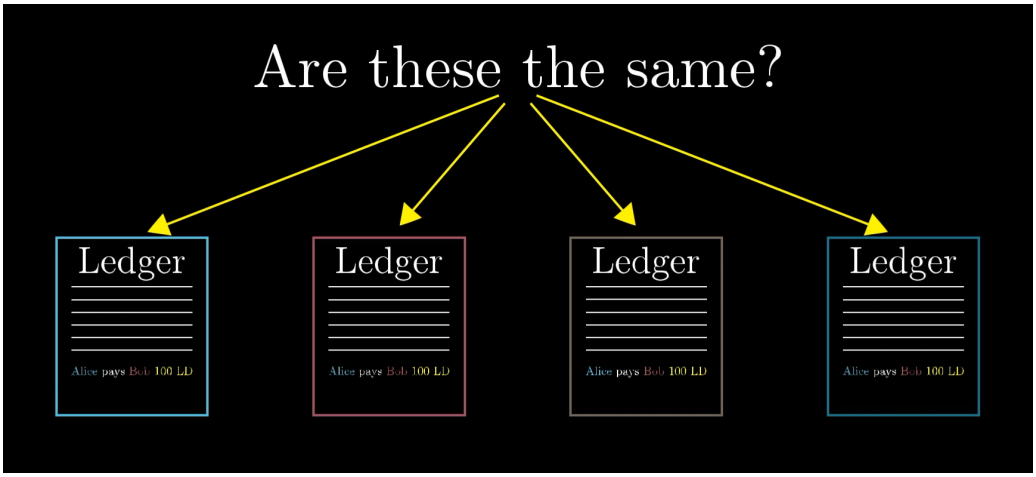


Figure II.6: If everyone keeps a unique copy of the ledger, how can we ensure that everybody agrees on what it should say? (Reference: [100]).

2008 for decentralized systems was a method to validate the validity of a growing document, such as a ledger, without relying on a central authority. This problem was solved through the use of computational work to determine trustworthiness, where the ledger with the most computational effort invested is considered legitimate. The idea is that if an individual attempts to manipulate the ledger, it would require an impractical amount of computational power, making fraudulent transactions computationally infeasible. This concept forms the core of Bitcoin and other cryptocurrencies.

II.2.3 Hash Functions

Cryptographic hash functions are the primary tool utilized by Nakamoto's solution to this puzzle. These functions take arbitrary messages or files as input and produce a fixed-length string of bits referred to as the "hash" or "digest" of the message, which is intended to exhibit randomness. The output of this process is deterministic and consistent for a given input, but minor alterations to the input can lead to drastically different hash values.

The property of unpredictability in the output changes when slightly changing the input is what makes SHA256 a cryptographic hash function [101]. This means that it is computationally infeasible to compute the original message from its hash value in the reverse direction [102]. Therefore, given a specific hash value such as $1001111100111100\ldots$, there is no efficient method to determine the corresponding input message other than brute force guessing and checking with random input.

Given the provided function, what is the empirical evidence indicating a significant

correlation between a specified set of Bitcoin transactions and an exceptional computational expenditure? *Proof-Of-Work*.

II.2.4 Proof-Of-Work

The task described involves manipulating a collection of transactions (enclosed within a container), whose hash value is computed using the SHA256 algorithm. The objective is to modify a specified element within the container such that the resulting hash begins with at least six consecutive zeros.

Achieving a solution to this problem is indeed possible, albeit requiring a considerable amount of time. Due to the inherent unpredictability of the output of the hash function, the predominant method of tackling this challenge remains a process of trial and error [103].

As the number of required leading zeros increases, the difficulty of the problem increases exponentially. Consider a scenario where an individual presents you with a list of transactions and asserts that they have identified a special number. They claim that by appending this number to the end of the transaction list and applying the SHA256 hash function to the entire sequence, the resulting output will exhibit 30 leading zero bits.

Assessing the level of difficulty involved in discovering the aforementioned number requires a thoughtful analysis. It is evident that the task likely presented significant challenges. When considering a randomly selected message, the probability that the resulting hash begins with 30 consecutive zeros is 1 in 2^{30}, which corresponds to approximately 1 in a billion [103]. Consequently, it is highly probable that the individual in question had to iterate through approximately one billion distinct guesses before successfully identifying this specific value.

Nevertheless, what proves intriguing is that once the number is known, its verification as a hash commencing with 30 zeros can be efficiently conducted. This verification process offers the ability to determine the substantial effort expended by the individual without necessitating replication of the original labor. Known as *proof-of-work*, this number has significance.

It is crucial to emphasize that the entirety of this endeavor is intrinsically linked to the underlying list of transactions. Even a slight modification to any transaction would result in a completely altered hash, which requires a full repetition of the laborious process to identify a new number that produces a hash with 30 zeros [41].

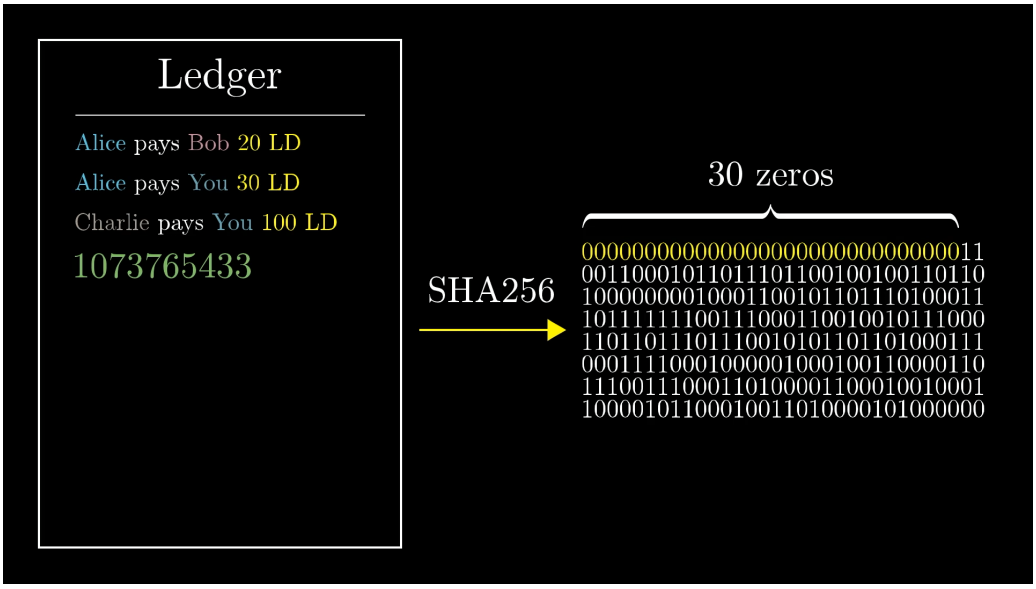


Figure II.7: There is no better way than guess and check for the special hash (Reference: [104]).

II.2.5 Blockchain

A distributed ledger system is composed of multiple nodes that broadcast transactions. To ensure consensus on the correct ledger, it is necessary to develop a mechanism that allows all nodes to agree on the validity of each transaction [105].

The core concept of the original Bitcoin paper [41] is based on the assumption that a distributed ledger will be trusted if it has been subject to a great deal of computational effort. This idea is implemented through the use of the many-zero game, which involves proving that a particular block in the chain contains a hash that is difficult to reverse-engineer.

Rather than hashing the entire ledger repeatedly, it is more efficient to allow for the accumulation of computational effort over time. Transactions are grouped into blocks and added to the chain in a linear fashion, with each new block containing a reference to the previous one. This approach allows for the creation of a tamper-evident history of transactions that is trusted by network participants due to the large amount of computational work required to manipulate it.

The block is a collection of transactions enclosed with a unique identifier, known as proof-of-work (PoW), which serves as evidence of the computational effort expended in validating the block. In PoW schemes, the miner must solve a complex mathematical problem to validate the block and add it to the blockchain. The difficulty level of this problem is determined by the number of leading zeros to be included in the hash value of the block.

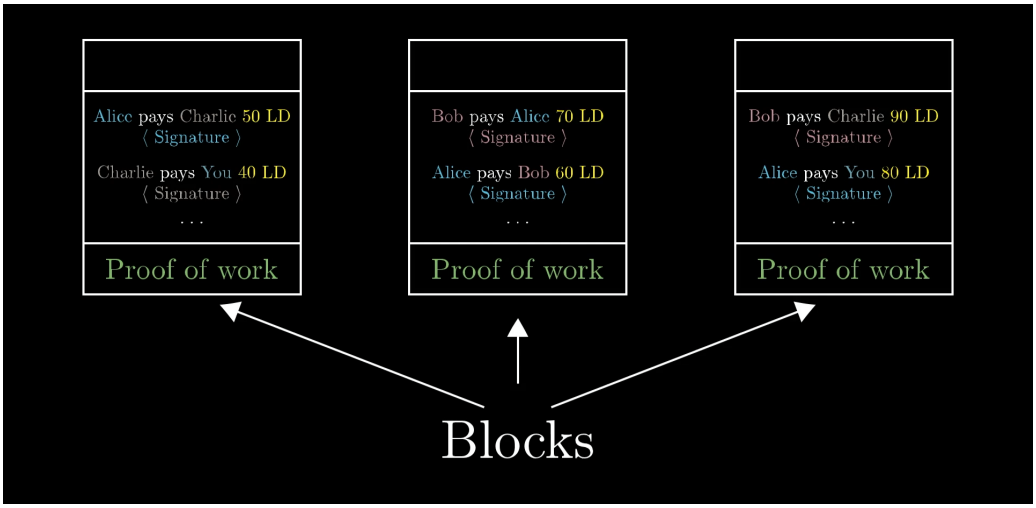


Figure II.8: Blocks on a blockchain (Reference: [106]).

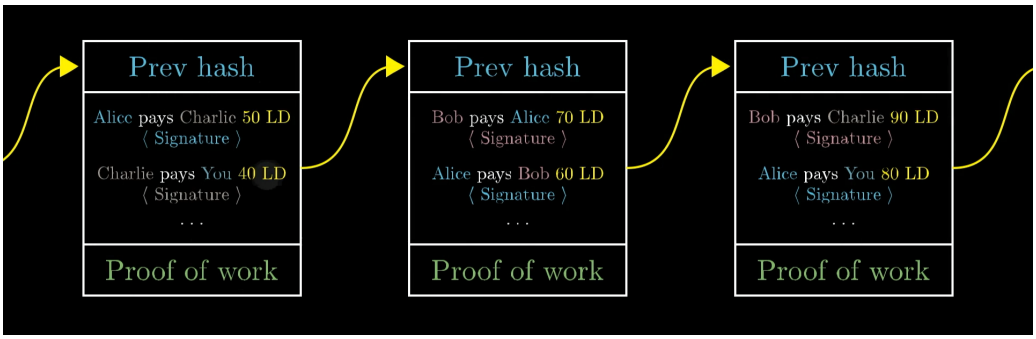


Figure II.9: Because blocks are chained together like this, instead of calling it a ledger, this is commonly called a “blockchain” (Reference: [107]).

A block is considered valid if it contains a proof-of-work (PoW) value, analogous to how a transaction is only considered valid when signed by its sender. Additionally, maintaining the integrity of the blockchain requires that blocks are not rearranged as this would disrupt the transaction history. To address this issue, each new block must begin with the hash of the previous block (hash-based chain), ensuring that the order of the blocks remains consistent.

Block Creators: Miners

To maintain the integrity of our ledger after it has been split into blocks, we have introduced a new process for adding new transactions. This involves grouping the transactions into blocks and computing a proof-of-work. As part of our updated protocol, anyone in the world is allowed to act as a "block creator". The responsibility of the block creator is to listen for broadcasted transactions, collect them into a block, and then perform a

significant amount of computational work to find a special number that will result in the hash of the block starting with 60 zeros. This computed hash value is then broadcasted to the network as proof of work [108].

A special transaction can be included at the beginning of each block, where the creator is rewarded with a predetermined amount of digital currency. This practice has been suggested as a means of compensating individuals for their efforts in building blocks within a distributed ledger system [109].

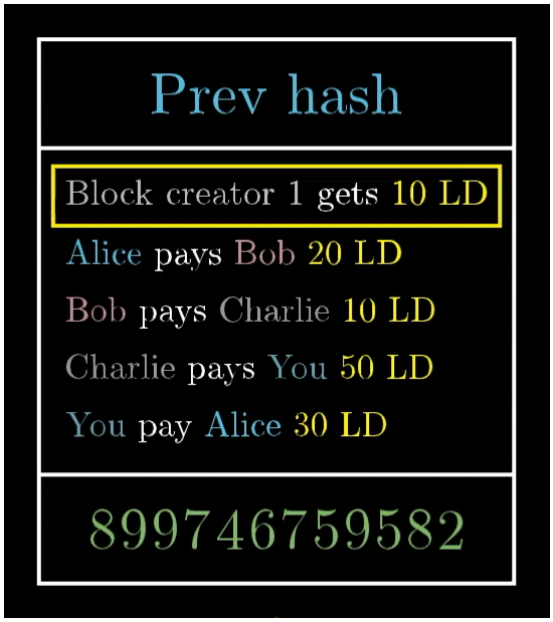


Figure II.10: Block reward (Reference: [110]).

The block reward is a unique exception to our usual transaction acceptance rules in the Ledger Dollar economy, as it does not require signature verification and increases the total number of currency units with each new block.

The process of creating blocks, known as "mining", involves a significant amount of work and introduces new currency into the economy. However, when discussing miners, it is essential to understand that they are primarily focused on listening to transactions, constructing blocks, broadcasting them, and receiving newly minted currency as a reward for their efforts.

For miners, each block can be thought of as a miniature lottery where individuals guess numbers rapidly until one person finds a combination that results in a hash starting with many zeros, earning the resulting reward. In contrast to mining, non-mining Bitcoin users no longer need to record all individual transactions on their personal ledger. Instead, they can simply monitor block production and rely on the fact that these blocks contain veri-

fied transactions. This approach is more manageable than maintaining a comprehensive transaction ledger.

In the consensus algorithm used by Bitcoin and other cryptocurrencies, a mechanism known as the "longest chain rule" is employed to resolve potential conflicts between competing blocks. Specifically, if two miners broadcast different blockchains with conflicting transaction histories, the system is referred to the one that has been the longest in terms of cumulative proof-of-work effort spent, which is assumed to be more resistant to manipulation [111]. If there is a tie between two competing blocks, it may be necessary to wait for additional information to determine which block is longer. This process relies on the assumption that the longest chain represents the most widely accepted version of the blockchain. However, this approach has been subject to criticism due to its reliance on proof-of-work mechanisms, which require significant computational effort and can lead to centralization.

Attempt Fraud On The Blockchain

To evaluate the trustworthiness of this method, it is instructive to consider what steps an individual, such as Alice, would need to take in order to deceive the system. In particular, suppose that Alice desires to purchase an item from Bob for 100 Ledger Dollars (LD), but does not actually possess those LDs. She might attempt to send a block to Bob containing a line indicating "Alice pays Bob 100 LD" without broadcasting this block to the broader network. By doing so, Bob would believe that he had been paid and provide Alice with the item she desires. However, at a later time, Alice could reenter the economy and spend those same 100 LD elsewhere. When Bob attempts to spend those same 100 LD, other individuals in the network may not recognize them as valid, leading to the potential for deception to be detected.

The process of creating a fraudulent transaction in a blockchain network requires a valid proof-of-work (PoW) that is found before other miners who are listening to the same set of transactions as the attacker, each working on their own block. This is a difficult task, but can be accomplished if the attacker has a significant portion of the network's computation power. If Alice is able to find the PoW before other miners, she can create a fraudulent transaction and present it to Bob (but not to anyone else) [112].

However, Bob will continue to receive broadcasts from other miners, and Alice did not inform these miners about the block she produced for Bob. Therefore, they will not include this block in their own versions of the blockchain. As a result, Bob will hear conflicting chains: one from Alice and another from everyone else [113]. According to the protocol, Bob always trusts the longest chain he knows about, which may create challenges for detecting and resolving fraudulent transactions in the network.

The probability of Alice's computational resources being smaller than the combined computational resources of the rest of the network is high, and as a result it is more likely for the rest of the network to find a valid proof of work for their next block before she does. Furthermore, if Alice has less than 50% of the total computation on the network (which is highly probable), she will outpace everyone else indefinitely, and it will be nearly impossible [41].

Eventually, when Alice fails to maintain her chain longer than the rest of the network, Bob will reject what he is hearing from Alice and follow the longer chain that everyone else is working on. This is because creating blocks requires significant computational effort, making it extremely difficult for any individual or group to manipulate the consensus [114].

It is worth noting that while building a single fraudulent block may be possible, maintaining the lie for an extended period is challenging. Therefore, users should exercise caution and wait for several new blocks to be added to a newly discovered block before trusting it as part of the main chain. By doing so, they can ensure that a malicious actor is not tricking them by trying to manipulate the network [115].

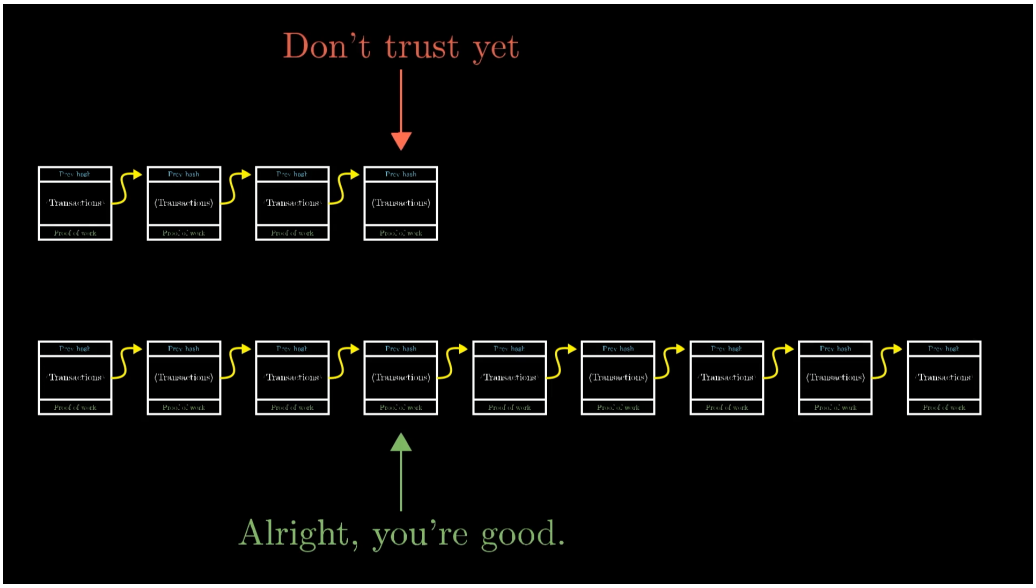


Figure II.11: Blocks are most trustworthy when they aren't brand new (Reference: [116]).

Ledger Dollars vs. Bitcoin

The distributed ledger system based on proof-of-work, as demonstrated by Bitcoin and other cryptocurrencies, involves a mining process where miners compete to solve a computational puzzle in order to validate transactions and add them to the blockchain. This is accomplished through the use of hash functions, which are designed to be difficult to

reverse engineer, thus ensuring the integrity of the distributed ledger [117]. The proof-of-work challenge may involve finding a special number that will make the hash of the block start with 60 zeros. However, in practice, this is achieved by systematically changing the number of zeros so that miners only need approximately 10 minutes to find a new block [41].

As a result of this process, a block reward is awarded to the miner who successfully validates a block. Initially, the reward was set at 50 Bitcoin per block, but has since been reduced to 6.25 Bitcoin per block every 210,000 blocks [41]. However, miners can also earn transaction fees by including them in the transaction validation process.

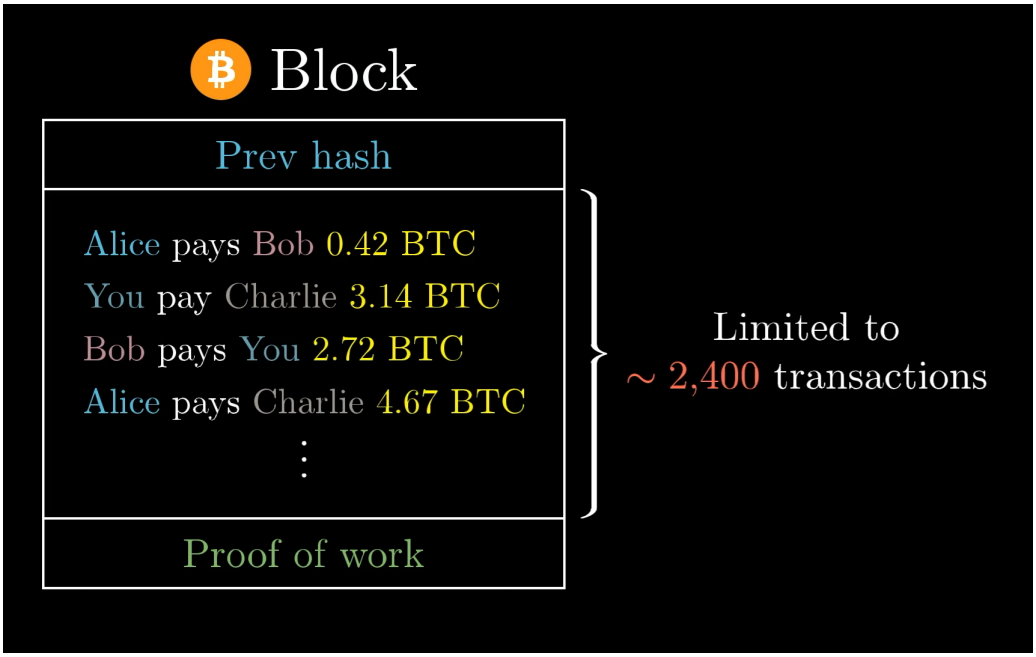


Figure II.12: Transactions on a bitcoin blockchain is limited (Reference: [118]).

Considering Bitcoin's objective of approximately one block addition per 10 minutes, its processing capacity is constrained to about 4 Bitcoin transactions per second, with some variability. By comparison, Visa handles an average of approximately 1,700 transactions per second, with the capability to process more than 24,000 transactions per second. The relatively slower processing speed of Bitcoin leads to higher transaction fees as they determine the selection of transactions included in new blocks by miners. Moreover, Bitcoin has faced criticism for its significant energy consumption. Although the proof-of-work concept effectively combats fraud, it requires a huge allocation of resources for block mining.

According to the Cambridge Bitcoin Electricity Consumption Index, the present annual electricity consumption for Bitcoin mining (as of 2021) is estimated at around 115

Terrawatt Hours. To provide context, this consumption exceeds the energy usage of the entire country of Finland. Since 2008, an alternative approach to proof-of-work, known as "proof of stake," has emerged, offering a substantial reduction in energy requirements. Several newer cryptocurrencies have adopted this methodology [119].

Supplement III

Replication

III.1 Replication of Baseline Anomaly Detection Methods

As a foundational step for our research, we first replicated key experiments from the work in [4]. This served two primary purposes: first, to validate our experimental setup and ensure our environment could reproduce established results on the Elliptic dataset; and second, to establish a performance baseline against which our proposed Energy Flow Classifier (EFC) approach could be compared. Lorenz et al. investigated several anomaly detection algorithms to identify illicit transactions under varying levels of data contamination in the training set, that is, the proportion of illicit (anomalous) samples mixed in with licit (normal) data.

We replicated their anomaly detection benchmark, focusing on several anomaly detection algorithms and a supervised baseline. Performance was measured using the F1-score for the illicit class, which is a suitable metric for imbalanced datasets. The results of our replication for key models are summarized in Table III.1.

Table III.1: Replicated Illicit F1-Scores by Contamination Level (based on [4]).

Contamination	IF	LOF	OC-SVM	Supervised Baseline
0.05	0.00	0.11	0.01	0.81
0.10	0.00	0.15	0.03	0.59
0.15	0.00	0.19	0.03	0.46
0.20	0.01	0.18	0.04	0.38

Note: The supervised baseline is a Random Forest classifier. IF stands for Isolation Forest, LOF for Local Outlier Factor, and OC-SVM for One-Class Support Vector Machine.

These replication results confirm the findings of Lorenz et al., demonstrating the performance of standard anomaly detection techniques on this task. More importantly, they provide a clear benchmark for evaluating the effectiveness of the EFC, which is the main focus of this dissertation. The subsequent chapters will detail our experiments with EFC and compare its performance with these established baselines.