Exploring Energy Flow Classifier to Identify Fraudulent Cryptocurrency Transactions

anonymous

[1]anonymous

Abstract. *This is a work in progress.*

1. Introduction

Bitcoin is an electronic transaction system operating without a third-party moderator [Nakamoto 2008]. It is built upon blockchain technology, where an immutable ledger of financial transactions is maintained through mathematics, programming, and advanced cryptography. This distributed ledger architecture eliminates the need for central authorities to establish trust. Although Bitcoin was designed to circumvent vulnerabilities in the traditional financial system [Nakamoto 2008], it is not immune to manipulation and anomalous activities, necessitating robust detection mechanisms [Fang et al. 2022, Zhang et al. 2020, Zainal et al. 2018].

Indeed, cryptocurrency-related fraud has emerged as a significant threat, causing substantial financial losses and shaking trust in the digital asset ecosystem. In 2023, for instance, illicit addresses received $24.2 billion in cryptocurrency, indicating the scale of financial losses from scams, stolen funds, and other illicit activities [Chainalysis 2024]. 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 the sustainable growth of the cryptocurrency industry [Scharfman 2024, Khiari et al. 2025].

However, 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 dimensionality, evolving characteristics, and a substantial volume, which complicates the application of traditional anomaly detection methods. In this context, the Energy-based Flow Classifier (EFC) presents a promising approach rooted in statistical physics. Originally formulated using the Inverse Potts model [Pontes et al. 2019], the EFC characterizes the probability distribution of normal data flows through an energy function derived from observed data patterns [Pontes et al. 2019]. Previous research has demonstrated the utility of EFC in classifying unusual network traffic, suggesting its potential for adapting to detect fraudulent activity within cryptocurrency systems [Pontes et al. 2019, Souza et al. 2022].

Building upon the promise of the Energy-based Flow Classifier (EFC) framework, this paper presents a comprehensive empirical evaluation of its application to detecting illicit Bitcoin transactions. To this end, we first replicate a previous study that employs machine learning algorithms such as K-Nearest Neighbors, One-Class Support Vector Machine, and Isolation Forest for anomaly detection on the Elliptic dataset [1]. We then

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

investigate the use of EFC as a potential alternative to these machine learning approaches, using the same dataset for consistency. Our findings confirm the EFC's ability to distinguish between licit and illicit transaction patterns based on their energy profiles, showing strong performance in identifying illegal activity even when trained solely on licit data. However, the results also highlight the critical sensitivity to specific configuration parameters. In particular, we observe significant trade-offs between maximizing the detection rate of illicit transactions (recall) and minimizing false positives (precision), especially concerning the energy threshold that defines anomalous behavior. Addressing the significant challenge of label scarcity inherent in datasets like Elliptic is crucial for 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. The Energy-based Flow Classifier (EFC) emerges as a promising candidate in this regard. Originally proposed for network intrusion detection [Pontes et al. 2019, Souza et al. 2022], EFC was designed specifically to address key limitations of conventional ML classifiers, including the reliance on extensive labeled datasets. A core strength highlighted in its foundational work is its ability to function as an anomaly-based classifier, inferring a statistical model of normal behavior using only labeled *benign* (or licit, in our context) examples [Souza et al. 2022]. Deviations from this learned norm, characterized by higher 'energy' scores, are then flagged as potential anomalies. This one-class learning paradigm directly tackles the label scarcity issue prevalent in the Elliptic dataset, allowing us to model legitimate transaction patterns effectively even with few confirmed illicit instances. Furthermore, EFC's demonstrated adaptability across different data distributions in network traffic analysis suggests potential robustness in the dynamic environment of cryptocurrency transactions. Consequently, this paper evaluates the suitability and performance of EFC for identifying illicit Bitcoin transactions by leveraging its capacity to model normality from available licit data.

In summary, the main contributions of this paper are:

- Novel Application and Empirical Evaluation of EFC for One-Class Bitcoin Anomaly Detection;
- In-depth analysis of the impact of data balancing and feature selection techniques on EFC's efficacy for identifying illicit transactions;
- Demonstration of a combined feature selection and data balancing (SMOTE) strategy as a robust approach for enhancing EFC performance on imbalanced cryptocurrency datasets.

2. Background and Related Work

Financial market manipulation, traditionally associated with actions taken by state-level entities often concerning currency exchange rates [Domanski and Sushko 2011, Khodabandehlou and Alireza Hashemi Golpayegani 2022], takes on a different character in the realm of cryptocurrencies. Cryptocurrency manipulation typically involves deceptive strategies employed by individuals or coordinated groups aiming to artificially distort market prices or activity, usually for illicit profit [Eigelshoven et al. 2021]. Understanding these tactics is crucial for developing effective detection mechanisms.

Among the prevalent forms of cryptocurrency fraud are pump-and-dump schemes and wash trading. **Pump-and-dump fraud** involves orchestrating an artificial inflation of a cryptocurrency's price ("pump") through misleading promotions or coordinated buying, attracting unsuspecting investors. The manipulators then sell their holdings ("dump") at the peak price, causing a market crash and significant losses for later investors [Karim and Mikhael 2018]. **Wash trading**, conversely, creates a false impression of high trading volume and liquidity. This is achieved when an entity or colluding group simultaneously buys and sells the same asset, often using multiple accounts or automated bots, effectively trading with themselves. The goal is to make the asset appear more active and desirable than it is, thereby manipulating market sentiment and potentially influencing its price [Gandal et al. 2018, Edelman et al. 2018]. The decentralized and pseudonymous nature of many cryptocurrency platforms can exacerbate the challenges in detecting and preventing such manipulations.

Addressing the challenge of identifying anomalous patterns within complex data streams, such as financial transactions, requires robust methodologies. One promising approach, grounded in statistical physics, is the **Energy-based Flow Classifier (EFC)**. Originally proposed using the Inverse Potts model to analyze network traffic data [Pontes et al. 2019], EFC operates on the principle of assigning an "energy" score to data points or flows. The model is trained to learn the probability distribution of normal system behavior, associating typical, expected patterns (e.g., legitimate transactions) with low-energy states. Conversely, configurations that deviate significantly from this learned normality, potentially representing anomalies or malicious activities, manifest as high-energy states. This energy score provides a quantitative measure of typicality. Subsequent research has extended the EFC framework, demonstrating its utility in classifying unusual network traffic for intrusion detection and highlighting its potential for open-set recognition—identifying novel anomalies not seen during training [Pontes et al. 2019, Souza et al. 2022].

The EFC framework appears particularly well-suited for detecting anomalous Bitcoin transactions for several key reasons. Firstly, its fundamental design as an anomaly detector allows it to learn the characteristics of *normal* (Licit) behavior, often from unlabeled or predominantly normal data. It then identifies anomalies (potentially Illicit transactions) as deviations exhibiting high 'energy' relative to this learned norm. This inherent capability to operate in a one-class or unsupervised manner directly addresses the significant challenge of label scarcity common in financial fraud datasets like Elliptic [Bansal et al. 2022]. Secondly, the energy function provides a holistic measure derived from the interplay of multiple features, potentially capturing complex, subtle deviations in the high-dimensional feature space of Bitcoin transactions that simpler methods might miss [Wilson and Anwar 2024]. Given its prior success in analogous domains involving complex flow data [Pontes et al. 2019, Souza et al. 2022], we hypothesize that EFC's unique characteristics offer advantages for identifying illicit activities within the Bitcoin blockchain.

Detecting illicit transactions in cryptocurrencies is fundamentally an anomaly detection task. Various approaches have been explored, ranging from rule-based systems identifying known patterns to sophisticated machine learning techniques [Samariya and Thakkar 2023, Li et al. 2023]. Traditional anomaly detection methods often face challenges when applied to cryptocurrency data due to its high dimensionality, the dynamic and evolving nature of transaction patterns, significant class imbalance (illicit transactions being rare), and the sheer volume of data [Pallathadka et al. 2022]. Techniques like statistical process control, clustering-based methods (e.g., DBSCAN, k-means variations), and distance-based methods (e.g., k-Nearest Neighbors anomaly detection) have been applied, each with varying degrees of success and limitations, particularly concerning scalability and the ability to capture complex, non-linear relationships [Hilal et al. 2022].

Building on these general principles, numerous machine learning models have been specifically investigated for identifying fraud and money laundering within the Bitcoin ecosystem. Supervised learning approaches, such as Random Forests, Support Vector Machines (SVM), Multilayer Perceptrons (MLP), and Gradient Boosting Machines, have shown promise when sufficient labeled data is available [Lorenz et al. 2021, Chen et al. 2021]. However, their performance heavily relies on the quality and quantity of labeled examples, which is often a bottleneck. Consequently, semi-supervised and unsupervised methods, including One-Class SVM (OC-SVM) and Isolation Forests, have gained attention as they can leverage unlabeled data or focus solely on modeling normal behavior [Lorenz et al. 2021, Kehinde et al. 2024]. More recently, Graph Neural Networks (GNNs) have emerged as a powerful tool, explicitly leveraging the graph structure of the blockchain to capture relational information between transactions, which can be crucial for identifying complex illicit schemes [Weber et al. 2019]. Our work contributes to this landscape by evaluating EFC, an alternative physics-inspired one-class approach, assessing its performance against the backdrop of these established techniques, particularly focusing on its behavior under label scarcity.

3. Study Settings

This section details the data and methods employed in our study, which builds upon the foundational research presented by [Lorenz et al. 2021]. 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 frameworks requiring at least some labels. Our research diverges by investigating the Energy Flow Classifier (EFC).

3.1. Dataset Description

This study utilizes the Elliptic dataset, a publicly available graph dataset of Bitcoin transactions introduced by [Weber et al. 2019] and subsequently used in foundational studies on machine learning for Bitcoin money laundering detection, including the work by [Lorenz et al. 2021] which highlighted the challenges of label scarcity. 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 full 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. One feature explicitly denotes the time step (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 relational patterns crucial 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. This represents a central challenge addressed by [Lorenz et al. 2021] and serves as a motivation for exploring EFC in this domain. The labels classify transactions into two main categories:

Licit Transactions: Transactions associated with known legitimate entities such as exchanges, miners, wallet providers, and other regulated services (42,019 instances in the original labeled set).

Illicit Transaction: Transactions linked to known illicit activities, including scams, ransomware, terrorist financing, Ponzi schemes, and dark market operations (4,545 instances in the original labeled set).

Table 1. Summary Statistics of the Elliptic Dataset (based on [Weber et al. 2019]).

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%)

3.2. Data Preprocessing

Preparing the Elliptic dataset for EFC involved several key steps focused on handling labels, selecting relevant features, scaling the data appropriately, and partitioning it for training and evaluation in a temporally meaningful way.

To prepare the dataset (Section 3.1), we filtered transactions based on their assigned labels. Since the EFC model learns patterns from *normal* data to detect anomalies, transactions labeled as *Licit* were used as the normal class for training. In contrast, transactions labeled as *Illicit* were treated as anomalies and reserved for evaluation. Transactions with *Unknown* labels were excluded from both training and testing to ensure that the evaluation relied solely on transactions with a known ground truth. For the binary classification task (*licit* vs. *illicit*), labels were mapped to numerical values—0 for licit and 1 for illicit.

Second, we 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. While this temporal information was essential for partitioning the data, it was excluded from the EFC model's input, as the model focuses on intrinsic transaction properties rather than 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 [Weber et al. 2019], we decided to apply the Min-Max scaling to the [0, 1] range to ensure consistency and enhance the stability of energy calculations within the EFC framework. Scaling was applied separately to the training and test sets, with the scaler fitted exclusively on the training data to prevent data leakage.

Third, we implemented a temporal data split, in line with common practice for this dataset [Weber et al. 2019, Lorenz et al. 2021], to simulate a realistic scenario in which a model trained on historical data is used to detect fraudulent activity in future transactions. Transactions from time steps 1 to 34 were allocated to the training set, while those from time steps 35 to 49 were reserved for testing. The EFC model was trained exclusively on licit (normal) transactions 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 assign higher energy scores to previously unseen illicit transactions compared to unseen licit ones.

3.3. Energy Flow Classifier Configuration

For detecting illicit transactions within the Elliptic dataset, we employed the Energy Flow Classifier (EFC), leveraging the Python package implementation [efc 2021] based on the recommendations by Pontes et al. [Pontes et al. 2019, Souza et al. 2022]. EFC operates on the premise that normal system behavior corresponds to low-energy states, while anomalies or deviations manifest as high-energy states. Our implementation specifically utilizes EFC as a one-class anomaly detector, tailored to the label scarcity challenge inherent in the dataset.

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 configured three key EFC hyperparameters: `n_bins`, `cutoff_quantile`, and `pseudocounts`. The `n_bins` parameter controls the discretization of input features, dividing each feature's range into a specified number of bins. These bins form the basis for estimating state probabilities and computing energy values. 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 instance, 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 energy computation.

Regarding the *training process*, a central aspect of our EFC configuration is its alignment with the anomaly detection setting under label scarcity. As outlined in Section 3.2, the EFC model was trained exclusively on licit (normal) transactions from the training period (time steps 1–34) by invoking its `fit` method. This follows the one-class classification paradigm, in which the model learns the energy landscape associated with legitimate Bitcoin transactions based solely on verified normal examples. Illicit transactions were entirely excluded from the training phase to preserve the model's ability to generalize and detect anomalies without prior exposure to them.

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 computed an energy score based on its features and the probability distributions learned during training. If a transaction's energy score exceeded the pre-determined 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). The energy scores were also used to compute evaluation metrics such as AUC, which assess ranking performance rather than relying on a fixed classification threshold.

4. Goal, Questions, and Metrics

The primary goal of this study is to evaluate the effectiveness of the Energy Flow Classifier (EFC), configured as described in Section 3.3, for identifying illicit transactions within the Elliptic Bitcoin dataset. This aligns with the broader goal of exploring alternative methodologies, particularly those suited for label scarcity, compared to the supervised approaches examined by [Lorenz et al. 2021]. Our aim is to answer the following research question: *How effective is the Energy-Based Flow Classifier (EFC) under conditions of label scarcity in identifying illicit transactions in the Elliptic Bitcoin dataset?*

Specifically, our research is thus framed as a *one-class anomaly detection problem*. Having trained the EFC model exclusively on Licit transactions from the initial time steps (1-34), the objective is to assess its ability to distinguish between licit and illicit transactions in the subsequent, unseen time steps (35-49) of the test set. This evaluation involves two main perspectives:

1. **Classification Performance:** Using the energy threshold derived from the training data, based on the `cutoff_quantile`, we assess how well EFC classifies unseen transactions as either licit (below threshold) or illicit (above threshold). Performance is measured using standard classification metrics suitable for imbalanced datasets, such as Precision, Recall, F1-Score, and potentially Balanced Accuracy, calculated on the labeled test set.
2. **Ranking Performance:** Independent of a specific threshold, we evaluate EFC's ability to assign consistently higher energy scores to illicit transactions compared to licit transactions in the test set. This is primarily assessed using the Area Under the Receiver Operating Characteristic Curve (AUC-ROC), which measures the model's ability to rank anomalies higher than normal instances across all possible thresholds.

The outcomes of this research provide insights into the potential of the EFC model as a viable tool for detecting fraudulent or anomalous Bitcoin transactions. By leveraging its unsupervised, energy-based approach, EFC aims to address challenges such as label scarcity and to identify novel deviations from normal transaction behavior.

Model performance was assessed using a combination of quantitative metrics and visual analysis. We use the **F1-Score Macro Average** as the primary evaluation metric, following the same design decision of [Lorenz et al. 2021]. 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 leverage **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 specifically calculated 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.** For each experiment we detail in the next section, we show histograms comparing the distribution of EFC energy scores assigned to Licit versus Illicit transactions in the test set that were generated. These plots provide a visual assessment of the model's separation capability.

5. Results

This section presents the empirical findings from the application of the Energy Flow Classifier (EFC) to the task of identifying illicit transactions within the Elliptic Bitcoin dataset. Following the methodology outlined in Section 3, the EFC was employed primarily as a one-class anomaly detector, trained exclusively on transactions labeled as Licit from the initial time steps (1-34). The core objective was to evaluate the model's capability to distinguish these known Licit patterns from potentially anomalous Illicit transactions present in the unseen test set (time steps 35-49). Performance is assessed based on the EFC's ability to assign distinct energy scores to the two classes and evaluated using metrics appropriate for imbalanced anomaly detection scenarios. The subsequent subsections detail the outcomes of specific experiments conducted, focusing on the model's baseline performance and sensitivity to key configuration parameters under the defined experimental setup. We performed three primary experiments focusing on data balancing, feature engineering/selection, and model comparison/tuning, respectively. All experiments utilized the Elliptical dataset, preprocessed as described previously, employing a standard train-test split methodology.

5.1. 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.

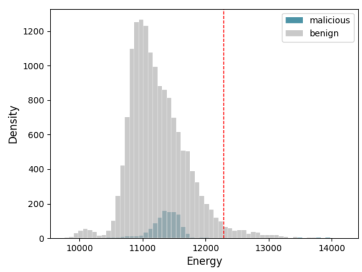
Baseline performance was measured using the original, unbalanced dataset. This baseline was then compared against four widely adopted balancing techniques, each applied to both the training and test datasets: (a) creating a balanced subset by undersampling the majority class prior to the train-test split, (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 across most techniques.

Model performance was evaluated using Accuracy, Precision, Recall, weighted F1-Score, Macro F1-Score, and confusion matrices. Table **??** presents a summary of the dataset characteristics resulting from each balancing strategy, along with the corresponding classification outcomes.

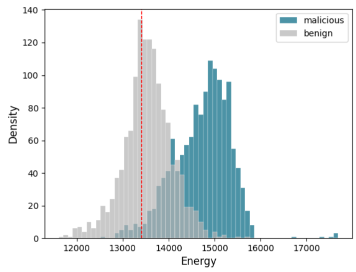
Table 2. EFC Performance Across Data Balancing Techniques (Experiment 1).

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

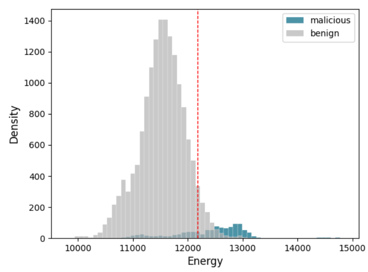
Note: TP=True Positives, FN=False Negatives, FP=False Positives, TN=True Negatives. Metrics are rounded. F1-Score is weighted average. F1-Macro for SMOTE is bolded as it's the highest among these techniques. Test set composition: (^{a}) Imbalanced, (^{b}) Balanced.



(a) Unbalanced Dataset.



(b) Balanced Dataset.



(c) SMOTE.

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

The results from Experiment 1 (Table 2) 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 a striking improvement, with an F1-Macro of 0.908. 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. While 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.

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

5.2. Experiment 2: Impact of Feature Selection

Following the analysis of data balancing, our second experiment focused on evaluating the impact of feature selection on classification performance using the Energy-Based Flow Classifier (EFC). We employed the `SelectKBest` algorithm from scikit-learn, 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 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 reduced 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 features.

Aggregated features, which represent statistical summaries of a node's neighborhood (as described in related financial forensics work, e.g., [Weber et al. 2019]), 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` process (Scenario 1) could lead to them consistently ranking highest, potentially masking the predictive contribution of the node's intrinsic, 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 feature set scenarios (with and without aggregated features) 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 performance, identify an optimal number of features (k), and understand the contribution of aggregated features within this selection context. We summarize the results in Table 3 Table 4.

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 3), 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), the peak F1-Macro was 0.689, also at $k = 10$.

This suggests that a small subset of the most relevant features is sufficient for EFC, and including more features beyond this optimal k (generally $k > 20$) tends to degrade performance, likely due to the introduction of noise or less informative features that can adversely affect EFC's energy calculations. This diminishing return with increasing

Table 3. EFC Performance with Feature Selection (Aggregated Features Excluded) for Varying k (Experiment 2a).

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. EFC Performance with Feature Selection (Aggregated Features Included) for Varying k (Experiment 2b).

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.

k is a common phenomenon in feature selection. The slightly higher F1-Macro obtained when aggregated features were included (0.689 vs. 0.686) indicates their strong predictive value, even when only a few are selected. These F1-Macro scores represent an improvement over the baseline (0.488 from Experiment 1) but are not as high as those achieved with data balancing techniques like Random Undersampling (0.652 on an imbalanced test set) or SMOTE (0.908 on a balanced test set).

This implies that while 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 this dataset. The results confirm that feature selection can be effective, but might not be a complete solution without tackling imbalance.

5.3. 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 upon the findings of Experiment 1 (data balancing) and Experiment 2 (feature selection). The central idea is to first reduce the dataset's dimensionality 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 this balanced, feature-reduced training data, and evaluated on the corresponding unbalanced test set (which contained the same k selected features).

We evaluated the performance using standard classification metrics (again, Accuracy, Precision, Recall, F1-Score, and Macro F1-Score). This evaluation aimed to determine whether applying 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 5 (Experiment 3a, standard unbalanced test set) and Table 6 (Experiment 3b, "Full Test Dataset" context, also an unbalanced test set).

In Experiment 3a, the combination yielded a peak F1-Macro score of 0.808 when $k = 30$ features were selected before applying SMOTE. This is a substantial improvement compared to using 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—dimensionality reduction to focus on salient features and SMOTE to address imbalance—leads to an improvement on the overall performance. The optimal $k = 30$ here is higher than the $k = 10$ found in Experiment 2 (FS alone), though, suggesting that SMOTE might benefit from a slightly richer, yet still reduced, feature set to generate more effective synthetic samples.

Experiment 3b, conducted under the "Full Test Dataset" context (which also utilized an imbalanced test set as per its note), showed a similar trend, with $k = 30$ also yielding the best F1-Macro score of 0.770. While this is still a strong result and a significant improvement over the baseline and feature selection alone, it is slightly lower than the 0.808 achieved in Experiment 3a. This minor 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). However, these scores still do not reach the F1-Macro of 0.908 achieved by SMOTE alone when evaluated on an ideally balanced test set (Experiment 1), further highlighting the profound impact of the evaluation scenario's balance on the F1-Macro metric.

Table 5. EFC Performance: SMOTE with Feature Selection for Varying k (Experiment 3a).

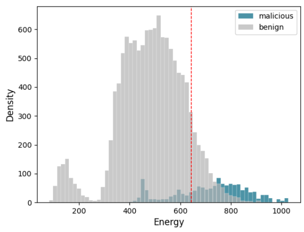
k Value	TP	FN	FP	TN	Accuracy	Precision	Recall	F1-Score (Weighted)	F1-Macro
10	11319	1287	369	995	0.891	0.931	0.891	0.891	0.770
20	11326	1280	263	1101	0.900	0.939	0.900	0.900	0.798
30	11331	1275	226	1138	0.903	0.942	0.903	0.903	**0.808**
40	11272	1334	223	1141	0.900	0.940	0.900	0.900	0.799
50	11233	1373	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.

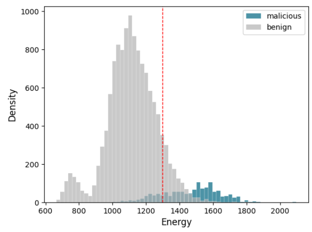
Table 6. EFC Performance: SMOTE with Feature Selection (Full Test Dataset Context) for Varying k (Experiment 3b).

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

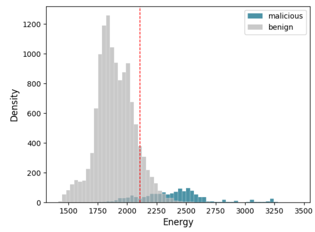
Note: SMOTE applied to training data after feature selection (including aggregated features). Test set is unbalanced (1364 Malicious, 12606 Benign). TP=True Positives, FN=False Negatives, FP=False Positives, TN=True Negatives. Metrics rounded. F1-Score is weighted average. F1-Macro for k=30 is bolded.



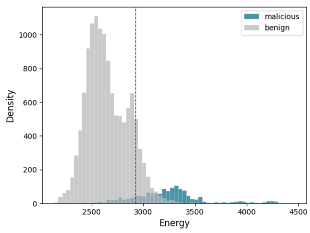
(a) k=10.



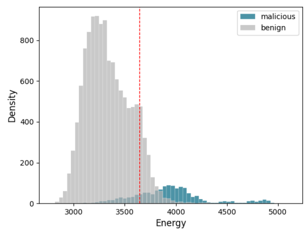
(b) k=20.



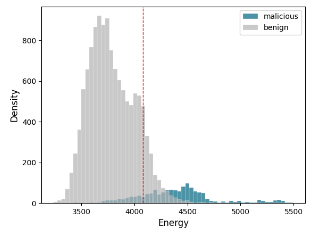
(c) k=30.



(d) k=40.

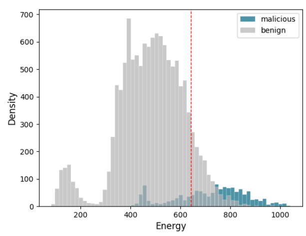


(e) k=50.

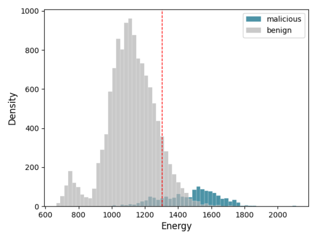


(f) k=60.

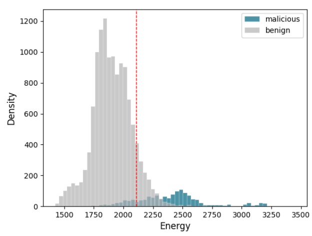
Figure 2. Experiment 3, Technique A: SMOTE With Feature Selection, Increasing Value of k, Energy Distribution Of Licit and Illicit Transactions.



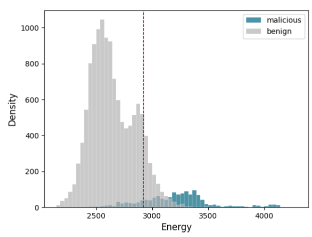
(a) k=10.



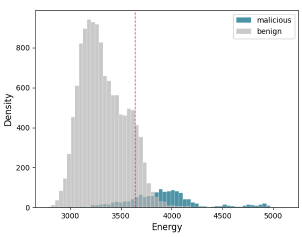
(b) k=20.



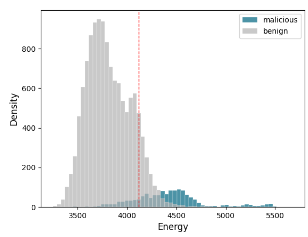
(c) k=30.



(d) k=40.



(e) k=50.



(f) k=60.

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

6. Discussion

In this section, we summarize the results of our three experiments by answering the research questions (Section 6.1), compare them with the results of a previous work that explored active learning on the Elliptic Dataset [**?**] (Section 6.2), and highlight some threats-to-validity that might compromise the generalization of our findings (Section 6.3).

6.1. Answers to the Research Questions

The primary research question guiding this study was: *How effective is the Energy-Based 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 solely on Licit transactions and applied to the raw, imbalanced dataset, EFC's ability to distinguish the minority Illicit class was limited, yielding a low F1-Macro score of 0.488 (Experiment 1, Baseline). This highlights that, in its basic one-class configuration, EFC struggles with severe imbalance.

However, the application of data balancing techniques, particularly SMOTE on the training data, significantly enhanced 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 regarding the optimal strategy for applying EFC:

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?** While SMOTE alone yielded 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 no 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 on imbalanced test data.

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

The findings confirm EFC's utility as a one-class classifier that can learn from licit-only data, a significant advantage under label scarcity. However, its practical success hinges on appropriate data preprocessing, particularly balancing and potentially feature selection, to effectively identify the rare illicit instances.

6.2. Comparison to Previous Work

This study builds upon and diverges from previous research on detecting illicit Bitcoin transactions, notably the work by Lorenz et al. [Lorenz et al. 2021], which also utilized the Elliptic dataset and addressed the challenge of label scarcity.

Lorenz et al. [Lorenz et al. 2021] 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:

- **One-Class Learning Paradigm:** EFC was configured as a one-class anomaly detector, trained *exclusively* on Licit transactions. This directly addresses label scarcity by not requiring any Illicit labels during the training phase, learning a model of "normal" behavior and identifying deviations. This contrasts with the supervised/semi-supervised methods in [Lorenz et al. 2021] 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 Lorenz et al.
- **Focus on Preprocessing for One-Class 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 [Lorenz et al. 2021], to facilitate a conceptual comparison regarding performance on imbalanced data.

While Lorenz et al. 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 is reliably labeled. The results, particularly with SMOTE and feature selection (F1-Macro 0.808 on imbalanced test data), suggest EFC can be competitive. A direct quantitative performance benchmark against the specific results of [Lorenz et al. 2021] 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.

6.3. Threats to Validity

Several factors could potentially affect the validity and generalizability of our findings. We categorize them as follows:

Internal Validity:

- **Hyperparameter Configuration:** The EFC model was configured with default hyperparameters (e.g., `n_bins=30`, `cutoff_quantile=0.9`). Performance, particularly the precision-recall trade-off, might be sensitive to these settings. A systematic hyperparameter optimization was not performed.
- **Stochasticity of SMOTE and Train-Test Split:** Techniques like SMOTE involve randomness. The specific temporal split (time steps 1-34 for training, 35-49 for testing), though standard, is one instance; other splits could yield different quantitative results.
- **Feature Selection Method:** We used `SelectKBest` with ANOVA F-value. Other techniques might lead to different feature subsets and performance.

Construct Validity:

- **EFC as a Proxy for Illicit Detection:** We measure EFC's ability to assign higher energy to "illicit" transactions as defined by the dataset labels. The "energy" concept is an abstraction.
- **Label Quality and Definition:** The study relies on the labels provided in the Elliptic dataset. Any noise or bias in labels would affect training and evaluation.
- **Feature Representation:** The Elliptic dataset provides anonymized features. Lack of semantic meaning limits interpretability and domain-specific feature engineering.

External Validity (Generalizability):

- **Dataset Specificity:** Findings are based solely on the Elliptic Bitcoin dataset. Generalization to other cryptocurrency transaction data or fraud domains requires further study.
- **Temporal Dynamics:** The dynamic nature of fraudulent activities means model performance might degrade over time. The study captures a specific period.
- **Ignoring Graph Structure (Partially):** While some features are aggregated from local graph neighborhoods, our EFC implementation primarily treated transactions based on node features without explicitly modeling the transaction graph topology in EFC's core calculation.
- **Scalability:** A formal scalability analysis for significantly larger datasets was not conducted.

Addressing these threats in future work will be crucial for a more comprehensive understanding of EFC's capabilities.

7. Conclusion

This study investigated the Energy Flow Classifier (EFC) as a one-class anomaly detection method for identifying illicit Bitcoin transactions within the Elliptic dataset, particularly addressing the challenge of label scarcity. Our findings indicate that EFC, when trained exclusively on Licit transactions, shows promise but is highly sensitive to the inherent class imbalance of such datasets.

The baseline performance of EFC on unbalanced data was modest in terms of distinguishing the minority Illicit class (F1-Macro 0.488). However, we demonstrated that its effectiveness can be significantly enhanced through data preprocessing. Specifically, applying the SMOTE balancing technique to the training data, especially when combined with feature selection (reducing to $k \approx 30$ features), substantially improved the F1-Macro score to 0.808 when evaluated on a realistic, imbalanced test set. This highlights that EFC can achieve robust performance if the data imbalance is appropriately managed, as detailed in Section 6.

In summary, EFC offers a valuable approach for one-class anomaly detection in cryptocurrency transactions due to its ability to learn from only normal data. Its practical application, however, requires careful consideration of data preprocessing strategies, notably class balancing and potentially feature selection, to optimize its detection capabilities for rare illicit events. Future work, as outlined in Section 8 and informed by the threats to validity discussed in Section 6.3, should focus on refining these aspects.

8. Future Work

While this study demonstrated the viability of the Energy Flow Classifier (EFC) for one-class anomaly detection in Bitcoin transactions, particularly when combined with techniques like SMOTE, several open questions and avenues for future research emerge:

- **Systematic Hyperparameter Optimization:** The current work used fixed default parameters (`n_bins=30`, `cutoff_quantile=0.9`). An open question is how sensitive EFC's performance, especially the precision-recall trade-off for the Illicit class, is to these parameters.
- **Advanced Balancing and Cost-Sensitive Learning:** SMOTE proved effective (Exp 1), but its interaction with EFC's energy calculation warrants further investigation. Are there other balancing techniques (e.g., ADASYN, Tomek Links, Edited Nearest Neighbours) or cost-sensitive learning approaches (adjusting the EFC threshold based on misclassification costs) that could yield better or more robust performance, perhaps with fewer synthetic samples or different biases?
- **Refining Feature Selection Strategies:** While Experiments 2 and 3 showed the benefit of feature selection, further exploration of different algorithms and criteria for selecting k could be beneficial. Key questions remain: What is the optimal number of features (k) when including aggregated features? Does the combination of FS and SMOTE truly outperform SMOTE alone with full features, considering both performance and computational cost?
- **Integration of Graph Structure:** The current EFC implementation primarily leverages node features, largely ignoring the rich topological information in the Elliptic graph dataset. Can explicitly incorporating transaction linkage improve detection?

- **Scalability Analysis:** How does the computational cost (training time, memory usage, prediction latency) of EFC scale with the number of transactions and features, especially compared to other methods?

Addressing these questions would further solidify the understanding of EFC's strengths and weaknesses in the financial forensics domain and guide its potential practical application.

9. Reproducibility

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

9.1. 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

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

Bansal, M. A., Sharma, D. R., and Kathuria, D. M. (2022). A systematic review on data scarcity problem in deep learning: solution and applications. *ACM Computing Surveys (Csur)*, 54(10s):1–29.

Chainalysis (2024). The 2024 crypto crime report. The latest trends in ransomware, scams, hacking, and more.

Chen, B., Wei, F., and Gu, C. (2021). Bitcoin theft detection based on supervised machine learning algorithms. *Security and Communication Networks*, 2021(1):6643763.

Domanski, D. and Sushko, V. (2011). Currency manipulation: The imf and wto. *BIS Quarterly Review*, September:79–91.

Edelman, B., Moore, T., and Oberman, T. (2018). Detecting pump and dump in cryptocurrency markets. *Journal of Economic Perspectives*, 32(2):81–102.

Eigelshoven, F., Ullrich, A., and Parry, D. (2021). Cryptocurrency market manipulation: A systematic literature review. In *42nd International Conference on Information Systems, ICIS 2021 TREOs: "Building Sustainability and Resilience with IS: A Call for Action"*. Association for Information Systems. 42nd International Conference on Information Systems: Building Sustainability and Resilience with IS: A Call for Action, ICIS 2021 TREOs ; Conference date: 12-12-2021 Through 15-12-2021.

Fang, F., Ventre, C., Basios, M., Kanthan, L., Martinez-Rego, D., Wu, F., and Li, L. (2022). Cryptocurrency trading: a comprehensive survey. *Financial Innovation*, 8(1):1–59.

Gandal, N., Hamrick, J., Moore, T., and Oberman, T. (2018). Price manipulation in the bitcoin ecosystem. In *Proceedings of the 2018 Conference on Economic and Financial Computing*, pages 1–7.

Hilal, W., Gadsden, S. A., and Yawney, J. (2022). Financial fraud: a review of anomaly detection techniques and recent advances. *Expert systems With applications*, 193:116429.

Karim, M. A. and Mikhael, S. (2018). Manipulation detection in cryptocurrency markets. *IEEE Access*, 6:11044–11054.

Kehinde, J., Ajayi, O. O., Adetayo, A., Obafemi, J. R., Akinrolabu, O. D., and Ebitigha, A. E. (2024). Machine learning model for detecting money laundering in bitcoin blockchain transactions. *Machine Learning*, 1(1).

Khiari, W., Lajmi, A., Neffati, A., and El Fahem, A. (2025). Cryptocurrency fraud and its effects on price volatility in the cryptocurrency market. *Journal of Chinese Economic and Foreign Trade Studies*.

Khodabandehlou, S. and Alireza Hashemi Golpayegani, S. (2022). Market manipulation detection: A systematic literature review. *Expert Systems with Applications*, 210:118330.

Li, Z., Zhu, Y., and Van Leeuwen, M. (2023). A survey on explainable anomaly detection. *ACM Transactions on Knowledge Discovery from Data*, 18(1):1–54.

Lorenz, J., Silva, M. I., Aparício, D., Ascensão, J. T., and Bizarro, P. (2021). Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. `https://bitcoin.org/bitcoin.pdf`.

Pallathadka, H., Tongkachok, K., Arbune, P. S., and Ray, S. (2022). Cryptocurrency and bitcoin: Future works, opportunities, and challenges. *ECS Transactions*, 107(1):16313.

Pontes, C. F. T., Gondim, J. J. C., Bishop, M., and Marotta, M. A. (2019). A new method for flow-based network intrusion detection using inverse statistical physics. *CoRR*, abs/1910.07266.

Samariya, D. and Thakkar, A. (2023). A comprehensive survey of anomaly detection algorithms. *Annals of Data Science*, 10(3):829–850.

Scharfman, J. (2024). *The Cryptocurrency and Digital Asset Fraud Casebook, Volume II: DeFi, NFTs, DAOs, Meme Coins, and Other Digital Asset Hacks*. Palgrave Macmillan.

Souza, M. M. C., Pontes, C., Gondim, J., Garcia, L. P. F., DaSilva, L., and Marotta, M. A. (2022). A novel open set energy-based flow classifier for network intrusion detection.

Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., and Leiserson, C. E. (2019). Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics.

Wilson, A. and Anwar, M. R. (2024). The future of adaptive machine learning algorithms in high-dimensional data processing. *International Transactions on Artificial Intelligence*, 3(1):97–107.

Zainal, A., Kamruzzaman, J., and Sarker, R. A. (2018). A review on machine learning techniques for the detection of financial statement fraud. *International Journal of Financial Studies*, 6(3):70.

Zhang, X., Wen, Y., Zhou, J., Zhang, W., and Liao, X. (2020). Financial fraud detection in cryptocurrency exchanges: A comprehensive survey. *IEEE Access*, 8:193150–193172.