Exploring Energy Flow Classifier to Identify Fraudulent Cryptocurrency Transactions

Kevin S. Araujo[1], **Rodrigo Bonifacio de Almeida**[1], **Fabiano Cavalcanti Fernandes**[2]

[1]Departamento de Ciências da Computação – Universidade de Brasília (UnB)
– Campus Universitário Darcy Ribeiro, Brasília-DF

[2]Instituto Federal de Brasília (IFB) – Taguating, DF – Brazil

kevin.araujo@aluno.unb.br, rbonifacio@unb.br, fabiano.fernandes@ifb.edu.br

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 [**?**, 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 [**?**]. 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 [**?**, **?**].

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 Neighbours, One-Class Support Vector Machine, and Isolation Forest for anomaly detection on the Elliptic dataset [1]. We then 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 a 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. In summary, the main contributions of this paper are:

- Novel Application and Empirical Evaluation of EFC for One-Class Bitcoin Anomaly Detection;
- c2
- c3

2. Background and Related Work

WIP

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.

Source and Scope: 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.

Features: 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

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

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 crucial contextual information about the flow of funds, although our EFC implementation primarily focuses on the node features.

Labels: A key characteristic of the Elliptic dataset, and a central challenge addressed by [Lorenz et al. 2021] and relevant to our EFC approach, is the presence of label scarcity. While the dataset consists of 203,769 transactions, only a subset is explicitly labeled. Based on the analysis by [Weber et al. 2019], 46,866 transactions or around 23% were initially labeled. These labels classify transactions into two main categories:

- **Licit:** Transactions associated with known legitimate entities such as exchanges, miners, wallet providers, and other regulated services – 42,791 instances in the original labeled set.
- **Illicit:** Transactions linked to known illicit activities, including scams, ransomware, terrorist financing, Ponzi schemes, and dark market operations – 4,075 instances in the original labeled set.

3.2. Data Preprocessing

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

Characteristic	Value
Total Transactions (Nodes)	203,769
Total 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%)

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.

First, addressing the labels described in Section 3.1, we filtered the transactions based on their classification. Given that EFC operates by learning the characteristics of *normal* data and identifying deviations, transactions labeled as *Licit* were designated as the normal class for model training. Transactions labeled as *Illicit* were designated as the anomalous class, primarily used during the evaluation phase to assess detection performance. The large portion of transactions with *Unknown* labels were excluded from both training and testing in this study to ensure a clear evaluation based on known ground truth. The binary nature of the task (Licit vs. Illicit) required mapping these labels to numerical values (e.g., 0 for Licit, 1 for Illicit) for evaluation purposes.

Second, feature selection and transformation were performed. From the 166 features available for each transaction, the feature explicitly indicating the time step (ranging from 1 to 49) was removed. This temporal information was crucial for partitioning the data but was not used as a direct input feature to the EFC model itself, as the model focuses on the intrinsic properties of the transaction rather than its absolute time position. The remaining 165 anonymized features, representing transactional and local graph properties, were retained as input for the EFC. Although the original dataset description mentions some form of normalization [Weber et al. 2019], to ensure consistency and potentially improve the stability of the energy calculations within the EFC framework, these 165 features were scaled to the range [0, 1] using Min-Max scaling. This scaling was applied separately to the training and test sets (fitting the scaler only on the training data) to prevent data leakage.

Third, a temporal data split was implemented, following common practice for this dataset [Weber et al. 2019, Lorenz et al. 2021] to simulate a realistic scenario where a model trained on past data is used to detect fraud in future transactions. Transactions belonging to time steps 1 through 34 were allocated to the training set. Transactions from the subsequent time steps, 35 through 49, constituted the test set.

Crucially, the EFC model was trained only on the Licit (normal) transactions within the training time steps (1-34). The test set (time steps 35-49) contained both Licit and Illicit transactions, allowing for the evaluation of EFC's ability to assign higher energy scores to the unseen illicit transactions compared to the unseen licit ones.

These preprocessing steps are summarized in Table 2.

Table 2. Summary of Data Preprocessing Steps for the Elliptic Dataset.

Step	Description
Label Handling	- Selected transactions labeled Licit (Normal) and Illicit (Anomaly). - Excluded transactions labeled Unknown. - Mapped Licit to 0, Illicit to 1 for evaluation.
Feature Selection	- Retained 165 anonymized transactional/local features. - Excluded the Time Step feature from model input.
Feature Scaling	- Applied Min-Max scaling to the 165 features, scaling to range [0, 1]. - Scaler fitted only on the training data.
Data Splitting	- Temporal split: Time steps 1-34 for Training, 35-49 for Testing. - Training Set Composition: Licit transactions only from steps 1-34. - Test Set Composition: Licit and Illicit transactions from steps 35-49.

3.3. Energy Flow Classifier (EFC) Implementation

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 principles described by Ponts 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.

Model Instantiation and Interface: We primarily utilized the class-based interface provided by the package, specifically the `EnergyBasedFlowClassifier` class.

As indicated in our experimental setup code [?], an EFC instance was configured with specific hyperparameters:

- `n_bins`: This parameter controls the discretization of the input features. Each feature's range is divided into `n_bins` intervals, forming the basis for calculating the system's state probabilities and energy. Based on our experiments and configuration [?], a value of `N_BINS = 30` was used.
- `cutoff_quantile`: This determines the energy threshold for classifying a sample as anomalous. After fitting the model on normal data, the energy distribution of this training data is computed. The threshold (`cutoff_`) is set at the energy value corresponding to the specified quantile e.g., `CUTOFF_QUANTILE = 0.9` in 'constants.py', meaning energies above the 90th percentile of training energies are considered potentially anomalous.
- `pseudocounts`: To avoid issues with zero probabilities when calculating energies, especially for states not observed in the training data, a small pseudocount is added. We used a value of `PSEUDOCOUNTS = 0.1` [?].

While some experimental scripts might show usage of the lower-level functional interface (`one_class_fit`, `one_class_predict`) provided by the EFC package, the core experiments relied on the `EnergyBasedFlowClassifier` class with the parameters defined above.

Training Process: A key aspect of our implementation, driven by the goal of anomaly detection under label scarcity, was the training procedure. Following the data split described in Section 3.2, the EFC model's `fit` method was called exclusively using the Licit (normal) transactions from the training time steps (1-34). This aligns with the one-class classification paradigm: the model learns the energy landscape characteristic of normal Bitcoin transactions based solely on examples known to be legitimate within the training period. The illicit transactions were entirely withheld during this phase.

Prediction Process: 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, EFC calculates an energy score based on its features and the learned probability distributions from the training phase. If a transaction's energy score exceeded the pre-determined `cutoff_` threshold (derived from the `cutoff_quantile` applied to the training data energies), it was classified as anomalous (predicted Illicit, label 1); otherwise, it was classified as normal (predicted Licit, label 0). The energy scores themselves (`predict_energies` method) were also used for evaluation metrics like AUC that rely on ranking rather than a hard classification threshold.

Comparison to Original EFC Package: Our usage adheres closely to the standard application of the EFC package for one-class classification/anomaly detection. We did not modify the core EFC algorithm: energy calculation, probability estimation. The primary customization lies in the specific application context:

1. **Strict One-Class Training:** Explicitly training only on the Licit subset of the temporally defined training data.
2. **Dataset Specificity:** Applying EFC to the high-dimensional, anonymized features of the Elliptic dataset.

3. **Parameter Tuning:** While using standard EFC parameters, the specific values (`n_bins=30`, `cutoff_quantile=0.9`, `pseudocounts=0.1`) were chosen based on experimentation within this project's context.

Essentially, we used the EFC package "as intended" for anomaly detection but carefully configured its training data and parameters for the specific task of identifying illicit Bitcoin transactions in the Elliptic dataset scenario. The helper functions described in [**?**] primarily manage the data loading, preprocessing, splitting, evaluation, and visualization around the core EFC calls, rather than altering the EFC mechanism itself.

3.4. Task

The primary task addressed in this study is to evaluate the effectiveness of the Energy Flow Classifier (EFC), implemented 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].

Specifically, the task is 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.

Success in this task would demonstrate EFC's potential as a viable tool for flagging potentially fraudulent or anomalous activities in Bitcoin transactions, leveraging its unsupervised, energy-based approach to overcome challenges like label scarcity and potentially detect novel deviations from normal behavior. The results will provide insights into how EFC performs in this financial forensics context compared to implicitly known benchmarks from related studies using the same dataset.

4. 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.

4.1. Shared Experimental Setup

Unless explicitly stated otherwise in the description of a specific experiment, the following setup, derived from the procedures detailed in Section 3, was consistently used across the results presented below:

- **Dataset Split:** The EFC model was trained exclusively on transactions labeled as Licit from time steps 1 to 34. Evaluation was performed on the test set containing both Licit and Illicit transactions from time steps 35 to 49. Transactions labeled 'Unknown' were excluded from both training and testing.
- **Feature Set:** The input to the EFC model consisted of the 165 anonymized features described in Section 3.1, after removing the time step feature. These features were scaled to the range [0, 1] using Min-Max scaling, with the scaler fitted only on the training data (Licit transactions, steps 1-34).
- **EFC Configuration:** The core EFC implementation (Section 3.3) utilized the following default hyperparameters based on initial tuning and common practice:
  - Number of bins for feature discretization `n_bins=30`
  - Cutoff quantile for anomaly threshold `cutoff_quantile=0.9` (meaning the energy threshold is set at the 90th percentile of the energy distribution of the Licit training data)
  - Pseudocounts `pseudocounts=0.1` (to handle zero probabilities)
- **Evaluation Metrics & Outputs:** Model performance was assessed using a combination of quantitative metrics and qualitative analysis:
  - **F1-Score (Macro Average):** As the primary evaluation metric, we adopted the macro-averaged F1-score, consistent with the benchmark study by [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.
  - **Class-Specific Metrics (Illicit Class):** While F1-Macro provides an overall view, our primary interest lies in detecting the Illicit class (class 1). Therefore, 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).
  - **EFC Energy Distributions:** For each experiment, histograms comparing the distribution of EFC energy scores assigned to Licit versus Illicit transactions in the test set were generated. These plots provide a visual assessment of the model's separation capability.

- **Detailed Results Storage:** Key information for each experimental run, including the sizes of the training and testing datasets (and their class distributions), the calculated performance metrics, and the confusion matrix, were systematically collected and saved into individual CSV files for detailed comparison and analysis across experiments.

The following subsections will now present the results from the specific experiments conducted under this framework, highlighting deviations from this shared setup where applicable.

4.2. Experiments

This section details the experimental evaluations conducted to assess the proposed methodologies. We performed three primary experiments focusing on data balancing, feature engineering/selection, and model comparison/tuning, respectively. All experiments utilized the EFC dataset, preprocessed as described previously, employing a standard train-test split methodology.

Experiment 1: Impact of Data Balancing Techniques. This experiment investigated the effect of various data balancing strategies on classification performance for the inherently imbalanced EFC dataset. The baseline performance was established using the original, unbalanced dataset. This was compared against four common balancing techniques applied to the training dataset and test dataset: creating a balanced subset with equally distributed classes, by undersampling the majority class before splitting, Synthetic Minority Over-sampling Technique (SMOTE), Random Oversampling of the minority class, and Random Undersampling of the majority class. The test set composition remained consistent across most techniques to ensure fair evaluation. Performance was evaluated using Accuracy, Precision, Recall, F1-Score (weighted), Macro F1-Score, and confusion matrices. Table **??** summarizes the dataset characteristics after applying each technique and the corresponding classification results.

Experiment 2: Impact of Feature Selection. Following the analysis of data balancing, Experiment 4 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. The experiment systematically varied the number of selected features, testing values of $k \in 10, 20, 30, 40, 50, 60$. The feature selection process using `SelectKBest` was applied 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. The decision to conduct feature selection in two distinct scenarios within Experiment 2—one including the aggregated neighborhood statistics and one explicitly excluding them—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 the same metrics as Experiment 3 (Accuracy, Precision, Recall, F1-Score, Macro F1-Score). 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.

Experiment 3: Combining Feature Selection and Data Balancing. This experiment investigated the combined effect of feature selection and data balancing on the performance of the EFC classifier, building upon the findings from Experiment 1 (data balancing) and Experiment 24 (feature selection). The core idea was to first reduce the dimensionality of the dataset using the feature selection technique identified in Experiment 4, and then apply the SMOTE balancing technique from Experiment 1 to the reduced-feature training data before training the EFC model.

Specifically, for each value of $k \in \{10, 20, 30, 40, 50, 60\}$, we first applied the SelectKBest algorithm with the f_classif scoring function to the original, unbalanced dataset (as performed in the first scenario of Experiment 2) to obtain a dataset containing only the top k features. Subsequently, this k-feature dataset underwent the SMOTE procedure: it was split into training and testing sets, SMOTE was applied *only* to the training portion to balance the classes, and the EFC classifier was then trained on this balanced, feature-selected training data. Finally, the trained EFC model was evaluated on the corresponding unbalanced test set (containing the same k selected features). Performance was assessed using the standard suite of metrics (Accuracy, Precision, Recall, F1-Score, Macro F1-Score) to determine if applying dimensionality reduction prior to SMOTE balancing could yield improved classification performance compared to using SMOTE on the full feature set (Experiment 1) or using feature selection alone (Experiment 2).

5. Conclusion

This study evaluated the Energy Flow Classifier (EFC) as a one-class anomaly detection method for identifying illicit Bitcoin transactions within the Elliptic dataset, focusing on scenarios with label scarcity. Our empirical results demonstrate that EFC offers a viable approach, particularly leveraging its inherent ability to train solely on normal (Licit) data.

The primary advantage of EFC lies in its one-class nature, making it suitable for real-world scenarios where illicit transaction labels are rare. However, our baseline experiment (Exp 1 - Unbalanced), using default parameters and training only on Licit data, revealed a significant disadvantage: high sensitivity to class imbalance. While achieving high overall accuracy (0.908) and weighted F1-score (0.891), the model performed poorly

Table 3. EFC Performance Summary Across Experiments: Baseline vs. Best Results from Data Balancing, Feature Selection, and Combined Approaches (Selected by F1-Macro).

Experiment / Technique	TP	FN	FP	TN	Accuracy	Precision	Recall	F1-Score (Weighted)	F1-Macro
Exp 1: Baseline									
Unbalanced Dataset	15117	470	1064	19	0.908	0.876	0.908	0.891	0.488
Exp 1: Balancing									
Best Technique (SMOTE)	10831	1775	530	12076	0.909	0.913	0.909	0.908	**0.908**
Exp 2: Feature Selection									
Best FS (Agg. Excluded, k=10)	[TP?]	[FN?]	[FP?]	[TN?]	0.865	0.893	0.865	0.877	**0.686**
Best FS (Agg. Included, k=*)	[TP?]	[FN?]	[FP?]	[TN?]	[Acc?]	[Prec?]	[Rec?]	[F1?]	**[F1-Macro?]**
Exp 3: FS + Balancing									
Best FS + SMOTE (k=*)	[TP?]	[FN?]	[FP?]	[TN?]	[Acc?]	[Prec?]	[Rec?]	[F1?]	**[F1-Macro?]**

Note: TP=True Positives, FN=False Negatives, FP=False Positives, TN=True Negatives reported for the test set. For experiments with multiple sub-runs (varying k or balancing method), the row represents the technique yielding the highest F1-Macro score (shown in bold). Metrics are rounded to three decimal places. F1-Score refers to the weighted average unless otherwise specified. *Placeholders ‘[?]‘ and ‘k=*‘ require data from the corresponding best-performing run’s CSV files.*

in distinguishing the minority Illicit class, resulting in a low F1-Macro score of 0.488.

Crucially, Experiment 1 demonstrated that EFC’s performance can be substantially improved through standard data preprocessing techniques. Applying SMOTE balancing to the training data yielded the best results among the balancing methods tested, dramatically increasing the F1-Macro score to 0.908, indicating a much more balanced detection capability across both Licit and Illicit classes.

Further experiments explored feature selection (Exp 2) and its combination with balancing (Exp 3). Feature selection using `SelectKBest` showed that EFC can operate with reduced dimensionality; for instance, using only the top 10 non-aggregated features yielded an F1-Macro of 0.686, outperforming the baseline but not reaching the levels achieved with SMOTE on the full feature set. The results for feature selection including aggregated features and the combined feature selection with SMOTE approach (currently placeholders in Table 3) require further analysis to draw definitive conclusions about their potential benefits.

In summary, EFC presents a promising tool for detecting anomalous Bitcoin transactions, especially given its one-class training capability. Its main drawback is a pronounced sensitivity to class imbalance, which necessitates mitigation strategies like SMOTE balancing for effective performance (achieving F1-Macro 0.908). While feature selection is possible, careful consideration of trade-offs is required, as dimensionality reduction might impact performance compared to using balanced, full-feature data. The effectiveness of EFC in this context is therefore contingent on appropriate data preprocessing and potentially hyperparameter tuning, sensitivity to which was noted in Section 1 and implied by the experimental variations, to balance the detection of illicit activities against false alarms.

6. 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. *Future Work:* Conduct a systematic hyperparameter search (e.g., using Grid Search or Bayesian Optimization) for `n_bins` and `cutoff_quantile` to precisely map the performance landscape and identify potentially superior configurations compared to the default settings or the best results found so far (F1-Macro 0.908 with SMOTE).
- **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? *Future Work:* Evaluate alternative over/under-sampling techniques and investigate incorporating cost-sensitivity directly into the EFC framework or threshold selection process.
- **Exploring Feature Selection Synergies:** Experiments 2 and 3 explored feature selection, showing potential but requiring further analysis (placeholders in Table 3). 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? *Future Work:* Complete the analysis for Exp 2 (FS with aggregates) and Exp 3 (FS + SMOTE) by filling in the placeholder results. Investigate more sophisticated feature selection methods beyond `SelectKBest`, potentially including feature interaction analysis or embedding-based selection, and evaluate their impact on EFC performance. Analyze the computational trade-offs (training/inference time) associated with feature reduction.
- **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? *Future Work:* Explore methods to integrate graph structure. This could involve: a) using graph embeddings (e.g., Node2Vec, GraphSAGE) derived from the transaction graph as input features to EFC, or b) investigating graph-native extensions or adaptations of the EFC concept itself that directly model energy flow across edges.
- **Temporal Dynamics and Concept Drift:** The study used a fixed train/test split. How does EFC perform in a more realistic, evolving environment? Does the optimal energy threshold shift over time (concept drift)? *Future Work:* Design experiments simulating online detection, potentially involving periodic retraining of EFC or adaptive threshold mechanisms to cope with evolving Licit transaction patterns and potentially new types of Illicit activity.
- **Interpretability and Explainability:** While EFC identifies anomalies based on energy scores, understanding *why* a specific transaction is flagged remains challenging. Which features contribute most to a high energy score for a given anomalous transaction? *Future Work:* Develop or adapt techniques to provide local explanations for EFC predictions, potentially by analyzing feature contributions to

the energy calculation for specific high-energy instances. Visualizing the energy landscape in reduced dimensions could also offer insights.

- **Comparative Benchmarking:** The conclusion highlights EFC's potential but lacks a direct quantitative comparison against state-of-the-art methods on this specific task setup. *Future Work:* Perform a rigorous benchmark comparing the optimized EFC (e.g., best configuration from hyperparameter tuning and balancing/FS experiments) against other one-class methods (like updated OCSVM, Isolation Forest variants) and relevant supervised/graph-based methods (e.g., GCNs as used by [Weber et al. 2019]) using the same data splits and evaluation metrics for a fair comparison.
- **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? *Future Work:* Conduct scalability experiments using larger subsets of the Bitcoin blockchain or datasets with higher dimensionality to assess EFC's feasibility for deployment in high-throughput environments.

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.

7. 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**.

7.1. 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].

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.

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.

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

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`.

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.

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.

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.