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 that operates without the need for a third-party moderator [Nakamoto 2008]. Its architecture relies on mathematics, programming, and advanced cryptography to maintain an immutable ledger of financial transactions — known as the blockchain thus eliminating 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].

It's crucial to differentiate between cryptocurrency manipulation and traditional currency manipulation. The latter is generally a state-level action, often legal but potentially contentious internationally [Domanski and Sushko 2011]. Cryptocurrency manipulation, however, refers to deceptive strategies used by various actors to distort the market artificially, usually for profit. Common types of such manipulation are:

Pump-and-dump fraud is a manipulative tactic where actors coordinate to artificially boost a cryptocurrency's price ('pump'). This attracts investment before the perpetrators sell their holdings ('dump') at the inflated value, leading to a price collapse and losses for subsequent buyers [Karim and Mikhael 2018].

Wash trading is a fraudulent practice in cryptocurrency markets where a single entity or colluding group creates artificial trading activity by simultaneously or rapidly buying and selling the same asset, effectively trading with themselves. This manipulation, often executed using multiple accounts or automated bots, aims to mislead market participants by creating a false impression of high trading volume and liquidity. The ultimate goal is to make the cryptocurrency appear more popular and active than it truly is, thereby attracting unsuspecting investors, manipulating market sentiment, and potentially influencing the asset's price [Gandal et al. 2018, Edelman et al. 2018].

However, identifying anomalous patterns in complex data streams—such as network traffic or financial transactions —remains a critical challenge, as traditional methods often struggle with the high dimensionality, evolving characteristics, and sheer volume of modern data. Within this context, a promising approach grounded in principles from statistical physics is the Energy-based Flow Classifier (EFC). Originally proposed using the Inverse Potts model, the EFC characterizes the probability distribution of normal network

flows through an energy function derived from observed data patterns [Pontes et al. 2019]. In this paper, we investigate the application of the Energy-based Flow Classifier (EFC) for detecting anomalous Bitcoin transactions. Previous research has applied EFC to classify unusual network traffic, demonstrating its potential for supporting intrusion detection mechanisms [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 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 in financial fraud datasets like Elliptic. 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. Finally, previous research has successfully applied EFC to classify unusual network traffic [Pontes et al. 2019, Souza et al. 2022], demonstrating its potential in analogous domains involving the detection of anomalies within complex flow data. Therefore, in this paper, we investigate the application of EFC specifically for detecting anomalous Bitcoin transactions, hypothesizing that its unique characteristics offer advantages in this challenging domain.

Building upon the promise of the EFC framework, this paper presents a comprehensive empirical evaluation of its application to the Elliptic dataset. Our investigation confirms EFC's capability to distinguish between licit and illicit transaction patterns based on their energy profiles, particularly demonstrating strong potential in identifying illicit activities even when trained solely on licit data. However, the results underscore a critical sensitivity to specific configuration choices, revealing significant trade-offs between maximizing the detection of illicit transactions (recall) and minimizing false alarms (precision), especially concerning the energy threshold defining anomalous behavior. **Notably, despite these sensitivities, optimized approaches combining data preprocessing techniques like SMOTE resampling with feature selection achieved the highest F1-Macro score of 0.757 in our experiments, demonstrating the potential for significant performance gains with tailored configurations.** The following sections delve into these findings, quantifying the performance under the one-class anomaly detection setup, exploring the impact of key hyperparameters like discretization bins and the anomaly cutoff threshold, and comparing against baseline data balancing approaches, ultimately providing crucial insights into EFC's practical utility and optimization challenges in this complex financial forensics domain.

2. Data and Methods

This section details the data and methods employed in our study, which builds upon the foundational research presented in "Machine learning methods to detect money laundering in the Bitcoin blockchain in the presence of label scarcity" [Lorenz et al. 2021]. That seminal 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 super-

vised or semi-supervised frameworks requiring at least some labels. Our research, forming part of a larger dissertation effort, diverges by investigating the Energy Flow Classifier (EFC), introduced in Section 1. EFC provides an alternative approach rooted in statistical physics, designed to model the typical 'energy' profile of normal transactions. By focusing on identifying high-energy deviations representing anomalies, EFC inherently addresses the label scarcity problem from an unsupervised or anomaly detection perspective, offering a framework potentially better suited for identifying novel or diverse fraudulent behaviors without relying heavily on pre-existing fraud labels. The following subsections detail our specific dataset, preprocessing steps, EFC implementation, evaluation task, and computational setup.

2.1. 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 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 contains over 200,000 transactions, only a subset is explicitly labeled. Based on the analysis by [Weber et al. 2019], approximately 46,564 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 (approx. 42,019 instances in the original labeled set).
- **Illicit:** Transactions linked to known illicit activities, including scams, ransomware, terrorist financing, Ponzi schemes, and dark market operations (approx. 4,545 instances in the original labeled set).

The vast majority of transactions (over 150,000) remain unlabeled. Within the labeled portion, illicit transactions represent a small minority approx. 9.8% of labeled data, or just over 2% of the total dataset. This significant class imbalance and the large volume of unlabeled data make the Elliptic dataset a realistic and challenging benchmark for evaluating fraud detection techniques, particularly unsupervised or anomaly-based methods like EFC that do not strictly rely on extensive labeling. Table 1 provides a summary of the dataset statistics based on the original publication [Weber et al. 2019].

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%)

2.2. 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.

First, addressing the labels described in Section 2.1, 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.

2.3. 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 2.2, 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.

2.4. Task

The primary task addressed in this study is to evaluate the effectiveness of the Energy Flow Classifier (EFC), implemented as described in Section 2.3, 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.

3. Background and Related Work

WIP

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 2, 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 2, 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 2.1, 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 2.3) 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. Experiment 3: Unbalanced Dataset Techniques

Experiment 3 investigates the effect of explicitly addressing the inherent class imbalance of the Elliptic dataset during the training phase on the performance of the Energy Flow Classifier (EFC). While the primary approach in this work treats EFC as a one-class anomaly detector trained solely on Licit data (as described in Section 4.1), this experiment compares that baseline against several standard techniques designed to handle imbalanced data, applied *before* training the EFC. The goal is to determine if these techniques, which expose the model to Illicit samples (or synthetic versions thereof) during training, yield better performance compared to the one-class baseline on the unseen test set.

The following configurations were evaluated:

- **Baseline (One-Class EFC):** This corresponds to the standard setup described in Section 4.1. The EFC model was trained using only the Licit transactions from the training time steps (1-34), setting `base_class=0`. Evaluation was performed on the original, imbalanced test set (time steps 35-49). This serves as the reference point.
- **SMOTE (Synthetic Minority Over-sampling Technique):** The training dataset (Licit and Illicit samples from time steps 1-34) was resampled using SMOTE [**?**]. SMOTE synthetically generates new samples of the minority (Illicit) class to balance the class distribution in the training set. The EFC model was then trained on this balanced training set (using `base_class=0`, although both classes are now present). Evaluation was performed on the original, imbalanced test set.
- **Random Over-Sampling:** The minority (Illicit) class in the training set (time steps 1-34) was randomly duplicated until its size matched the majority (Licit) class, creating a balanced training set. The EFC model was trained on this over-sampled training set and evaluated on the original, imbalanced test set.
- **Random Under-Sampling:** The majority (Licit) class in the training set (time steps 1-34) was randomly down-sampled until its size matched the minority (Illicit) class, creating a balanced training set. The EFC model was trained on this under-sampled training set and evaluated on the original, imbalanced test set.
- **Artificial Equal Distribution (Subsampling):** A distinct approach was tested where the entire labeled dataset (train + test, time steps 1-49) was pooled. An artificially balanced dataset was created by taking all Illicit samples and an equal number of randomly selected Licit samples from this pool. This balanced pool was then split into new training and testing sets. The EFC was trained and evaluated on these artificially balanced splits. Note that this method involves data leakage from the test set into the balancing process and uses a modified test set, making it less representative of a real-world scenario compared to the other techniques but included for comparative analysis within this specific experiment.

The performance results for the different imbalance handling techniques applied prior to EFC training are summarized in Table 3. The table presents the primary F1-Macro score, alongside Precision, Recall, and F1-Score calculated specifically for the Illicit class based on the confusion matrices obtained from evaluating on the respective test sets.

Table 3. Performance Metrics for EFC with Different Imbalance Handling Techniques (Experiment 3). Illicit class metrics are calculated from confusion matrices.

Technique	F1-Macro	Precision (Illicit)	Recall (Illicit)	F1-Score (Illicit)
Baseline (One-Class)[a]	0.488	0.934	0.970	0.952
Random Oversampling (ROS)[a]	0.533	0.938	**0.988**	**0.962**
Random Undersampling (RUS)[a]	0.652	**0.971**	0.885	0.926
SMOTE[b]	**0.908**	0.953	0.859	0.904
Balanced (Artificial)[c]	0.644	0.933	0.378	0.538

[a] Evaluated on the original, imbalanced test set (16,670 samples: 1,083 Illicit, 15,587 Licit).

[b] Evaluated on a test set modified by SMOTE (25,212 samples, balanced class distribution). Results may not be directly comparable to others.

[c] Evaluated on an artificially created balanced test set (2,727 samples: 1,363 Illicit, 1,364 Licit). Results may not be directly comparable to others.

The results presented in Table 3 highlight the significant impact of data balancing strategies on EFC performance evaluation. Focusing first on the techniques evaluated on the original, imbalanced test set (Baseline, ROS, RUS), we observe distinct trade-offs. The Baseline (One-Class) approach, despite being trained only on Licit data, achieves a high F1-score for the Illicit class (0.952), primarily driven by a very high recall (0.970). However, its F1-Macro score is the lowest (0.488), indicating extremely poor performance in classifying the Licit class (as evidenced by only 19 True Negatives vs. 1064 False Positives in the confusion matrix), suggesting the default 0.9 quantile threshold might be too low when trained only on Licit data, leading to excessive flagging of transactions as Illicit.

Comparing the resampling techniques on the original test set, Random Oversampling (ROS) yields the highest Recall (0.988) and F1-Score (0.962) for the Illicit class, suggesting it is most effective at capturing the majority of illicit transactions among these methods. However, similar to the baseline, its F1-Macro (0.533) is relatively low, indicating struggles with the Licit class (only 70 True Negatives). Random Undersampling (RUS) achieves the highest F1-Macro (0.652) among this group and the highest Precision (0.971) for the Illicit class, meaning that when it flags a transaction as Illicit, it is highly likely to be correct. However, this comes at the cost of lower Recall (0.885), missing more Illicit transactions compared to ROS and the Baseline.

The SMOTE technique shows the highest F1-Macro score overall (0.908). However, this result must be interpreted with caution, as both the training and testing sets were balanced using SMOTE, as indicated by the test set size (25,212) and confusion matrix totals. Evaluating on a balanced test set artificially inflates metrics like F1-Macro compared to evaluation on the naturally imbalanced distribution, limiting its direct comparability to a real-world deployment scenario. Similarly, the 'Balanced (Artificial)' technique, which used subsampling to create balanced train and test sets from the pooled data, shows moderate F1-Macro (0.644) but very poor Recall (0.378) for the Illicit class on its specific test set.

In summary, Experiment 3 suggests that while standard resampling techniques applied before EFC training can influence the balance between Precision and Recall for the

Illicit class and potentially improve F1-Macro scores compared to the one-class baseline (particularly RUS), none dramatically outperform the baseline's ability to *recall* Illicit transactions on the original test set. However, the baseline's extremely low F1-Macro highlights a significant issue with false positives using the default threshold. Techniques like SMOTE appear promising based on F1-Macro, but their evaluation on modified test sets makes direct comparison difficult. This underscores the importance of evaluating on the true, imbalanced test distribution and suggests that threshold tuning (explored in later experiments) might be crucial for optimizing the practical utility of the one-class EFC.

5. Conclusion

WIP

6. Future Work

WIP

7. Reproducibility

WIP

7.1. Computational Environment

WIP

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.