BitCaught: Accurately Identifying Ransomware Related and Generally Malicious Bitcoin Addresses Using Machine Learning and Past Blockchain Activity

Daniel Leef
Computer & Network Security
University of Oregon
Eugene, OR
dleef@uoregon.edu

Abstract— **The innately anonymous nature of Bitcoin, among numerous other cryptocurrencies, has unsurprisingly made it a uniquely attractive financial vehicle for criminals. With Bitcoin continuing to increase in social prevalence and fiscal value, the need to counteract the criminal activity within its peer-to-peer network is crucial to enforce legal accountability, limit the pecuniary freedom of malicious individuals, and subsequently improve its attractiveness for regulatory adoption. This paper proposes a method for detection within the Bitcoin network using machine learning and novel features formulated through readily available blockchain activity. The two datasets used for feature creation and model training differ in their corresponding criminal classifications: BitcoinAbuse provides a user-reported, crowdsourced dataset of generally malicious individuals (crimes relating to blackmail scams, darknet transactions, ransomware, etc.) while BitcoinHeist provides a much more extensive and verified dataset consisting of addresses associated exclusively with ransomware. A deep neural network (four hidden layers) and simple neural network (one hidden layer) were trained and validated on data with a maximum of 19 features, mainly achieved with self-labeled "Close Network" features, for detecting generally malicious and ransomware related users with BitcoinAbuse and BitcoinHeist, respectively. The entire system, titled BitCaught, accomplished a maximum accuracy of 95.50% in detecting users participating in ransomware related activities and 90.81% in detecting users engaging in generally illegal activities.**

***Keywords*—Bitcoin, Ransomware, Cybercrime, Blockchain, Machine Learning, Cryptocurrency, Supervised Learning**

I. Introduction

As cryptocurrencies, notably Bitcoin, become increasingly prevalent and economically influential, their innate anonymity, legitimized through the blockchain, is uniquely attractive for illicit activity [1]. Without ever having to disclose any identifying information, nefarious individuals can engage relatively unfettered in transactions pertaining to illegal behavior using Bitcoin or any of the numerous alternatives. With a lack of criminal accountability continuing to proliferate Bitcoin's peer to peer network, there exists an increasingly apparent need to develop systems and methodologies which can accurately determine malicious activity, mainly corresponding to ransomware, blackmail scams, and darknet transactions, on the blockchain. While one such approach contributing to this effort is the highly effective deanonymization of Bitcoin wallet addresses through address clustering [2], this project focuses on accurately detecting maliciousness instead of undoing anonymity. Being able to determine if a given address is engaging in criminal activity, without any previous reports on that wallet, would be beneficial for a multitude of reasons. Firstly, the ability to accurately detect criminal behavior would give government bodies the necessary additional confidence for supporting cryptocurrencies through regulatory adoption; this mainstream, economic incorporation is necessary for the future growth and usage of Bitcoin and other cryptocurrencies. Secondly, a flagging system could be utilized where addresses suspected of criminal behavior could be flagged so that they're unable to engage in future transactions or convert their digital assets to fiat currency. Lastly, an accurate detection system could be highly useful in the context of address clustering, where the deanonymization technique could additionally mark addresses within a cluster suspected of malicious behavior, thus incriminating all of the addresses within the cluster that belong to a given entity [3].

Bitcoin, as with all cryptocurrencies, relies on the blockchain in order to function: this permanent, decentralized ledger verifies and stores the transaction history of all users within the Bitcoin network. Machine learning is a uniquely plausible avenue of detection within the Bitcoin network due to the accessibility and extensiveness of transaction data provided by the blockchain. The information corresponding to each wallet address, such as total Bitcoin spent and received, can be used to form novel features which distinguish criminally related addresses from innocent ones. In order to accomplish a supervised learning approach, additional datasets which accurately categorize some addresses as nefarious (i.e. engaging in criminal behavior) are of course necessary; BitcoinAbuse.com serves as my source for generally criminal addresses and the BitcoinHeist dataset as my source for specifically ransomware associated addresses. While

BitcoinAbuse.com contains crowdsourced, user reported wallet addresses, BitcoinHeist uses verified ransomware related addresses from previous widely adopted studies [4]. BitcoinHeist also contains a large number of categorized non-malicious wallet addresses, and I use a mix of these along with randomly selected wallet addresses nonexistent in either criminal address records as my non-malicious address data source.

While this work utilizes neural networks for the ultimate goal of accurate detection, the novelty and effectiveness of this work revolves around feature formulation derived from blockchain information. I train and test the models on multiple datasets with varying features and dimensionality in order to compare and contrast the accuracy resulting from those features. In this process, I also modify the attributes of the neural networks for higher accuracy, while weighing heavily the importance of training time and computational resource exhaustiveness. Ultimately, the end result system, BitCaught, seeks to reveal if *criminally related wallet addresses reflect identifiable trends and behavior* in comparison to their innocent peers, and more specifically if *the subgroup of criminality in ransomware showcases more easily identifiable attributes than the all-encapsulating, generally criminal parent group.*

II. Related Work

While there are numerous pieces highlighting the usage of machine learning with problems in the space of cryptocurrency, the works most relevant to this paper can be separated into three categories. The first consists of research which is closely related to the work discussed in this paper. The second highlights varying approaches and implementations of address clustering methods followed by brief discussion of potential incorporation of this work into those methodologies mentioned. The third showcases theoretical frameworks and explanations for the association of criminal activity within cryptocurrency ecosystems.

A. Detecting Criminal Cryptocurrency Activity

In the area of active detection on cryptocurrency blockchains, four papers stand out in their relevance. While machine learning is by no means a requirement for this area of related research, each of these works utilize learning (supervised and unsupervised) for the purpose of criminal detection within cryptocurrency networks, which gives additional credence to the methods used in this paper.

Two of these papers focus on the same end goal as with this work, which is the accurate detection of malicious users. A publication shows the high accuracy achieved with an XGBoost classifier for identifying accounts in the Ethereum network associated with illegal activity. A relatively small dataset was used (< 5000 data points) with a high dimensionality of 42. The combination of the highly effective gradient boosting library XGBoost and a small dataset with abundant features yielded exceptional results of ~96% accuracy using the same validation method as in this work (10-fold cross validation). This high accuracy was accomplished for detecting generally malicious Ethereum addresses, that is users associated with crimes such as scam lotteries, Ponzi schemes, mirroring websites, and imitating other contract addresses providing tokens, among others. Similar to this work as well, the machine learning model was tasked with binary classification for whether a given address was related to illegal activity or not [5]. Another publication which utilizes an ensemble tree-based model for predicting illicit Bitcoin accounts from licit ones was able to accomplish an accuracy of 66% [6]. A common attribute within these works is that they also identify the most important features in the training and efficacy of their models, a research trend which, aside from these two papers, proved to be very helpful for my own methodologies involving feature formulation.

Other publications focus more generally on anomaly detection within the Bitcoin blockchain, which is somewhat analogous to criminally related activity. One combines both unsupervised and supervised learning with a mixture of classifiers (random forest, XGBoost, logistic regression) in order to specifically identify potential money laundering trends. Perhaps one of the biggest takeaways from this study is that supervised learning proved to be significantly more effective than unsupervised, especially in the context of the experiment where the learning either switches from unsupervised to supervised or remains unsupervised [7]. Another publication exclusively uses unsupervised learning in order to detect anomalies among user and transaction graphs (data derived from the blockchain) [8]. Both of these publications accomplished accuracies in the low to mid 80s (%).

B. Deanonymization

An incredibly vibrant area of research relating to cryptocurrency is that of deanonymization, specifically through the usage of address clustering methods. Some works use supervised machine learning to classify Bitcoin addresses into their corresponding entity "types", with two of these studies accomplishing accuracies of 77% and 79% for correct cluster classification [9][10], while other works focus on truly clustering together addresses which belong to the same user. The latter of these works which focuses on address clustering accomplish high accuracies (mid 80's to low 90's %) using various address, transaction, and user relation graph approaches [11]; one also utilizes a combination of blockchain and off-blockchain information in order to deanonymize the addresses corresponding to users [12].

For the deanonymization methods which focus on categorizing addresses into different types, BitCaught wouldn't be a particularly useful addition as those types could also include criminal subgroups (blackmail, darknet transaction, etc.) if the data were available to the researchers at the time of the studies. However, the latter methods which specialize in assigning multiple addresses back to their single entity would uniquely benefit from the BitCaught system. Within a given a cluster of addresses which corresponds to a user, BitCaught could be used to identify if any of the given addresses reflect criminal behavior, which would thus incriminate the entire cluster. This work would be beneficial in the context of address clustering methods as the clusters could be additionally checked for

malicious activity. In the potential realm of instantiating a flagging system across cryptocurrency networks, this approach would allow batches of addresses, belonging to clusters, to be flagged for malicious activity instead of just single addresses. Overall, the incorporation of this work into those deanonymization methods would prove efficient in uncovering criminally related addresses and would subsequently allow more swift action against the users behind those addresses.

C. Analysis of Criminals and Cryptocurrency

A 2019 publication confirms the association of criminal activity with Bitcoin through a comprehensive analysis of newspapers, magazines, journals, trade publications, and reviews covering cryptocurrency related crimes. Bitcoin, even in comparison to the Monero cryptocurrency which garners more anonymity, is the most used cryptocurrency for legal and illegal purposes due to user familiarity and trust. In the context of politics, Bitcoin is an accepted form of political donation as determined by the U.S. Federal Election Commission in 2014, however in regard to country-wide adoption, the world's leading central banks are resistant due to their inability to exert economic control in a cryptocurrency landscape. Bitcoin has also been used in assisting terrorist organizations from within the United States, due to its security and ease of transfer [13].

Despite Bitcoin's popularity, lack of regulation, and inherent pseudo anonymity, therefore making it the cryptocurrency of choice by cybercriminals, law enforcement has yet to notice any significantly large criminal activity, with a focus on money laundering, occurring on its blockchain. Possible explanations for this are that the relevant populations are technically inept for its usage or lack resources, or that Bitcoin is primarily used as a transition currency where criminals immediately convert it to fiat currency upon finishing their transactions. Of course, another possibility is that Bitcoin is being used extensively for criminal activities, however law enforcement hasn't been able to resolve investigative complexities for tracking this activity or simply aren't looking deep enough into it. Regardless, criminal justice systems should exert more energy in understanding and conveying the potential or existing threats relating to virtual currencies, specifically Bitcoin [14].

III. Methodology

The methodology for this work involved the datasets selected and used for training, the features formulated, and the types of machine learning models used.

A. Data Collection

I used two main datasets in this work, one of which contains wallet addresses associated with generally malicious behavior and the other which contains wallet addresses exclusively associated with ransomware. For both of these datasets, the only important raw information for this work were the wallet addresses and their corresponding malicious binary (criminal or not criminal).

BitcoinAbuse

BitcoinAbuse.com is a website which allows users to report and classify Bitcoin wallet addresses with the following categories: ransomware, darknet market, bitcoin tumbler, blackmail scam, sextortion, and other. The original dataset was immense as it contained every report on the website, so the following was required before the dataset was to be used for this work:

- *Cleaning:* separate reports corresponding to the same address simply incremented a *num_reports* variable for that address and appended the corresponding criminal activity of that address to a list. The mode of this list served as the malicious activity corresponding to that wallet address. This cleaning process resulted in a more comprehensive dataset with wallet addresses, number of reports, and the corresponding crime they're accused of.
- *Managing Noisiness*: because this data is crowd sourced, there is a need to eliminate wallet addresses which might be falsely reported as criminal. To solve this, only wallet addresses with more than 5 reports were allowed into the final dataset; this threshold was reached through trial and error based off of an acceptable size of the resulting data.

BitcoinHeist

Whereas BitcoinAbuse required maintenance due to being crowdsourced and thus not completely dependable with the addresses it classifies as criminal, BitcoinHeist is a dataset containing ransomware addresses from multiple widely adopted studies (Princeton, Montreal, and Padua) [4]. Thus, this dataset is highly dependable and substantially larger than BitcoinAbuse due to it not requiring any cleaning.

Non-malicious

BitcoinHeist contained numerous addresses classified as "white", meaning they don't correspond to any ransomware family as defined in the dataset. In order for them to be fully classified as non-malicious for this work, each "white" address used was cross-checked with the BitcoinAbuse records. To account for any possible bias within the "white" records, I also randomly selected addresses off of the Bitcoin blockchain which were nonexistent in either dataset. I evenly distributed these two types of non-malicious wallet addresses for the final non-malicious dataset.

B. Feature Formulation

The features used in this project can be separated into three distinct categories: *basic*, *close network*, and *additional*. I define *target_addr* as any wallet address within the finished datasets used for model training and testing (with binary labels indicating criminality).

Basic

The basic features are foundational starting points for the data used for training and testing. Their core purpose is to provide a general picture of a given wallet address's activity on the Bitcoin blockchain. They consist of the following:

- *Total_balance*: the total Bitcoin balance of *target_addr*
- *Bitcoin_spent:* the amount of Bitcoin spent by *target_addr*
- *Bitcoin_received:* the amount of Bitcoin received by *target_addr*

- *Num_tx*: the number of Bitcoin transactions that *target_addr* has been involved with

Close Network

The close network features were conceived with the desire to paint an accurate picture of the spending habits of the wallet addresses closest to *target_addr*, hence the name close network. I theorize these features to be useful as criminal users are more likely to have criminal "friends" that further reflect trends and behaviors not commonly prevalent with non-malicious users; this logic also applies vice versa for innocent users. The close network features consist of the following:

- *CN_sender_bitcoin_sent_avg*: the average amount of Bitcoin sent from those addresses which *target_addr* has sent Bitcoin to
- *CN_sender_bitcoin_sent_sd*: the standard deviation of the amount of Bitcoin sent from those addresses which *target_addr* has sent Bitcoin to
- *CN_sender_bitcoin_received_avg*: the average amount of Bitcoin received by those addresses which *target_addr* has sent Bitcoin to
- *CN_sender_bitcoin_received_sd*: the standard deviation of the amount of Bitcoin received by those addresses which *target_addr* has sent Bitcoin to
- *CN_sender_total_transactions_avg*: the average number of transactions by those addresses which *target_addr* has sent Bitcoin to
- *CN_sender_total_transactions_sd*: the standard deviation of the number of transactions by those addresses which *target_addr* has sent Bitcoin to
- *CN_sender_total_balance_avg*: the average total balance of Bitcoin of the addresses which *target_addr* has sent Bitcoin to
- *CN_receiver_bitcoin_sent_avg*: the average amount of Bitcoin sent from those addresses which *target_addr* has received Bitcoin from
- *CN_receiver_bitcoin_sent_sd*: the standard deviation of the amount of Bitcoin sent from those addresses which *target_addr* has received Bitcoin from
- *CN_receiver_bitcoin_received_avg*: the average amount of Bitcoin received by those addresses which *target_addr* has received Bitcoin from
- *CN_receiver_bitcoin_received_sd*: the standard deviation of the amount of Bitcoin received by those addresses which *target_addr* has received Bitcoin from
- *CN_receiver_total_transactions_avg*: the average number of transactions by those addresses which *target_addr* has received Bitcoin from
- *CN_receiver_total_transactions_sd*: the standard deviation of the number of transactions by those addresses which *target_addr* has received Bitcoin from
- *CN_receiver_total_balance_avg*: the average total balance of Bitcoin of the addresses which *target_addr* has received Bitcoin from

Additional

These features were exclusively influenced by a paper in which the authors sought to accurately classify Bitcoin addresses, with a notable innovation being their usage of temporal data attributes called "Moments" in the training and testing of their learning models [15]. The additional features are derived from the section in their paper where the most important, or heavily weighed, attributes determined by their model are revealed ranked; these features consist of the first four fields in those rankings:

- *F_tx*: transaction frequency, measured as the average number of transactions involving *target_addr* from the day of the first transaction to the day of data recording
- *N_inputs_spent_avg*: the average number of input addresses in spent transactions by *target_addr* (transactions where *target_addr* is sending Bitcoin)
- *N_outputs_spent_avg*: the average number of output addresses in spent transactions by *target_addr*
- *N_received_tx*: the number of received transactions by *target_addr* (transactions where users send Bitcoin to *target_addr*)

Due to issues with accurate calculation of the latter three additional features, only transaction frequency ended up being included in the datasets for testing and analysis.

C. Supervised Learning Algorithms

The type of data being used in this work is complex in nature as it relates to cryptocurrency activity, a not entirely well understood area of data, thus algorithms immediately ruled out were decision trees, linear regressions, logistic regressions, naïve Bayes, and nearest neighbor. A SVM (support vector machine) was considered, however the high dimensionality of the data would most likely lead to suboptimal results in comparison to neural networks which can more accurately map complex features to the correct output. Thus, the supervised learning models determined for usage were neural networks (multi-layer perceptron), one deep (four hidden layers) and one simple (one hidden layer). The following are attributes of the model:

Deep Neural Network

- Input Layer: accepts input data with varying dimensionality determined by features. The number of neurons in this layer is equivalent to the dimensionality of the data.
- Hidden Layers: four hidden layers are used with decreasing neurons in each layer so as to filter out the less relevant inferred features in each layer. The number of neurons was determined through trial and error in recording accuracy, with the layers holding 100, 50, 25, and 12 neurons, respectively for 19 and 18-dimensional data (basic, close network, and transaction frequency features vs. basic and close network features), and 50, 40, 25, and 10 neurons for 4-dimensional data (basic features only).
- Activation Function: Sigmoid as shown in (1). Sigmoid is uniquely useful for binary classification problems since it normalizes neuron weights and assigns the

output a probability that is mapped to either 1 or 0. Except for the input layer, Sigmoid activation is used at every layer of the network.

- Output Layer: the output layer has one neuron for the binary output of the program, indicating whether the inputted wallet address reflects criminal activity or not.

Another multi-layer perceptron was used for comparison as it's been remarked that one hidden layer is enough to solve a vast majority of complex learning problems [16]:

Neural Network

- Hidden Layers: only one hidden layer is used which has 200 neurons for 19-dimensional data (basic, close network, and transaction frequency features), 175 neurons for 18-dimensional data (close network and basic features), and 50 neurons for 4-dimensional data (only basic features). This number of neurons was determined through trial and error in accuracy tests.

All other attributes of the neural network are the same as with the deep neural network.

$$S(x) = 1 / (1 + e^{-x}) \quad (1)$$

IV. Implementation

The following is a breakdown of the separate files, technologies, and sequential data aggregation approach used in this work to ensure the previously outlined methodology was fulfilled as intended.

A. Data Collection

- *BlockCypher API*: used to retrieve all the necessary blockchain information for calculating the basic, close network, and additional features. I registered three different email addresses with the service in order to use 15 tokens, with each account being allowed a maximum of 2000 requests per day. Thus, this API freely allowed me to make 6000 requests per day.

- *MaliciousCSV.py*: cleaned and managed noisiness within the raw BitcoinAbuse dataset, as outlined in III. Methodology, A. After this was finished and the remaining addresses were stored in a dictionary (hashmap), the addresses were iterated through with each address being used in a BlockCypher API request to retrieve basic features. A row with the wallet address, basic features, and criminal binary were then stored in a spreadsheet whose title specified the data as "filtered". The data cleaned, fetched, and stored by this file exclusively relates to BitcoinAbuse, which are the wallet addresses associated with generally malicious activity.

- *NonMaliciousCSV.py*: retrieved and stored basic feature data points for non-malicious wallet addresses both randomly selected off of the Bitcoin blockchain (and checked for absence within the malicious datasets) as well as from the BitcoinHeist dataset (classified as "white" addresses).

- *JoinBitcoinHeist.py*: same function as MaliciousCSV.py except for the BitcoinHeist dataset; no cleaning or noisiness management was needed. The data fetched and stored by this file exclusively relates to BitcoinHeist, which are the wallet addresses associated with generally malicious activity.

- *CloseNetworkAggregation.py*: used the BlockCypher API to fetch the data necessary to calculate the close network features; the close network features were then appended to the basic features already existing for generally malicious, ransomware related, and non-malicious addresses.

- *AdditionalStatsAggregation.py*: used the BlockCypher API to fetch the data necessary to calculate the transaction frequency additional feature; this was appended to the close network and basic features already existing for generally malicious, ransomware related, and non-malicious addresses.

- *BlockchainInfoStatsAggregation.py*: used the open Bitcoin blockchain API to fetch and calculate the remaining additional features; this file is unfinished as it does not accurately compute these features. Because of this, the transaction frequency is the only additional feature included in the results and analysis.

The most notable takeaway from the data collection implementation is that datasets with more features were built on top of one another. First, the basic features were retrieved and stored for a set of addresses; then, close network features were retrieved for the same addresses and appended onto the preexisting basic features. Finally, additional features were appended onto all of the other features already existing. This structure allowed easy testing and comparison between different features for the varying datasets.

B. Machine Learning

The Tensorflow Keras API was used to build, compile, train, and validate both supervised learning models used in this project. The scikit-learn library was used for cross validation and Matplotlib was used for graphs showing model performance over time.

- *MaliciousModel.py*: implemented, trained, and validated a neural network with one hidden layer.

- *MaliciousModelDeep.py*: implemented, trained, and validated a deep neural network with four hidden layers

For collecting results on varying data sources, the dimensions within the files were changed as were the corresponding neurons in the hidden layers.

V. Results

All results in the Figures and Tables section were obtained using ten-fold cross validation, in which the dataset being used is split into ten different clusters, of which nine are used to train the model on and the remaining one is used for validation. This process is repeated ten times and the average of the ten sessions is recorded along with the standard deviation. In the

Performance section, the graphs represent a training / validation split of 80 / 20 (%) over 80 epochs with a batch size of 20.

A. Performance

The following figures highlight accuracy and loss trends of the two supervised learning models on varying datasets. Figure 7 reflects a deep neural network with increased neurons in the following order of hidden layers: 200, 100, 50, 25.

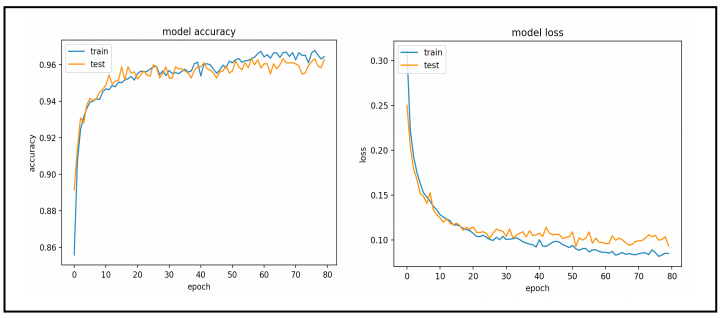

Fig. 1. Neural Network on 19-dimensional data, BitcoinHeist

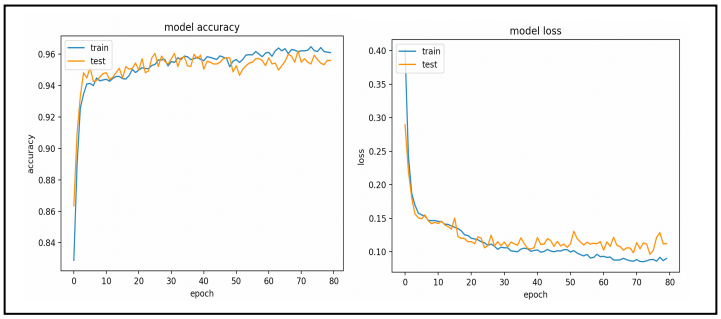

Fig. 2. Deep Neural Network on 19-dimensional data, BitcoinHeist

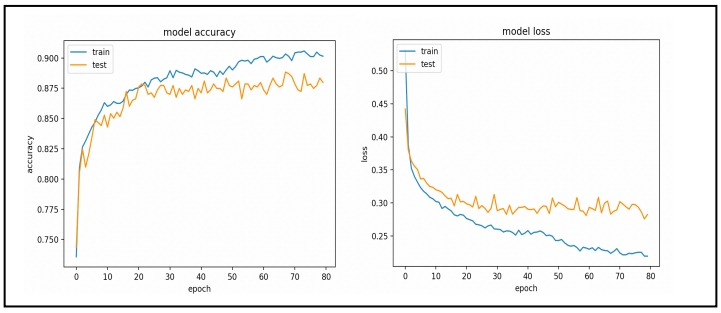

Fig. 3. Neural Network on 18-dimensional data, BitcoinAbuse

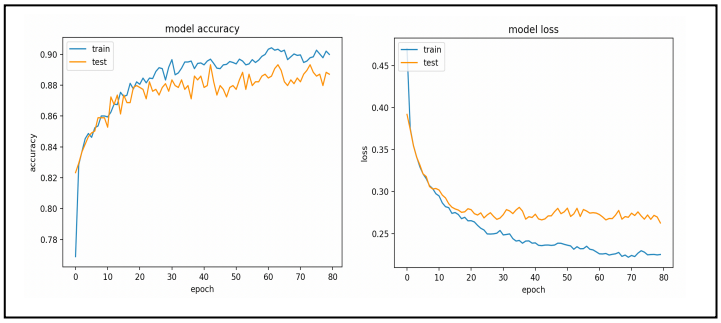

Fig. 4. Deep Neural Network on 18-dimensional data, BitcoinAbuse

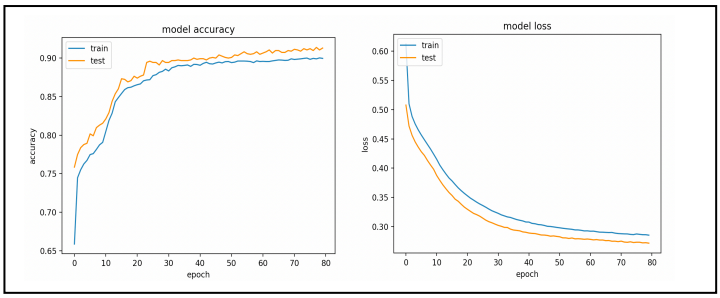

Fig. 5. Neural Network on 4-dimensional data, BitcoinAbuse

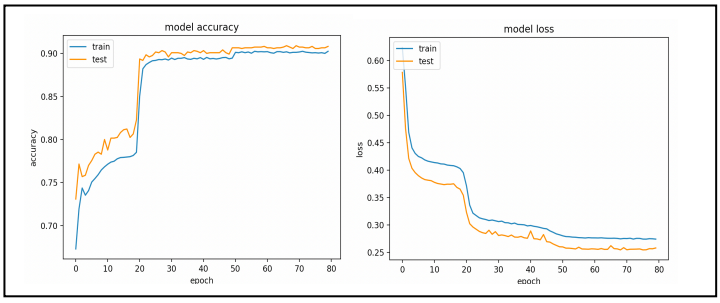

Fig. 6. Deep Neural Network on 4-dimensional data, BitcoinAbuse

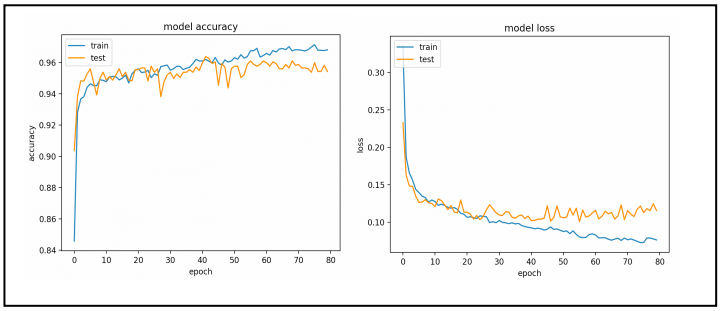

Fig. 7. Deep Neural Network on 19-dimensional data, increased neurons, BitcoinHeist

B. Figures and Tables

TABLE I. NEURAL NETWORK, ONE HIDDEN LAYER WITH 50, 175, AND 200 NEURONS FOR BASIC, CLOSE NETWORK, AND ADDITIONAL FEATURES

Data Source, Features	***Number of Data Points***	***Distribution of Malicious / Non-malicious (%)***	***Average Accuracy (%)***	***Standard Deviation Accuracy (%)***
BitcoinHeist, Basic (4-dim)	14,490	23.6 / 76.4	91.19	1.03
BitcoinHeist, Close Network (18-dim)	14,054	26.6 / 73.4	94.03	0.30
BitcoinHeist, TX Frequency (19-dim)	10,139	27.7 / 72.3	**95.75**	0.74
BitcoinAbuse, Basic (4-dim)	6,150	28.5 / 71.5	**90.31**	0.63
BitcoinAbuse, Close Network (18-dim)	4,073	26.3 / 73.7	88.56	**1.94**

TABLE II. DEEP NEURAL NETWORK, FOUR HIDDEN LAYERS WITH 100, 50, 25, AND 12 NEURONS FOR CLOSE NETWORK AND ADDITIONAL FEATURES / 50, 40, 25, AND 10 NEURONS FOR BASIC FEATURES

Data Source, Features	***Number of Data Points***	***Distribution of Malicious / Non-malicious (%)***	***Average Accuracy (%)***	***Standard Deviation Accuracy (%)***
BitcoinHeist, Basic (4-dim)	14,490	23.6 / 76.4	91.59	0.46
BitcoinHeist, Close Network (18-dim)	14,054	26.6 / 73.4	94.54	0.48
BitcoinHeist, TX Frequency (19-dim)	10,139	27.7 / 72.3	**95.11**	0.69
BitcoinAbuse, Basic (4-dim)	6,150	28.5 / 71.5	**89.89**	0.82
BitcoinAbuse, Close Network (18-dim)	4,073	26.3 / 73.7	88.26	**1.74**

VI. ANALYSIS

A. Performance

The neural network and deep neural network reflected similar training, validation, and loss trends across the varying datasets. With BitcoinHeist-derived 19-dimensional data, consisting of basic, close network, and transaction frequency features, both models reached similarly high levels of training and validation accuracy, with the two lines remaining close to each other through the 80 epochs as in Figures 1 and 2. The models both reflected worse results with the BitcoinAbuse-derived 18-dimensional data, consisting of basic and close network features, as seen in Figures 3 and 4; a notable difference in performance between BitcoinAbuse and BitcoinHeist is the substantial gap between training and validation accuracy during the later epochs. The difference in loss between training and validation also increased going from BitcoinHeist to BitcoinAbuse (Figure 1 to Figure 3). When the models were fitted and tested against BitcoinAbuse-derived basic features (4-dimensional), the trends of training and validation accuracy were significantly smoother: the two accuracies also stayed significantly closer together over the epochs than with the higher dimensionality BitcoinAbuse data, as seen in Figures 5 and 6. The deep neural network also yielded significantly steeper jumps in accuracy and loss than with the simple neural network (Figure 6).

The most interesting attribute of Figures 5 and 6, however, is that validation accuracy always remained higher than training accuracy and validation loss always remained lower than training loss. This doesn't occur in any of the other scenarios and is most likely due to the fixed training / validation split of 80 / 20 (%). If the training and validation were randomized across the epochs, then this most likely wouldn't have occurred; none of the other figures reflect this as the other datasets were, by chance, not organized in such a way to allow it. While these graphs were partially due to issues in the dataset distribution, Figures 5 and 6 are provide additional justification for using a k-fold cross validation for the true results in this work, as those results reflect performance of the models outside of one set of test data. Figure 7 shows that a deep neural network worsens the accuracy and loss on 19-dimensional BitcoinHeist data, with the gap between lines increasing over the epochs. Lastly, a notable observation was that the neural network trained ~17% faster than the deep neural network, indicating that the simple neural network should be preferred over the deep neural network if they yield similar accuracies as it requires significantly decreased training time.

B. Cross Validation Results

The two learning models accomplished similar efficacy across the varying datasets and features, with the most notable result for both models occurring with the highest dimensionality BitcoinHeist data. Both models accomplished over 95% accuracy in detecting whether Bitcoin wallet addresses corresponded to ransomware related or innocent users, with the highest accuracy of **95.75%** being achieved using the simple neural network, as seen in Table I. The highest accuracy achieved for classifying generally malicious Bitcoin wallet addresses from the BitcoinAbuse dataset was **90.31%** also using the simple neural network, however only with basic features involved. When close network features were added to the BitcoinAbuse dataset, the accuracy for detecting criminally related wallet addresses actually decreased for both supervised models. The opposite was true for the BitcoinHeist dataset, as the accuracy was positively correlated with the number of features included. The highest standard deviation within the cross-validation process occurred in the same dataset, BitcoinAbuse with close network features, for both models, bolded in both Tables I and II. Additionally, not reflected in these tables is the training time of the two models, however the simple neural network trained and validated noticeably quicker than that of the deep neural network. Despite this, both models trained slowly with the first two BitcoinHeist datasets as those files contained the most data points.

The distribution of wallet addresses reflecting malicious activity and wallet addresses reflecting non-malicious activity was similar across all of the datasets, and this was by design. Different distributions of addresses were tested in additional datasets and the accuracy consistently worsened as the distribution approached 50 / 50 (%). The accuracies were also worse when including less criminally related addresses in the datasets, thus the distributions determined to be optimal are reflected in the tables as mid to high 20's / low to mid 70's (%). As for data point density, the number of data points decreased among the sets for two main reasons: calculating and appending features required more time to amalgamate the data from APIs, and the BitcoinAbuse dataset simply had less criminally related addresses to work with. Not reflected in either table are the results of a perceptron (neural network with no hidden layers) and supper vector machine on the datasets; overall, both supervised learning models achieved significantly higher accuracies than the perceptron and support vector machine for all datasets. The resulting averages and standard deviations from these other models were based on rudimentary datasets, however they performed worse than the deep and simple neural networks to such an extent that they'd serve little purpose in being retested on final datasets.

VII. Discussion and Future Work

Both models returned worse results for the BitcoinAbuse dataset, and this was likely partially due to the lesser number of data points available in comparison to that of BitcoinHeist. Another potential explanation is that the crowdsourced data from BitcoinAbuse is more susceptible to false positives in terms of maliciousness as opposed to the verified ransomware addresses from BitcoinHeist, even with managed noisiness. However, a more intuitive explanation for the worse accuracy is that generally malicious activity reflects less discernible trends and behavior than that of specific ransomware related activity. Ransomware is a subgroup of the criminal wallet addresses included in the BitcoinAbuse dataset, among other crimes like blackmail scams and darknet transactions, thus the criminal activity reflected through these users' blockchain activity is generalized across multiple different areas of illegality. With BitcoinHeist, however, the criminal users to be detected are specifically engaged in ransomware, thus this criminal subgroup is more likely to reflect more definitive and identifiable trends in blockchain behavior than with general illicit activity.

Another notable result from cross-validation was the reasonably high accuracy accomplished with basic features for generally criminal addresses (~90%), showing that malicious users can be discerned through simple traits. However, these traits aren't dependable in the long run as users can easily change their spending habits to go against the data the model is trained on. The close network features assist in combatting this issue as they provide the supervised learning model with a cohesive picture of the spending habits of those addresses closest to a given address, which is information not easily changeable by a criminal user. When the close network features were appended to the BitcoinAbuse data points, both models performed worse in detecting the criminally related wallet addresses, which is consistent with intuition as additional information further separates the various criminally related addresses from one another and thus the overall behavior of that group of criminals. Also in line with intuition is the substantial increase in accuracy with the addition of the close network features in the BitcoinHeist data (~3%), as the extra information further refined the trends existing among ransomware related wallet addresses. Perhaps the most significant attribute of the BitcoinHeist dataset was the transaction frequency feature, as it drastically increased the accuracy of both models (~0.6% - ~1.75%) while only increasing the dimensionality by one.

In comparing the accuracy accomplished by the two models, the simple neural network performed better or similarly to the deep neural network while also sustaining a shorter training time due to its lesser number of hidden layers. Both the number of epochs (80) and the batch size (20) led the models to train fast and accomplish good accuracies, with increased epochs and decreased batch sizes resulting in marginally increased accuracies but significantly worse training times. In the future, however, I'd like to formally record the accuracies resulting from altering the epochs and batch sizes as well as their corresponding training times for quantitative data to reference. Additionally, I only managed to append the close network features to the BitcoinAbuse dataset, and while they worsened the models' performances, I'd like to expand the BitcoinAbuse close network features dataset in the future to be comparable to that of the BitcoinAbuse basic features dataset to allow for better accuracy comparisons. Furthermore, being able to accurately calculate and append the additional features not including the transaction frequency would be helpful for further analysis of the models' efficacies and the impact of those features. More generally, further data collection would be highly beneficial for future experimentation, with a target of ~20,000 data points for the BitcoinHeist sets and ~8,000 for the BitcoinAbuse sets.

Outside of data collection, I'd also like to generalize this work to a wider variety of cryptocurrencies, specifically Monero which provides even more anonymity than Bitcoin. Formal conceptualization of a theoretical framework for cryptocurrency that utilizes flagging in partnership with BitCaught would also be beneficial to highlight the potential benefits and downsides of such a system. Lastly, in the future I believe live data collection, involving using BitCaught on all Bitcoin transactions within a 24-hour period to see what percentage of addresses it deems ransomware related and generally malicious in a given day, would be highly useful in determining whether BitCaught identifies more or less criminal activity than is expected on the Bitcoin blockchain, as measured by other studies.

VIII. Conclusion

Bitcoin continues to be used every day across the world, and as it grows in popularity as well as monetary value, the need for criminal accountability within its network becomes increasingly apparent. BitCaught is not a final solution to this problem, as no cybersecurity solution ever is, however it's a useful stepping stone towards enforcing legality in Bitcoin's rapidly growing ecosystem. This work provides useful insight into the detectability of users engaging in ransomware as well as those engaging in generally illicit activities. More importantly, however, BitCaught is able to accurately detect nefarious individuals participating in ransomware schemes with accuracy greater than 95%. In the future, BitCaught will hopefully evolve to be effective in a more generic criminal landscape, however the accurate detection it currently provides on ransomware related wallet addresses is highly beneficial as ransomware remains a prevalent issue for organizations, corporations, and individuals alike, especially with cryptocurrencies such as Bitcoin involved.

Acknowledgment

The research work disclosed in this paper was partially guided by the Center for Cybersecurity and Privacy at the University of Oregon as well as the University of Oregon Ripple Scholarship for cybersecurity research relating to financial transactions and cryptocurrencies.

REFERENCES

[1] Sean Foley, Jonathan R Karlsen, Tālis J Putniņš, Sex, Drugs, and Bitcoin: How Much Illegal Activity Is Financed through Cryptocurrencies?, *The Review of Financial Studies*, Volume 32, Issue 5, May 2019, Pages 1798–1853, https://doi.org/10.1093/rfs/hhz015

[2] M. Harrigan and C. Fretter, "The Unreasonable Effectiveness of Address Clustering," 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress, Toulouse, France, 2016, pp. 368-373, doi: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0071.

[3] D. Ermilov, M. Panov and Y. Yanovich, "Automatic Bitcoin Address Clustering," *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Cancun, Mexico, 2017, pp. 461-466, doi: 10.1109/ICMLA.2017.0-118.

[4] Akcora, C.G., Li, Y., Gel, Y.R. and Kantarcioglu, M., 2019. BitcoinHeist: Topological Data Analysis for Ransomware Detection on the Bitcoin Blockchain. IJCAI-PRICAI 2020.

[5] Farrugia, Steven & Ellul, Joshua & Azzopardi, George. (2020). Detection of illicit accounts over the Ethereum blockchain. Expert Systems with Applications. 113318. 10.1016/j.eswa.2020.113318.

[6] Nerurkar, Pranav & Busnel, Yann & Ludinard, Romaric & Shah, Kunjal & Bhirud, Sunil & Patel, Dhiren. (2020). Detecting Illicit Entities in Bitcoin using Supervised Learning of Ensemble Decision Trees. 10.1145/3418981.3418984.

[7] Lorenz, J., Silva, M.I., Aparício, D.O., Ascensão, J.T., & Bizarro, P. (2020). Machine learning methods to detect money laundering in the Bitcoin blockchain in the presence of label scarcity. *ArXiv, abs/2005.14635*.

[8] Pham, Thai & Lee, Steven. (2016). Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods.

[9] Harlev, M.A., Yin, H., Langenheldt, K.C., Mukkamala, R., & Vatrapu, R. (2018). Breaking Bad: De-Anonymising Entity Types on the Bitcoin Blockchain Using Supervised Machine Learning. *HICSS*.

[10] Hao Hua Sun Yin, Klaus Langenheldt, Mikkel Harlev, Raghava Rao Mukkamala & Ravi Vatrapu (2019) Regulating Cryptocurrencies: A Supervised Machine Learning Approach to De-Anonymizing the Bitcoin Blockchain, Journal of Management Information Systems, 36:1, 37-73, DOI: 10.1080/07421222.2018.1550550

[11] Zheng, Baokun & Zhu, Liehuang & Shen, Meng & Du, Xiaojiang & Guizani, Mohsen. (2020). Identifying the vulnerabilities of bitcoin anonymous mechanism based on address clustering. Science China Information Sciences. 63. 10.1007/s11432-019-9900-9.

[12] D. Ermilov, M. Panov and Y. Yanovich, "Automatic Bitcoin Address Clustering," *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Cancun, Mexico, 2017, pp. 461-466, doi: 10.1109/ICMLA.2017.0-118.

[13] Kethineni, Sesha & Cao, Ying. (2019). The Rise in Popularity of Cryptocurrency and Associated Criminal Activity. International Criminal Justice Review. 30. 105756771982705. 10.1177/1057567719827051.

[14] Brown, Steven David. (2016). Cryptocurrency and criminality: The Bitcoin opportunity. The Police Journal. 89. 10.1177/0032258X16658927.

[15] Lin, Yu-Jing & Wu, Po-Wei & Hsu, Cheng-Han & Tu, I-Ping & Liao, Shih-wei. (2019). An Evaluation of Bitcoin Address Classification based on Transaction History Summarization.

[16] Jeff Heaton. 2008. Introduction to Neural Networks for Java, 2nd Edition (2nd. ed.). Heaton Research, Inc.

APPENDIX

Below is a list of all of the technologies and libraries used in this work.

- Python 3 for all code files
- Tensorflow with Keras API for building, compiling, training, and validating the two supervised learning models
- Scikit-learn library for cross validation
- Matplotlib for performance graphs
- Panda for reading in data from spreadsheets
- GitHub for saving and updating BitCaught repository