XRAD: Ransomware Address Detection Method based on Bitcoin Transaction Relationships

KAI WANG, School of Computer Science, Fudan University, Shanghai, China
MICHAEL TONG, software school, Fudan University, Shanghai, China
JUN PANG, Computer Science and Communications, University of Luxembourg, Esch-sur-Alzette, Luxembourg
JITAO WANG, School of Computer Science, Fudan University, Shanghai, China
WEILI HAN, Software School, Fudan University, Shanghai, China

Recently, there is a surge in ransomware activities that encrypt users' sensitive data and demand bitcoins for ransom payments to conceal the criminal's identity. It is crucial for regulatory agencies to identify as many ransomware addresses as possible to accurately estimate the impact of these ransomware activities. However, existing methods for detecting ransomware addresses rely primarily on time-consuming data collection and clustering heuristics, and they face two major issues: (1) The features of an address itself are insufficient to accurately represent its activity characteristics, and (2) the number of disclosed ransomware addresses is extremely less than the number of unlabeled addresses. These issues lead to a significant number of ransomware addresses being undetected, resulting in a substantial underestimation of the impact of ransomware activities.

To solve the above two issues, we propose an optimized ransomware address detection method based on Bitcoin transaction relationships, named XRAD, to detect more ransomware addresses with high performance. To address the first one, we present a cascade feature extraction method for Bitcoin transactions to aggregate features of related addresses after exploring transaction relationships. To address the second one, we build a classification model based on Positive-unlabeled learning to detect ransomware addresses with high performance. Extensive experiments demonstrate that XRAD significantly improves average accuracy, recall, and F1 score by 15.07%, 19.71%, and 34.83%, respectively, compared to state-of-the-art methods. In total, XRAD detects 120,335 ransomware activities from 2009 to 2023, revealing a development trend and average ransom payment per year that aligns with three reports by FinCEN, Chainalysis, and Coveware.

CCS Concepts: • **Security and privacy** → *Malware and its mitigation;*

Additional Key Words and Phrases: Ransomware, Bitcoin transaction, transaction relationships, illegal activities

This article is supported by National Key Projects of Research and Development (Grant No. 2022YFC3301305, 2023YFC3304400), and NSFC (Grant nos. 62172100, 92370120).

Authors' Contact Information: Kai Wang, School of Computer Science, Fudan University, Shanghai, China; e-mail: 20110240034@fudan.edu.cn; Michael Tong, software school, Fudan University, Shanghai, China; e-mail: 21210240466@m.fudan.edu.cn; Jun Pang, Computer Science and Communications, University of Luxembourg, Esch-sur-Alzette, Luxembourg; e-mail: jun.pang@uni.lu; Jitao Wang, School of Computer Science, Fudan University, Shanghai, China; e-mail: jtwang21@m.fudan.edu.cn; Weili Han, Software School, Fudan University, Shanghai, China; e-mail: wlhan@fudan.edu.cn.

ACM 1559-1131/2024/10-ART48
https://doi.org/10.1145/3687487

ACM Reference Format:
Kai Wang, Michael Tong, Jun Pang, Jitao Wang, and Weili Han. 2024. XRAD: Ransomware Address Detection Method based on Bitcoin Transaction Relationships. *ACM Trans. Web* 18, 4, Article 48 (October 2024), 33 pages. https://doi.org/10.1145/3687487

1 Introduction

Bitcoin [56] is designed to provide a pseudonymous payment scheme, making value transfers between participants without revealing any personally identifiable information. Due to Bitcoin's increasing popularity and adoption, an increasing number of users conduct Bitcoin transactions to carry out commercial activities [65]. As of December 31, 2023, there are around 974 million transactions in Bitcoin. Many online retailers, such as *Jomashop* [76] and *Newegg* [39], now accept Bitcoin as a payment method for their services. Furthermore, the decision of countries such as El Salvador and the Central Africa Republic to adopt Bitcoin as a legal tender [5, 46] further suggests that Bitcoin is gradually becoming a world currency.

Bitcoin's pseudonymity and its status as a world currency make it attractive to criminals seeking to transfer illegal income without revealing their identities. This leads to the proliferation of various illegal activities utilizing Bitcoin, particularly ransomware activities. The growth of ransomware activities affects many industries, e.g., finance, energy, and medical care, causing serious economic losses. For example, Colonial Pipeline [53], the largest fuel pipeline system in the USA, paid nearly 5 million dollars via bitcoins after a ransomware infection in May 2021. The **United Kingdom's National Cyber Security Centre (NCSC-UK)** has recognized ransomware as the biggest cyber threat to the country [10]. Moreover, the adverse impact of ransomware activities is more serious than we realize. According to Hernandez-Castro et al. [34], the prevalence of CryptoLocker ransomware seems significantly higher than anticipated, with a larger proportion of victims (41%) claiming to pay the ransom than previously estimated by Symantec (3%) and Dell SecureWorks (0.4%). However, despite the discovery of over 600 ransomware families that demand bitcoins as ransom [51], the associated Bitcoin addresses are rarely disclosed, leading to an underestimation of the severity of ransomware activities. This highlights the need to detect more ransomware addresses.

There are efforts to detect ransomware addresses in various ways. In the beginning, some researchers search and crawl webpages related to ransomware activities through tedious data-gather steps [60]. However, this method could only detect a very small number of ransomware addresses due to the limited number of relevant webpages. More recently, clustering heuristics and BitcoinHeist [1] are developed to detect ransomware addresses. Many studies analyze Bitcoin transactions using various clustering heuristics [23, 37, 49, 52, 60, 75]. For example, the *co-spend* heuristic [52] considers all input addresses of a transaction belonging to the same person. If only part Bitcoin addresses of a ransomware activity are known, clustering heuristics use these address as the seed address and can detect additional Bitcoin addresses controlled by this ransomware activity by analyzing Bitcoin transactions of these seed addresses. However, clustering heuristics are limited in their ability to detect unknown ransomware activities due to the lack of seed addresses. The recent study titled BitcoinHeist [1] partitions the Bitcoin network into 24-hour-long windows and extracts six features for Bitcoin addresses from the daily transaction graph. It subsequently employs the topological data analysis method to calculate the distance between Bitcoin addresses and cluster them. However, the study solely focuses on the transaction behavior of an address itself, without taking into account the transaction behavior of the addresses with which they interact. Given that criminals frequently utilize multiple Bitcoin addresses to conceal the ransom transfer trajectory, relying solely on the transaction behavior of an address itself is insufficient to accurately reflect the ransom transfer pattern.

In practice, detecting ransomware addresses in large-scale Bitcoin transactions faces two issues. The first one is that relying solely on the features of an address itself is insufficient to represent a criminal's activity characteristics. To conceal their identities, criminals use multiple Bitcoin addresses to carry out the collection, transfer, and withdrawal of ransoms in a ransomware activity [75]. Therefore, only the features of an address itself are insufficient to achieve precise detection, and we must also take into account the features of other addresses it transacts with. The second one is that the number of disclosed ransomware addresses and unlabeled addresses is highly unbalanced. There are approximately 21,000 disclosed ransomware addresses [75], representing only 0.002% of the complete Bitcoin addresses. There exist several other labeled addresses in Bitcoin [32, 40, 51], but the vast majority of Bitcoin addresses are unlabeled and involved in various transaction activities. As a consequence, even though the existing labeled datasets contain addresses involved in various transaction activities, they only represent a small portion of complete Bitcoin addresses. Thus, the transaction behavior in these labeled datasets cannot represent the transaction behavior of all unlabeled addresses, which makes it difficult to distinguish the differences in address features between ransomware addresses and all unlabeled addresses only using the existing labeled datasets. These two issues together make it an interesting but challenging research problem to detect ransomware addresses in large-scale Bitcoin transactions.

To overcome the aforementioned issues, we propose XRAD, a high-performance ransomware address detection method based on Bitcoin transaction relationships. To address the first issue, we design a cascade feature extraction method for Bitcoin transactions to aggregate features of related addresses after defining transaction relationships (*Neighbor* and *Sibling*) between addresses. This method can also be applied to other cryptocurrencies that use the UTXO model, such as Litecoin. To address the second issue, we treat ransomware address detection as **Positive-Unlabeled learning (PU-learning)** problem [29] and build a classification model based on the **bagging-based PU-learning (PU-Bagging)** [55], using a variant of bagging to select unlabeled Bitcoin addresses as negative samples.

According to our performance evaluation, XRAD demonstrates significantly higher performance in detecting ransomware addresses than existing methods. In three scenarios, our approach achieves an average accuracy, recall, and F1 score that are 15.07%, 19.71%, and 34.83% higher than the best results obtained by state-of-the-art methods. Finally, we apply XRAD to complete Bitcoin addresses from 2009 to 2023. We detect 120,335 ransomware activities in Bitcoin, which cause the loss of 345,890.56 bitcoins (approximately $3,043,136,328.63[1]). The revealed development trend of ransomware activities and the average ransom payment per year are similar to three reports [16, 24, 30] by the **Financial Crimes Enforcement Network (FinCEN)**, Chainalysis, and Coveware. To the best of our knowledge, our study is the first to detect ransomware addresses on the complete set of Bitcoin addresses, and our result reflects the serious impact of ransomware activities in Bitcoin.

Roadmap. After the introduction, Section 2 provides a brief introduction to Bitcoin and ransomware. Section 3 shows the detailed process of dataset preparation. We present the design of our method in Section 4. The experimental results are summarized and analyzed in Section 5. We further discuss the achievements and associated limitations of our method in Section 6. Finally, we discuss related works in Section 7 and conclude this article in Section 8.

[1] We crawl the historical exchange price of Bitcoin to USD and use the lowest price of each day to calculate the amount of ransom.

2 Background Knowledge

2.1 Bitcoin and UTXO

Bitcoin [56] is a decentralized digital currency that enables secure and efficient peer-to-peer transactions without the need for intermediaries such as banks. Bitcoin's design is based on the concept of pseudonymity, where users can transfer funds without revealing their personal information. In Bitcoin, users can generate as many Bitcoin addresses as they want to make transactions with other participants. Besides, Bitcoin utilizes blockchain technology to maintain a public distributed ledger of complete transactions, which is usually managed by a peer-to-peer network. This mechanism enables anyone to access complete transaction data publicly, making it feasible for researchers to study ransomware activities through Bitcoin transactions.

Unlike the account model in banks, Bitcoin employs the **Unspent Transaction Output (UTXO)** model for privacy and to prevent double-spending. A UTXO refers to the remaining bitcoins after a Bitcoin transaction is executed. A Bitcoin transaction spends the UTXOs generated by previous transactions and generates new ones. These new UTXOs are then recorded into a database as inputs, which can be used later for a new transaction. Each UTXO is a discrete piece and must be spent as a whole. To facilitate users, UTXOs can be combined and split up through Bitcoin transactions to make payments in any denomination. Therefore, a Bitcoin transaction can have multiple inputs and outputs. Bitcoin transaction format exposes more complex transaction relationships between Bitcoin addresses. An in-depth analysis of these transaction relationships can help determine the purpose of the address's transaction activities.

2.2 Ransomware

The threat of malware and cybercrime is rapidly increasing and cannot be ignored [11, 68, 72, 73]. One of the most concerning forms of malware is ransomware, which infects a victim's data or resources and demands a ransom to release them. Ransomware works by blocking victims from accessing their valuable data in two primary ways. The most common one is encrypting files [21], which does not destroy other device functions. The other way is locking victims' computers or other devices, which restricts all operations but does not directly encrypt the data stored on the device.

The earliest known ransomware, called the *PC Cyborg Trojan*, appeared in 1989 and required victims to send $189 to the *PC Cyborg* mailbox in Panama to unlock their devices. Since the emergence of Bitcoin, ransomware activities have become more rampant due to the decentralized and pseudonymous payment mechanism provided by Bitcoin [52]. For example, the worldwide ransomware WannaCry infected over 300,000 computers across 150 countries in 2017 [48], and victims were required to pay $300—$600 via bitcoins. On June 9, 2021, Meatpacker JBS USA paid a ransom via bitcoins equivalent to $11 million following a ransomware activity that disrupted the company's North American and Australian operations [9]. In particular, a special transaction pattern known as the mixing transaction exists in Bitcoin, which consolidates fund transfers between multiple users who do not know each other and their respective recipients into one transaction. This type of transaction obfuscates the transfer of funds between inputs and outputs, thereby enhancing privacy. Therefore, ransomware criminals frequently employ mixing transactions to obscure the ransom transfer trajectory. In conclusion, the emergence of Bitcoin has significantly contributed to the increase in ransomware activities, making it easier for criminals to collect ransoms without worrying about being tracked.

Ransomware families are typically named after the first variant that is discovered, and subsequent variants are given a new name to differentiate them from the original. Each ransomware family has some unique ransomware strains with their own code signature and functionality.

MalwareHunterTeam identifies over 600 ransomware families by specific filename extensions, ransom note names, known hex patterns, email addresses, Bitcoin addresses, and other identifiers [51]. Several ransomware address detection methods are developed to identify more Bitcoin addresses related to ransomware activities. Researchers first obtain the Bitcoin addresses used to collect ransoms through analysis of ransomware code or related webpages, and then detect more ransomware addresses primarily by analyzing Bitcoin transactions. However, due to the pseudonymity of Bitcoin, a significant number of ransomware addresses are not disclosed and identified [34], leading to the underestimation of the impact of ransomware activities. Therefore, detecting additional ransomware addresses to improve the measurement of ransom payments and the overall economic impact of ransomware activities is a crucial question with significant societal implications.

3 Dataset Preparation

To design an effective ransomware addresses detection method, we construct three valuable datasets: (1) a complete Bitcoin transaction dataset, which contains complete transactions from January 3, 2009, to December 31, 2023; (2) a mixing transaction dataset, which contains 12,307,535 Bitcoin mixing transactions; and (3) a disclosed ransomware address dataset, which contains 38 ransomware families with 21,468 Bitcoin addresses.

Complete Bitcoin Transaction Dataset. The complete Bitcoin transaction dataset contains a complete record of all bitcoin transfers among addresses on the Bitcoin blockchain. To obtain this dataset, we first download all raw transaction data. We then develop a data parser tool to convert the raw transaction data into formatted transactions that could be easily processed and analyzed. As a result, we collect all Bitcoin transactions from January 3, 2009, to December 31, 2023, containing approximately 946 million Bitcoin transactions and 1,372 million Bitcoin addresses.

Mixing Transaction Dataset. This dataset contains Bitcoin mixing transactions that we collect and identify. First, we gather Bitcoin addresses from three mixing services, namely, *Bitcoin Fog*, *HelixMixer*, and *CoinJoinMess*, by scraping data from the WalletExplorer website [40]. Subsequently, we extract Bitcoin transactions in which these addresses are involved, including both deposits and withdrawals. In total, we obtain 11,053,907 mixing transactions, of which 431,071 are for *Bitcoin Fog*, 320,379 for *HelixMixer*, and 10,302,457 for *CoinJoinMess*. Second, we use the Coinjoin transaction identification algorithm implemented in Blocksci [42] to identify 591,055 Coinjoin transactions. Third, we discover a dataset containing a portion of mixing transactions for Samourai and Wasabi [57]. Specifically, the dataset includes 7,310 mixing transactions for Samourai and 4,031 mixing transactions for Wasabi. Finally, we use all collected 11,648,890 deduplicated mixing transactions as the seed and apply the mixing transaction expansion algorithm [77], resulting in the expansion of an additional 658,645 mixing transactions. In the end, our mixing transaction dataset contains 12,307,535 unique mixing transactions in total. It is worth noting that the mixing transaction dataset is not only used in the construction of the ransomware address dataset, but also in extracting the address features.

Disclosed Ransomware Address Dataset. Our dataset of disclosed ransomware addresses is a combination of datasets from three well-known studies: Princeton [37], Padua [23], and Montreal [60].

In the Princeton dataset, Huang et al. extract ransomware addresses from reports of ransomware infection on public forums. They also execute a subset of ransomware binaries and collect memory dumps, created files, and screenshots to extract Bitcoin addresses. Their publicly available dataset [36] contains two ransomware families with 16,128 Bitcoin addresses.

In the Padua dataset, Conti et al. extensively search various public forums related to ransomware activities and then use clustering heuristics to identify sets of addresses controlled by the same ransomware family. In the end, they release the dataset [22] containing 24 ransomware families with 4,027 Bitcoin addresses, 4,025 of which are unique.

In the Montreal dataset, Masarah et al. collect 7,037 addresses from the Anti-Phishing Working Group and identify 139 Bitcoin addresses in a thread maintained by Michael Gillespie related to Locky ransomware. Additionally, they find another 46 Bitcoin addresses through further online searches. In total, their publicly available dataset [59] contains 7,222 seed addresses related to 67 ransomware families as well as 10,458 addresses that are identified, 10,444 of which are unique.

To ensure the reliability of our ransomware address dataset, we perform the following four steps on the above three datasets: (1) Although these three datasets are constructed in different ways, there are overlaps among them. For example, both the Montreal dataset and the Princeton dataset collect Bitcoin addresses associated with the Locky ransomware family, resulting in 5,963 duplicated Bitcoin addresses. After merging the above three datasets and excluding duplicate addresses, we obtain 53 ransomware families with 24,487 Bitcoin addresses. (2) We identify that 2 addresses are labeled as CryptoWall and CryptoDefense, and 14 addresses are labeled as Cryptohitman and TowerWeb. This could potentially have a negative impact on the subsequent analysis, so we remove these 16 addresses that are labeled with different tags. (3) Forty-seven Bitcoin addresses obtained by crawling public forums or executing ransomware in a simulated environment do not actually receive any ransom payments, i.e., these addresses are not involved in any transactions. Thus, we decide to exclude those addresses that have never been involved in any Bitcoin transactions. (4) Due to the withdrawal of ransomware addresses through cryptocurrency exchanges and the use of mixing services to obscure the trajectory of ransom transfers, there is a close relationship between ransomware addresses and addresses of the exchange and mixing service. We find that in the ransomware address expansion stage, these studies [23, 60] mistakenly identify some addresses of the exchange and mixing service as ransomware addresses. In this way, we filter out 3,255 misidentified ransomware addresses using labeled data collected from the WalletExplorer website. This involves four ransomware families: CryptoLocker, CryptoWall, DMALocker, and PopCornTime. Among them, 824 addresses labeled as CryptoLocker and 2,398 addresses labeled as CryptoWall are actually BTC-e exchange addresses. After implementing the four steps, we obtain 21,169 ransomware addresses.

The aforementioned study [23] uses clustering heuristics to expand its ransomware address dataset based on the latest available transactions at the time. However, new Bitcoin transactions occur continuously, leading to changes in the dataset of ransomware addresses expanded through clustering heuristics. To ensure that we can collect ransomware addresses in new transactions, we employ a similar approach. Specifically, we use the 21,169 filtered ransomware addresses as seed addresses and expand the ransomware address dataset based on new transactions using *co-spend* heuristics [3], while excluding mixing transactions. We do not use the *change address* clustering heuristic to expand the ransomware address dataset due to its potential for producing false positives. Using the clustering heuristic, we discover additional 299 ransomware addresses, which involve four ransomware families: CryptoWall, DMALocker, Flyper, and APT.

Our final ransomware address dataset contains 38 ransomware families with 21,468 Bitcoin addresses labeled as ransomware addresses. Table 1 shows the number of Bitcoin addresses included in each ransomware family. This dataset will be utilized in the experiments in Sections 4.3 and 5.

4 Design of XRAD

In this section, we present the whole process of XRAD for detecting ransomware addresses in Bitcoin.

Table 1. Number of Bitcoin Addresses in Each Ransomware Family

Name	Number	Name	Number	Name	Number
Cerber	10,165	Locky	7,094	CryptXXX	1,742
CryptoLocker	968	CryptoWall	813	DMALocker	200
CryptoTorLocker2015	159	Globe	106	SamSam	48
EDA2	33	Flyper	31	NoobCrypt	28
Jigsaw	17	KeRanger	12	CryptConsole	7
WannaCry	6	VenusLocker	5	CTB-Locker	4
XLocker	4	XTPLocker	4	APT	3
BadRabbit	2	Chimera	2	Exotic	1
GoldenEye	1	GlobeImposter	1	NotPetya	1
KillDisk	1	NullByte	1	CryptoHost	1
TeslaCrypt	1	Phoenix	1	DoubleLocker	1
7ev3n	1	Xorist	1	ZCryptor	1
ComradeCircle	1	Bucbi	1	–	–

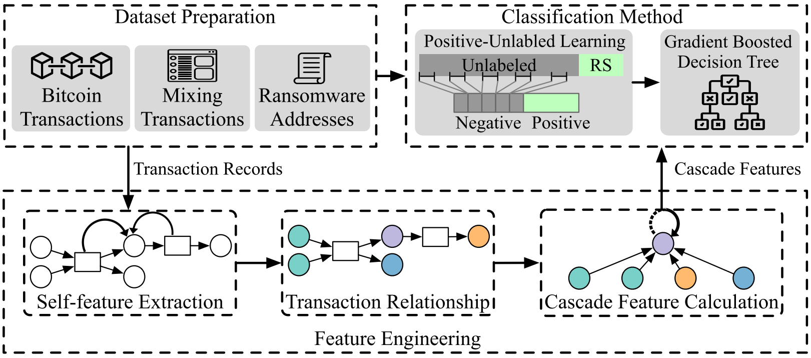


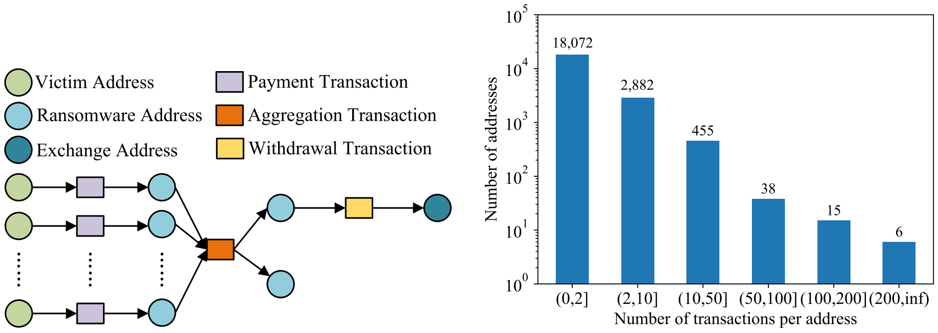
Fig. 1. An overview of XRAD.

4.1 Overview

XRAD, as shown in Figure 1, consists of three components: dataset preparation, feature engineering, and classification method.

First, we construct a comprehensive dataset of all Bitcoin transactions that have occurred since the inception of Bitcoin up to December 31, 2023. In addition, we collect Bitcoin mixing transactions, which involve the use of a third-party service to obscure the bitcoin transfer trajectory. Furthermore, we gather a dataset of Bitcoin addresses that are associated with ransomware activities. The detailed information about these three datasets is described in Section 3.

Second, we develop a three-step approach to extract rich features for each Bitcoin address from both itself and its related addresses. To begin with, we calculate a series of features for each address based on transactions it participates in, using our previous analysis of ransomware activities [75]. These address features are effective in characterizing the transaction behavior of a given Bitcoin address. Then, we explore transaction relationships between addresses, such as *Sibling* and *Neighbor*, and design an innovative cascade feature extraction method for Bitcoin transactions. This method involves aggregating the address features of an address's *Neighbors* and *Siblings* to create cascade features that reflect the comprehensive ransom transfer behavior of criminals in a ransomware activity. By incorporating not only features of the address itself but also those of related addresses, these cascade features provide a more comprehensive reflection of ransomware activities. These



(a) Three main stages of a ransomware activity in Bitcoin

(b) Ransomware address distribution according to different intervals of transactions the addresses are involved in.

Fig. 2. Different functions and characteristics of ransomware addresses at different stages.

cascade features are used to construct the classification model instead of the individual address features.

Finally, we build a classification model based on the PU-Bagging approach, utilizing a variant of the bagging method to select unlabeled samples to construct the training set, which helps to mitigate the impact of the data imbalance issue.

4.2 Feature Engineering

Simply extracting features from a ransomware address does not provide a comprehensive picture of the transaction behavior of criminals involved in ransomware activities. Previous works [23, 37] show that a ransomware activity in Bitcoin is accomplished mainly through three stages: (1) collecting ransoms from victims, (2) aggregating ransoms from addresses, and (3) withdrawing ransoms in cryptocurrency exchanges. Figure 2(a) illustrates a typical case of ransom transfer in three stages in CryptoWall. It is worth noting that the ransom payment by victims and the withdrawal by criminals may involve transactions with cryptocurrency exchanges. Thus, trading with exchanges is an important part of ransomware activities. Therefore, the transaction relationship between exchange addresses and ransomware addresses is critical to modeling the ransomware activity.

To obscure their real identities and the ransom transfer trajectory, criminals usually use different Bitcoin addresses at different stages. Addresses with no more than two transactions are typically one-time ransom transfer addresses. Specifically, Figure 2(b) shows that approximately 18,000 ransomware addresses in our dataset are involved in no more than two transactions, while 21 ransomware addresses participate in more than 100 transactions. The ransomware addresses at different stages exhibit different transaction behavior, and they conspire together to accomplish all three stages of a ransomware activity. Therefore, identifying ransomware addresses requires a focus on transaction relationships between Bitcoin addresses.

To more effectively capture the transaction behavior of ransomware criminals, we design a cascade feature extraction method inspired by Reference [19] to aggregate features of Bitcoin addresses that are involved in the same Bitcoin transaction with the ransomware address. To clarify the cascade feature extraction method clearly, we pre-define four terms for an address.

- **Address-txs Data.** The address-txs data contains all Bitcoin transactions in which a specific address is involved. Each transaction notes the time when the transaction occurs, the

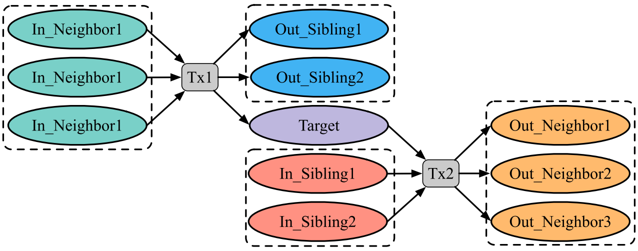


Fig. 3. A target address with two involved Bitcoin transactions (Tx1 and Tx2 are not the mixing transaction).

addresses involved, and the amount attached to each address. Some Bitcoin addresses are involved in only one Bitcoin transaction to receive bitcoins and do not transfer these bitcoins out. To simplify the following description, we will continue to use the term address-txs data to refer to the one Bitcoin transaction.

- **Address Features.** The address features are extracted from the address-txs data, such as the total number of transactions, the active period of an address, and the total bitcoins received. The address features capture the transaction behavior of an address over its entire active period, rather than over a fixed time period (e.g., see Reference [1]).
- **Neighbor and Sibling.** For a given Bitcoin address, we define two transaction relationships according to the address-txs data, namely, *Neighbor* and *Sibling*. As shown in Figure 3, a *Target* address is involved in two Bitcoin transactions. *Neighbor* addresses are on the opposite side of a transaction with the *Target* address, while *Sibling* addresses are on the same side of a transaction with the *Target* address.
- **Cascade Features.** The cascade features of an address are extracted in a cascading manner from its *Neighbors* and *Siblings*, which contain not only features of the address itself but also address features of its *Neighbors* and *Siblings*.

After defining the four key terms, we can now describe the cascade feature extraction method in three steps: self-feature extraction, transaction relationship determination, and cascade feature calculation.

Self-feature Extraction. First, we extract the address-txs data for each Bitcoin address. Then, based on the transaction format for multiple inputs and outputs, we calculate a series of address features that well reflect the differences between ransomware addresses and unlabeled addresses. To explain the meaning of address features and the process of cascade feature extraction clearly, we assign each feature a specific name that intuitively describes its calculation method. More specifically, Table 2 shows the name and the content of each address feature. In total, we extract 18 address features from address-txs data for each Bitcoin address.

Transaction Relationship Determination. Figure 3 depicts a target address with two involved Bitcoin transactions. Taking this as an example, we present the detailed steps of transaction relationship determination. First, we extract the address-txs data of the *Target* address. Then, we traverse each transaction in the address-txs data and determine the transaction relationships between the *Target* address and other Bitcoin addresses according to their positions in the transactions. Since mixing transactions contain addresses that do not know each other's identities, these transactions create a transaction relationship between unrelated addresses, which messes up transaction relationships. Therefore, to avoid confusion, we filter out mixing transactions in the process of transaction relationship determination, but we still take into account mixing transactions in the process of self-feature extraction.

Table 2. Features of Each Address Itself Extracted from Address-txs Data

Feature Name	Content
tx_num	the total number of transactions that the address is participating in
send_num	the number of transactions sending bitcoins to others
rec_num	the number of transactions receiving bitcoins from others
nei_num	the number of neighbors
sib_num	the number of siblings
dec_num	the number of decimals that the transaction amount is accurate to
send_amount	the total amount of bitcoins sent by the address
rec_amount	the total amount of bitcoins received by the address
avg_send_amount	the avg amount of bitcoins sent by the address at one time
avg_rec_amount	the avg amount of bitcoins received by the address at one time
max_sender_num	the maximum number of senders among the address-txs data
min_sender_num	the minimum number of senders among the address-txs data
max_receiver_num	the maximum number of receivers among the address-txs data
min_receiver_num	the minimum number of receivers among the address-txs data
type	the type of Bitcoin address according to Blocksci [42]
locktime	whether the address is involved in locktime transactions, i.e., 0 or 1
mixing	whether the address is involved in mixing transactions, i.e., 0 or 1
active_period	the active period of the address in Bitcoin

For a transaction, we divide all Bitcoin addresses involved in the transaction into two sets: the input set and the output set. The input set includes all addresses that contribute bitcoins to the transaction, while the output set includes all addresses that receive bitcoins from the transaction. According to whether the *Target* address is in the input set or output set of a transaction, we can determine transaction relationships between the *Target* address and related addresses. In the first case, where the *Target* address appears in the input set of a transaction (e.g., Tx2 in Figure 3), we consider all other addresses in the input set as *Sibling* addresses and all addresses in the output set as *Neighbor* addresses. We further refine these *Sibling* addresses as *In_Sibling* addresses (e.g., *In_Sibling3,4*), since they are inputs of the transaction, and we refine these *Neighbor* addresses as *Out_Neighbor* addresses (e.g., *Out_Neighbor4,5,6*), since they are outputs of the transaction.

In the second case where the *Target* address is in the output set of a transaction (e.g., Tx1 in Figure 3), we consider all other addresses in the output set as *Sibling* addresses and all addresses in the input set as *Neighbor* addresses. Similarly, we refine these *Sibling* addresses as *Out_Sibling* addresses (e.g., *Out_Sibling1,2*) and these *Neighbor* addresses as *In_Neighbor* addresses (e.g., *In_Neighbor1,2,3*). By determining these transaction relationships, we can extract cascade features from the *Neighbor* and *Sibling* addresses of the *Target* address.

To summarize, we categorize related addresses into four groups, i.e., *In_Neighbor*, *Out_neighbor*, *In_Sibling* and *Out_Sibling*, based on their transaction relationships with the *Target* address.

Cascade Feature Calculation. After dividing related addresses into the four groups, we conduct statistical calculations on the address features of addresses in each group.

More specifically, Figure 4 shows the detailed process of feature aggregations from these four groups, including *In_Neighbor*, *Out_neighbor*, *In_Sibling*, and *Out_Sibling*. First, we extract 18 address features for each address itself according to the address-txs data. Then, for each address feature except the *type*, *locktime*, and *mixing*, we use four statistical methods to compute the aggregate information of the feature in each group. For example, for the total number of transactions (*tx_num*), we compute the maximum, minimum, average, and standard deviation of

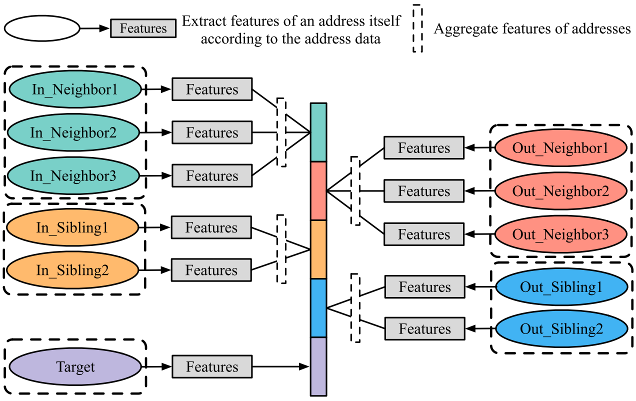


Fig. 4. Cascade feature extraction method for Bitcoin transactions.

In_Neighbors as a part of the cascade features. For the feature *type*, we compute the number of different *types* in a group. For the two features *locktime* and *mixing*, we do not use statistical methods for feature aggregation. We simply define the aggregate information of *locktime* in a group to be 1 if the *locktime* of any one Bitcoin address in the group is 1, otherwise, it is set as 0. The same calculation applies to the features *type* and *mixing*. Feature aggregation operations of the other three groups are defined in the same way. Therefore, we gain 63 (15 * 4 + 3) features for each group.

After the feature aggregation process, we obtain a total of 270 (18 + 4 * (15 * 4 + 3)) cascade features for each address, which consists of 18 features from the address itself and 252 features obtained by aggregating information from related addresses in the four groups. Compared to the six features explored in the previous work [1], the cascade features extracted by our method contain much more information about the interaction of addresses with their related addresses, which are effective in detecting ransomware addresses in Bitcoin.

Extracting the cascade features of addresses in a period mainly involves four steps: extracting features of each address, determining all neighbors and siblings of these addresses, extracting features of neighbor addresses and sibling addresses, and calculating the cascade features. The complexity of the first step is $O(n)$, and that of the subsequent step is also $O(n)$. Assuming an average of k neighbor addresses and sibling addresses per address, the complexity of the third step is $O(kn)$. The complexity of the final step is $O(n)$. Therefore, the total complexity of our method is $O(n + n + kn + n)$, which is again $O(n)$.

During a one-week experiment spanning from December 1, 2023, to December 7, 2023, approximately 911,631 addresses are involved in transactions each day. The average processing time for each day is approximately 20 minutes. It should be noted that the variability in both daily transaction volumes and the number of addresses involved can lead to variations in processing times across different periods. For addresses whose features have been extracted, we can save the features and use them directly later, which can speed up the process of feature engineering.

4.3 Classification Method

Identifying ransomware addresses is essentially building a Bitcoin address classification model, but the task is hindered by the highly imbalanced ratio between the disclosed ransomware addresses and unlabeled addresses. There are around 21,000 disclosed ransomware addresses, just accounting for 0.002% of complete Bitcoin addresses. Although there are some other labeled addresses in Bitcoin, most Bitcoin addresses are unlabeled and involved in various transaction activities. To develop a model that can distinguish ransomware addresses from all other addresses instead of one or a few types of entities, we need a labeled dataset that is representative of the wide range of transaction activities in Bitcoin. Although existing labeled datasets contain addresses involved in various transaction activities, they only represent a small portion of complete Bitcoin addresses, so the transaction behavior in these labeled datasets cannot represent the transaction behavior of all unlabeled addresses. As a result, using only negative samples consisting of existing labeled Bitcoin addresses may not capture all differences in features between ransomware addresses and unlabeled addresses. This data imbalance issue renders many existing methods ineffective. To overcome the imperfect labeling issue, we treat the ransomware address detection task as a PU-learning problem and build a classification model based on the PU-Bagging approach.

Researchers study the issue of imperfect labeling and produce a series of results from three research directions [79], including one-class classification, PU-learning, and self-supervised learning. After considering the nature of the ransomware address detection task mentioned above, we believe that PU-Bagging [55], an approach of PU-learning [29], is the most suitable approach without having enough labeled negative addresses in Bitcoin. First, PU-Bagging is widely used in many applications, where a classifier is trained on the dataset consisting of only positive and unlabeled samples. Besides, PU-Bagging is effective even with a small percentage of positive samples in the dataset [55], which fits well with the situation where ransomware addresses represent a very small portion of complete Bitcoin addresses.

To further validate our consideration, we compare the performance of four specific methods from the three research directions. **One-class support vector machine (OCSVM)** [66] and **support vector data description (SVDD)** [69] are two common one-class classification approaches. Self-supervised learning-based classification methods are proposed for image anomaly detection, and GOAD [6] is a recent work that can be applied to all data types by generalizing the class of transformations. We choose BaggingSVM [55] as an example of the PU-Bagging approach, because it performs well when the number of positive examples is extremely limited and can also run considerably fast, especially when the set of unlabeled examples is large. We evaluate the effectiveness of these four methods focusing on their ability in detecting specific ransomware families, which is an important scenario for ransomware address detection. The experiment procedure is as follows: We choose a specific ransomware family and allocate 80% of its ransomware addresses for training and 20% for testing. In addition, we randomly select 1 million and 10,000 addresses as negative samples for training and testing, respectively. We select the top five ransomware families with the highest number of ransomware addresses as our experimental subjects.

Table 3 shows that BaggingSVM exhibits excellent performance compared to the other three methods, in terms of precision and F1 score. OCSVM and SVDD have lower precision and F1 score than BaggingSVM for every ransomware family. GOAD outperforms BaggingSVM in the classification of CryptXXX and exhibits similar results to XRAD in the case of CryptoWall. However, GOAD demonstrates significantly inferior performance in the classification of other ransomware families. Therefore, we choose the PU-Bagging approach to construct our classification model.

PU-learning is a type of machine learning paradigm that deals with binary classification problems where the negative class (unlabeled data) is only partially observed or completely unobserved. In PU-learning, there are two data types: positive samples (samples from the positive class) and unlabeled samples (samples that may be from either the positive or negative class, but the class label is unknown). PU-learning is to train a classifier that can accurately classify positive samples.

PU-Bagging is a specific approach within PU-learning that combines the concepts of bagging (Bootstrap Aggregating) with PU-learning. PU-Bagging works by generating multiple bootstrap samples from the positive samples and the unlabeled samples. Each bootstrap sample consists of a subset (or all) of positive samples and an equal number of randomly selected unlabeled samples.

Table 3. Performance Comparison of Four Classification Frameworks for Resolving the Data Imbalance Issue

Precision (%)					
Method	Ransomware family				
	CryptoLocker	CryptoWall	CryptXXX	Locky	Cerber
BaggingSVM [55]	**16.63**	**18.34**	**15.82**	**53.82**	**64.29**
OCSVM [66]	4.33	4.86	10.59	27.24	25.35
SVDD [69]	4.91	3.58	7.43	26.81	31.41
GOAD [6]	7.72	18.40	36.86	33.93	24.47
F1 score (%)					
Method	Ransomware family				
	CryptoLocker	CryptoWall	CryptXXX	Locky	Cerber
BaggingSVM [55]	**27.62**	**29.95**	**25.98**	**63.51**	**75.67**
OCSVM [66]	8.11	9.11	18.64	37.86	35.54
SVDD [69]	9.22	6.80	13.38	39.36	43.15
GOAD [6]	14.04	30.56	42.39	47.23	31.64

ALGORITHM 1: PU-Bagging in Ransomware Address Detection

```
Input  : $\mathcal{D}$: Training samples, $\mathcal{N}$: Samples to be predicted, $\mathcal{B}$: Number of base learners
Output : $S$: Predicted samples with labels
$\mathcal{D}_+ \leftarrow$ Ransomware addresses in $\mathcal{D}$;
$\mathcal{D}_- \leftarrow$ Unlabeled addresses in $\mathcal{D}$;
$\mathcal{D}' \leftarrow \emptyset$;
for $i \leftarrow 1$ to $\mathcal{B}$ do
    $\mathcal{D}_i \leftarrow$ Randomly sample $|\mathcal{D}_+|$ addresses from $\mathcal{D}_-$;
    $\mathcal{D}_i \leftarrow \mathcal{D}_i \cup \mathcal{D}_+$;
    // Add ransomware addresses
    $f_i \leftarrow$ TrainClassifier($\mathcal{D}_i$);
    // Train a base classifier with feature sampling
    $\mathcal{D}' \leftarrow \mathcal{D}' \cup \{(f_i, \mathcal{D}_i')\}$;
    // Store classifier and its data
$S \leftarrow \emptyset$;
for $\mathcal{N}_i$ in $\mathcal{N}$ do
    $p \leftarrow \frac{1}{\mathcal{B}} \sum_{j=1}^{\mathcal{B}} f_j(\mathcal{N}_i)$;
    // Aggregate predictions
    Label $\mathcal{N}_i$ with $p$;
    $S \leftarrow S \cup \{\mathcal{N}_i\}$;
return $S$;
```

A classifier is trained on each bootstrap sample, and the final prediction is obtained by aggregating the predictions of all classifiers, typically using a simple voting mechanism. In this way, PU-Bagging aims to improve the robustness and generalization performance of the PU-learning model by averaging over multiple classifiers trained on different subsets of the data.

In our study, we take the disclosed ransomware addresses as positive samples and consider other addresses as unlabeled samples. As shown in Algorithm 1, we use pseudocode to describe how PU-Bagging can be used for ransomware address detection.

Table 4. Performance Comparison of Five Base Models in Ransomware Detection

Precision (%)					
Method	Ransomware family				
	CryptoLocker	CryptoWall	CryptXXX	Locky	Cerber
GBDT [31]	**53.52**	**63.52**	**76.59**	**78.33**	**70.57**
LR [54]	3.77	6.25	2.48	9.74	42.55
SVM [28]	9.03	9.32	8.78	46.10	54.17
LightGBM [44]	7.57	7.28	24.63	59.71	74.46
XGBoost [18]	13.89	7.91	21.80	71.73	43.15
F1 score (%)					
Method	Ransomware family				
	CryptoLocker	CryptoWall	CryptXXX	Locky	Cerber
GBDT [31]	**67.18**	**74.75**	**82.74**	**84.06**	**80.52**
LR [54]	7.08	11.60	4.49	14.55	55.57
SVM [28]	16.01	16.69	15.55	57.32	66.45
LightGBM [44]	13.37	13.35	38.73	67.88	77.28
XGBoost [18]	23.90	14.28	34.68	71.10	57.27

The choice of the base model plays a critical role in the PU-Bagging approach. Several established classification models, including **logistic regression (LR)** [54], **support vector machine (SVM)** [28], and **decision tree (DT)** [64], can serve as the base model. However, among these models, **gradient boosting decision tree (GBDT)** [31] shows favorable results in many problems. GBDT is flexible enough to handle various types of data, whether they are continuous or discrete. So, it is suitable for learning the cascade features we extract including both continuous and discrete values. In addition, the GBDT prediction phase is very efficient because of the parallel operation between trees, so it is suitable for ransomware address detection with fast identification requirements. As shown in Table 4, compared with other models, GBDT performs best in ransomware address detection.

To further corroborate our analysis, we present five combinations of PU-Bagging with various base models. Under the experiment procedure described in Section 5.2, Table 5 shows that GBDT achieves the best performance among the above models. Although LightGBM and XGBoost are variations of GBDT that prioritize speed at the expense of some level of accuracy, our focus is on the accurate detection of ransomware addresses, making GBDT the optimal choice as the base model in the PU-Bagging approach.

The working principle and steps of GBDT for Bitcoin ransomware address detection are as follows: The first step is to initialize the first weak learner F_0, and its expression is $F_0(x) = log\frac{P(Y=1|x)}{1-P(Y=1|x)}$, where P is the proportion of ransomware addresses in the training set. We can use this prior information to initialize the learner.

The second step is to establish M **classification and regression trees (CART)**, where $m = 1, 2, \ldots, M$. GBDT uses the fastest descent approximation method, and the key is to use the negative gradient of the loss function as an approximation of the residual. By using the Log-likelihood loss as the loss function $L(y, f(x)) = log(1 + exp(-yf(x)))$ with $y \in [-1, 1]$, the negative gradient can be simplified as below:

$$r_{m,i} = -\frac{\partial L(y_i, F(x_i))}{\partial F(x)} = y_i - \frac{1}{1 + e^{-F(x_i)}}, where\ F(x) = F_{m-1}(x).$$

Table 5. Performance Comparison of Five Proposed Combinations of PU-Bagging with Various Base Models

Precision (%)					
Method	Ransomware family				
	CryptoLocker	CryptoWall	CryptXXX	Locky	Cerber
GBDT [31]+PU	**78.05**	**83.51**	**84.11**	**99.15**	**98.54**
LR [54]+PU	5.34	7.59	3.04	10.87	58.82
SVM [28]+PU	16.63	18.43	15.82	53.82	64.29
LightGBM [44]+PU	11.11	14.69	48.33	67.86	86.07
XGBoost [18]+PU	18.27	11.07	34.49	82.04	45.58
F1 score (%)					
Method	Ransomware family				
	CryptoLocker	CryptoWall	CryptXXX	Locky	Cerber
GBDT [31]+PU	**87.27**	**90.76**	**90.77**	**99.19**	**99.12**
LR [54]+PU	9.80	13.93	5.45	16.07	69.23
SVM [28]+PU	27.62	30.11	25.98	63.51	75.67
LightGBM [44]+PU	18.77	25.22	64.16	73.06	85.10
XGBoost [18]+PU	30.44	19.41	50.08	80.42	60.94

Using the above method, GBDT replaces the loss function with a log-likelihood function, which allows for the conversion of class predictions into probabilistic predictions. A linear search is then performed to estimate the value of the leaf node area that minimizes the loss function. The learner is then updated continuously through these steps. Ultimately, a final strong classifier is obtained, along with the predicted probability of a Bitcoin address. The classification model can be expressed as $P(Y = 1|x) = \frac{1}{1+e^{-F_M(x)}}$.

5 Evaluation

In this section, we present a series of experiments to evaluate the effectiveness of XRAD for detecting ransomware addresses. Our aim is to provide practical applications of ransomware address detection methods, and we focus on the following three evaluation scenarios: (1) Specific ransomware family detection. If we have a portion of Bitcoin addresses associated with a specific ransomware family, then we aim to use XRAD to identify additional addresses of this same ransomware family; (2) Unknown ransomware family detection. If there is an unknown ransomware family in the past, then XRAD needs to capture generic cascade features of known ransomware families to detect addresses of the unknown ransomware family; (3) Timely ransomware family detection. Since new ransomware families continue to occur, we aim to use XRAD to timely detect the existence of a new ransomware family that appears recently. Finally, we apply XRAD to complete Bitcoin addresses to detect more ransomware activities, thus further revealing the widespread and serious impact of ransomware activities in Bitcoin. Besides, we utilize the permutation importance method to assess the significance of features. It is worth noting that the importance of features may differ across various scenarios.

5.1 Baselines and Metrics

In this section, we describe the baseline methods used in the experiments and the evaluation process.

As shown in Table 6, we select a total of four baseline methods for comparative experiments. Among them, XRAD_base refers to the method that uses only the features of the address itself

Table 6. Four Baselines in Bitcoin Ransomware Address Detection

Name	Description
XRAD_base	XRAD without cascade features.
BitcoinHeist [1]	We contact authors to obtain their code repository and relevant dataset.
Clustering Heuristics [33]	We use the implementation provided by Reference [33], which consists of two heuristics: the *co-spend* heuristic with excluding mixing transactions and the refined *change address* heuristic with considering peel-chain transactions and locktime transactions.
BAClassifier [38]	We modify the code of BAClassifier to detect ransomware addresses in Bitcoin.

without employing cascading features. Additionally, we choose two state-of-the-art studies in the field of Bitcoin ransomware address detection: the widely used clustering heuristics [33] and the latest study BitcoinHeist [1]. Since XRAD extracts cascade features based on Bitcoin transaction relationships, we also compare with methods exploring graph-structured data such as BAClassifier [38].

The construction process of the training set and test set is consistent across all five methods. Therefore, all five methods are evaluated on the same set of addresses during both training and inference, ensuring that each method is evaluated under the same set of disclosed ransomware addresses. In terms of evaluation metrics, we compute the accuracy, recall, and F1 score. For each experiment, we repeat the process 10 times and report the mean of the detection results. Our proposed XRAD is set to have 1,000 tree estimators in all experiments.

In the following experiments, we choose the top five ransomware families based on the number of Bitcoin addresses they contain to demonstrate the performance of the five methods. These chosen ransomware families have a significant number of Bitcoin addresses, while other ransomware families have only a small number of Bitcoin addresses. Additionally, these chosen ransomware families cause serious impacts and affect victims globally, thereby gaining widespread attention.

5.2 Specific Ransomware Family Detection

In practice, we usually cannot gain all Bitcoin addresses controlled by a ransomware family, but only a fraction of them. Given a partial set of Bitcoin addresses of a specific ransomware family, XRAD is capable of learning the cascade features of these addresses to identify additional Bitcoin addresses controlled by the same ransomware family. This ability is crucial for recovering ransoms and accurately measuring the impact of ransomware activities.

We conduct this experiment according to the following procedure. Additionally, we use Table 7 to describe the dataset partitioning process.

(1) Consider a specific ransomware family, denoted as rs, and assume that its active period in Bitcoin, as measured by transactions, falls between t_1 and t_2, where $t_1 < t_2$.
(2) Construct the ground truth dataset $X_{[t_1,t_2]}$ that contains the cascade features and labels of all addresses occurring between t_1 and t_2, including ransomware addresses $X^{rs}_{[t_1,t_2]}$ and unlabeled addresses $X^{u}_{[t_1,t_2]}$.
(3) Sample a ransomware dataset X^{rs}_{train} randomly from $X^{rs}_{[t_1,t_2]}$, where $|X^{rs}_{train}| = 0.8 * |X^{rs}_{[t_1,t_2]}|$.
(4) Sample an unlabeled dataset X^{u}_{train} randomly from $X^{u}_{[t_1,t_2]}$, where $|X^{u}_{train}| = 1,000,000 \gg |X^{rs}_{train}|$.
(5) Regard the rest ransomware addresses as X^{rs}_{test}, where $X^{rs}_{test} = X^{rs}_{[t_1,t_2]} \setminus X^{rs}_{train}$.

Table 7. The Dataset Partitioning for Specific Ransomware Family Detection

Ground Truth	Label	Process	Size	Annotation
$X_{[t_1, t_2]}$	$X^{rs}_{[t_1, t_2]}$	X^{rs}_{train}	0.8 * $\lvert X^{rs}_{[t_1, t_2]} \rvert$	randomly selected from $X^{rs}_{[t_1, t_2]}$
		X^{rs}_{test}	0.2 * $\lvert X^{rs}_{[t_1, t_2]} \rvert$	randomly selected from $X^{rs}_{[t_1, t_2]}$
	$X^{u}_{[t_1, t_2]}$	X^{u}_{train}	1,000,000	randomly selected from $X^{u}_{[t_1, t_2]}$
		X^{u}_{test}	10,000	randomly selected from $X^{u}_{[t_1, t_2]}$

Notes: (1) $[t_1, t_2]$ represents the active period of the specific ransomware family. (2) X represents the cascade features and labels of addresses.

Table 8. Results of Specific Ransomware Family Detection with XRAD

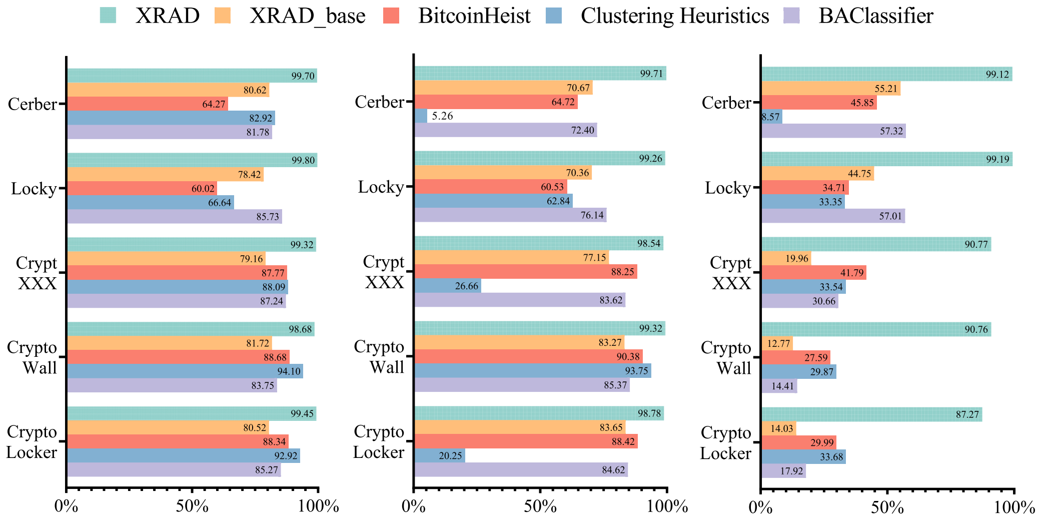
Ransomware	**TP**	**FP**	**TN**	**FN**	**Acc (%)**	**Rec (%)**	**F1 (%)**
CryptoLocker	192	54	9,946	2	99.45	98.78	87.27
CryptoWall	162	32	9,968	1	99.68	99.32	90.76
CryptXXX	344	65	9,935	5	99.32	98.54	90.77
Locky	1,408	12	9,988	11	99.80	99.26	99.19
Cerber	2,027	30	9,970	6	99.70	99.71	99.12

(6) Sample an unlabeled dataset X^u_{test} randomly from $X^u_{[t_1, t_2]}$, where $|X^u_{test}| = 10,000$ and $X^u_{train} \cap X^u_{test} = \emptyset$.
(7) Train a classifier using the dataset $\{X^u_{train} \cup X^{rs}_{train}\}$ to classify the dataset $\{X^u_{test} \cup X^{rs}_{test}\}$.

It is worth noting that an individual classification model is trained for each ransomware family in this scenario. Each model learns the cascade features of a specific ransomware family and detects additional Bitcoin addresses controlled by the family. Besides, the construction process of the training and test sets in the four baseline methods is the same.

Table 8 shows the detection results of five well-known ransomware families. The high accuracy and recall of different ransomware families indicate that XRAD has an outstanding detection performance and can effectively learn the transaction behavior of Bitcoin addresses of each ransomware family. For example, the accuracy of XRAD is 99.80% and the recall is 99.26% when detecting the Locky ransomware family. As ransomware criminals typically withdraw their revenue through cryptocurrency exchanges, the exchange address involved in the transaction with ransomware addresses may be misidentified as a ransomware address. In the case of the CryptoWall ransomware family, we find two of the false positives are BTC-e exchange addresses. In this scenario, the most significant feature for classification is *rec_amount*. This observation can be attributed to the fact that ransomware addresses associated with a specific ransomware family tend to receive similar ransom amounts.

Figure 5 shows the comparison results of XRAD and the four baselines in each ransomware family. The accuracy, recall, and F1 score of XRAD are improved to varying degrees in different ransomware families, compared to the baseline methods. Compared to XRAD_base, XRAD shows an accuracy improvement of approximately 10% to 20% in detecting various ransomware families. This indicates that cascading features can effectively model the characteristics of ransomware activities. Figure 5(b) shows that the recall of the clustering heuristics in Cerber, CryptoXXX, and CryptoLocker is relatively low. These three ransomware families are skilled at concealing their ransom transfer trajectory, which involves instructing victims to transfer the ransom to different Bitcoin addresses. This strategy reduces the likelihood of transactions between ransomware addresses, making it difficult for clustering heuristics to identify additional ransomware addresses. Notably, Cerber assigns each victim a unique Bitcoin address, such that there are no direct transactions between different ransomware addresses.



(a) Accuracy (b) Recall (c) F1 score

Fig. 5. Performance comparison of XRAD and the state-of-the-art methods in specific ransomware family detection.

BitcoinHeist exhibits considerable performance variability in detecting different ransomware families, achieving an accuracy of only 60.02% when detecting Locky. Besides, we find that BitcoinHeist exhibits similar performance on both the training set and the test set, with no significant differences. Therefore, the poor performance on the test set is not due to over-fitting.

BAClassifier demonstrates relatively minor performance variations when detecting different ransomware families, with accuracy all above 80%. BAClassifier is designed to detect four types of addresses: exchange, gambling, mining, and service. These addresses have a high transaction volume. To extract temporal information, BAClassifier divides all transactions of an address into several groups based on timestamps, with each group containing 100 transactions. Each graph contains the information from the 100 transactions in a group. As illustrated in Figure 2(b) of our manuscript, the majority of ransomware addresses have fewer than 10 transactions. Consequently, the design of BAClassifier is not fully suitable for ransomware address detection, resulting in its performance being inferior to that of XRAD.

Table 1 in Section 3 reveals that over half of ransomware families have less than 10 disclosed ransomware addresses. With such a limited number of addresses, it becomes challenging for models to learn the distinct characteristics of ransomware activities from a vast number of unlabeled addresses. Apart from the five ransomware families, we apply XRAD to other ransomware families containing more than 10 addresses (9 ransomware families in total). Among them, the accuracy ranges from 85% to 90% for two ransomware families, from 90% to 95% for four ransomware families, and from 95% to 98% for three ransomware families.

In this scenario, XRAD successfully achieves the purpose of accurately detecting ransomware addresses for a specific ransomware family.

5.3 Unknown Ransomware Family Detection

To evaluate the ability of XRAD to detect unknown ransomware families, we assume that one of the ransomware families in our ransomware address dataset is unknown. Then, we use XRAD to learn the cascade features of all other known ransomware addresses to detect this assumed unknown

Table 9. The Dataset Partitioning for Unknown Ransomware Family Detection

Ground Truth	Label	Process	Size	Annotation
$X_{[t_1, t_2]}$	$X^{mrs}_{[t_1, t_2]}$	X^{mxrs}_{train}	$\lvert X^{mxrs}_{train} \rvert$	$X^{mrs}_{[t_1, t_2]} \setminus X^{rs}_{[t_{1'}, t_{2'}]}$
		X^{rs}_{test}	$\lvert X^{rs}_{test} \rvert$	$X^{rs}_{test} = X^{rs}_{[t_{1'}, t_{2'}]}$
	$X^{u}_{[t_1, t_2]}$	X^{u}_{train}	1,000,000	randomly selected from $X^{u}_{[t_1, t_2]}$
		X^{u}_{test}	10,000	randomly selected from $X^{u}_{[t_1, t_2]}$

Notes: (1) $[t_1,\ t_2]$ represents the active period of all ransomware families in our dataset. (2) $[t_{1'},\ t_{2'}]$ represents the active period of the assumed unknown ransomware family *rs*. (3) X represents the cascade features and labels of addresses.

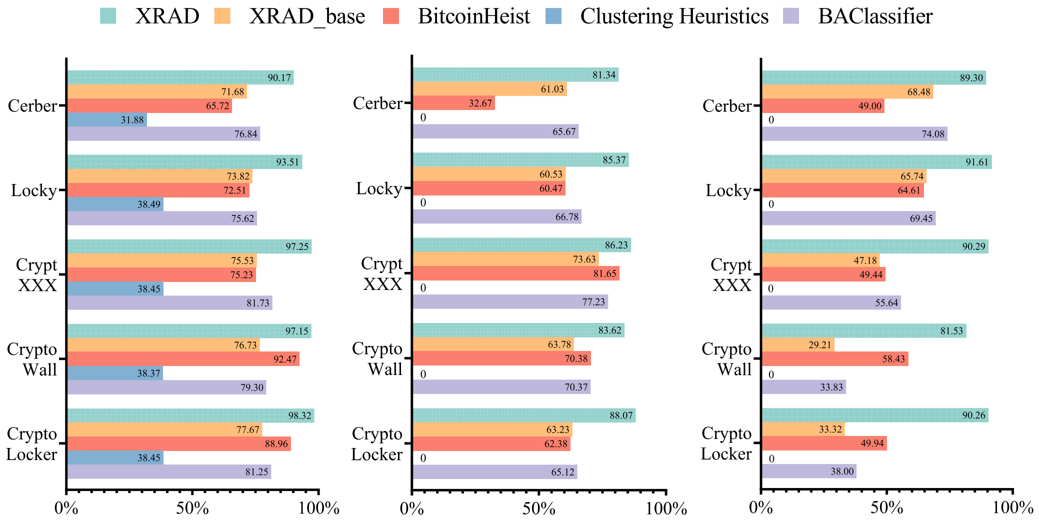
ransomware family. We conduct this experiment according to the following procedure. Additionally, we use Table 9 to describe the dataset partitioning process.

(1) Group all Bitcoin addresses of disclosed ransomware families into one set labeled as Multiple Ransomware (abbreviated as *mrs*) and identify the active time period $[t_1, t_2]$ of the *mrs* through Bitcoin transactions of all disclosed ransomware addresses.
(2) Select a specific ransomware family as the assumed unknown ransomware family, denoted as *rs*, and assume that its active period in Bitcoin, as measured by transactions, falls between $t_{1'}$ and $t_{2'}$, so $t_1 \leq t_{1'} \leq t_{2'} \leq t_2$.
(3) Construct the ground truth dataset $X_{[t_1, t_2]}$ that contains the cascade features and labels of all addresses in $[t_1, t_2]$, including ransomware addresses $X^{mrs}_{[t_1, t_2]}$ and unlabeled addresses $X^{u}_{[t_1, t_2]}$.
(4) Construct a mixed ransomware dataset X^{mxrs}_{train} that contains Bitcoin addresses of all ransomware families except for the *rs*, where $X^{mxrs}_{train} = X^{mrs}_{[t_1, t_2]} \setminus X^{rs}_{[t_{1'}, t_{2'}]}$.
(5) Sample an unlabeled dataset X^{u}_{train} randomly from $X^{u}_{[t_1, t_2]}$, where $|X^{u}_{train}| = 1,000,000$.
(6) Construct a ransomware dataset X^{rs}_{test} that contains all addresses of the *rs*, where $X^{rs}_{test} = X^{rs}_{[t_{1'}, t_{2'}]}$.
(7) Sample an unlabeled dataset X^{u}_{test} randomly from $X^{u}_{[t_{1'}, t_{2'}]} \subseteq X_{[t_{1'}, t_{2'}]}$, where $|X^{u}_{test}| = 10,000$.
(8) Train a classifier using the dataset $\{X^{mxrs}_{train} \cup X^{u}_{train}\}$ to classify the dataset $\{X^{rs}_{test} \cup X^{u}_{test}\}$.

In this scenario, we train one classification model for every assumed unknown ransomware family. This model learns the generic cascade features of all other ransomware families and detects the assumed unknown ransomware family. It is worth noting that, in addition to the ransomware families shown in Figure 6, other ransomware families are also included in the training set to train the model.

Table 10 presents the results of the unknown ransomware family detection on five ransomware families. It is observed that XRAD outperforms existing methods in detecting various ransomware families. For instance, XRAD has the highest accuracy of 98.32% when detecting the CryptoLocker ransomware family. XRAD has a specific design in self-feature extraction and transaction relationship determination to deal with the issue that mixing transactions create a transaction relationship between multiple unrelated users, resulting in better detection results. For example, 162 addresses in the CryptoLocker ransomware family participated in 9,741 mixing transactions. For other ransomware families with fewer disclosed addresses in our dataset, XRAD is also able to achieve accurate detection, even for those only containing one or two.

Figure 6 demonstrates that XRAD performs better than previous studies in detecting various ransomware families. Although the performance of XRAD in this scenario is not as high as in the previous scenario, it is still enough to be practical. Additionally, the clustering heuristic has a recall of zero, which implies that it is only capable of analyzing the transactions of public ransomware addresses to identify additional Bitcoin addresses belonging to known ransomware families, and



(a) Accuracy (b) Recall (c) F1 score

Fig. 6. Performance comparison of XRAD and the state-of-the-art methods in unknown ransomware family detection.

Table 10. Results of Unknown Ransomware Family Detection with XRAD

Ransomware	TP	FP	TN	FN	Acc (%)	Rec (%)	F1 (%)
CryptoLocker	853	69	9,931	115	98.32	88.07	90.26
CryptoWall	599	94	9,906	214	97.15	73.62	79.55
CryptXXX	1,502	83	9,917	240	97.25	86.23	90.29
Locky	6,056	71	9,929	1,038	93.51	85.37	91.61
Cerber	8,268	85	9,915	1,897	90.17	81.34	89.30

that it is ineffective in detecting novel ransomware families. XRAD_base demonstrates good performance in terms of accuracy and recall, but its F1 score is somewhat lower, indicating a certain imbalance between precision and recall. BitcoinHeist exhibits relatively balanced performance, but its F1 score varies significantly, suggesting considerable differences in its effectiveness across different ransomware families. BAClassifier performs well in terms of accuracy and recall, but its F1 score is not as high as that of XRAD, indicating that it may struggle to balance precision and recall in certain situations.

In this scenario, the most significant feature for classification is *nei_num*. The operational stages of ransomware activity can be divided as follows: the collection of ransom payments, their aggregation, and their subsequent withdrawal. Each ransom payment transaction typically pertains to a single victim, ensuring a one-to-one relationship during the ransom collection stage. Subsequently, in the ransom aggregation stage, multiple ransom payments are consolidated into a single ransom address. Similarly, during the ransom withdrawal stage, the ransom address usually engages in trades with just one exchange address for each withdrawal. Consequently, in these three crucial stages, the *nei_num* of most ransomware addresses remains at one, representing the victim address, the aggregation address, and the exchange address, respectively.

Overall, XRAD demonstrates excellent performance in learning the generic cascade features of Bitcoin addresses in multiple ransomware families and has the potential to be applied in real-world scenarios for detecting unknown ransomware activities.

Table 11. The Dataset Partitioning for Timely Ransomware Family Detection

Ground Truth	Label	Period	Process	Size	Annotation
$X_{[t_0, t_2]}$	$X_{[t_0, t_1]}$	$X^{mrs}_{[t_0, t_1]}$	X^{mrs}_{train}	$\lvert X^{mrs}_{train}\rvert$	$X^{mrs}_{train} = X^{mrs}_{[t_0, t_1]}$
		$X^{u}_{[t_0, t_1]}$	X^{u}_{train}	1,000,000	randomly selected from $X^{u}_{[t_0, t_1]}$
	$X_{[t_1, t_2]}$	$X^{rs}_{[t_1, t_2]}$	X^{rs}_{test}	$\lvert X^{rs}_{test}\rvert$	$X^{rs}_{test} = X^{rs}_{[t_1, t_2]}$
		$X^{u}_{[t_1, t_2]}$	X^{u}_{test}	10,000	randomly selected from $X^{u}_{[t_1, t_2]}$

Notes: (1) t_0 represents the time when the first known ransomware family CryptoLocker starts to be active. (2) $[t_1,\ t_2]$ represents the active period of the ransomware family occurring newly. (3) X represents the cascade features and labels of addresses.

5.4 Timely Ransomware Family Detection

To evaluate the ability of XRAD to timely detect a ransomware family occurring newly, we select a target ransomware family and group all ransomware families that occurred before the target ransomware family as the training set. We use XRAD to learn the cascade features of the ransomware families in the training set and then detect addresses of the target ransomware family. We use t_0 to represent the time when the first known ransomware family CryptoLocker starts to be active in Bitcoin. We conduct this experiment according to the following procedure. Additionally, we use Table 11 to describe the dataset partitioning process.

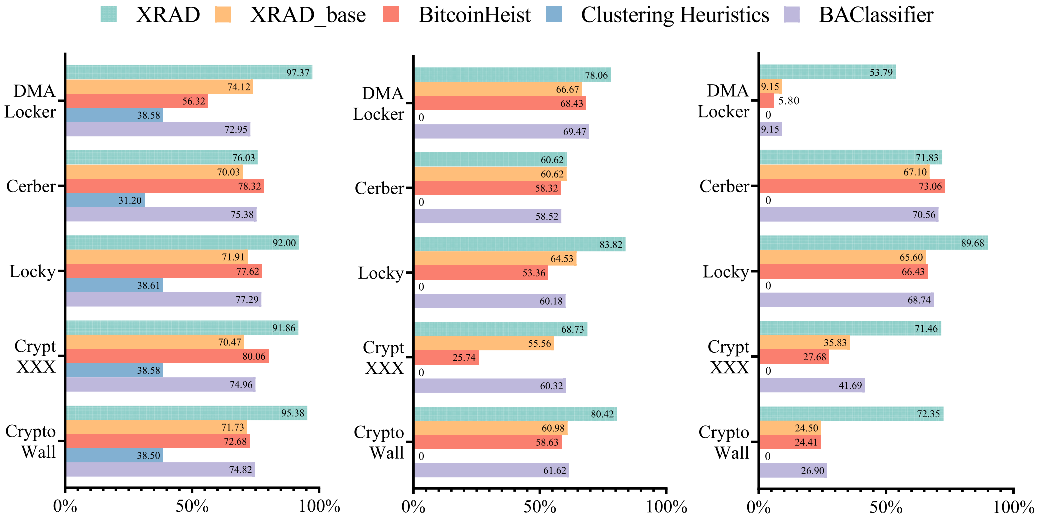
(1) Select a specific ransomware family, denoted as *rs*, and assume that its active period in Bitcoin, as measured by transactions, falls between t_1 and t_2, so $t_0 \leq t_1 \leq t_2$.
(2) Construct a dataset $X_{[t_0, t_1]}$ that contains the cascade features and labels of all addresses occurring between t_0 and t_1, and a dataset $X_{[t_1, t_2]}$ that contains the cascade features and labels of all addresses occurring between t_1 and t_2. These two datasets contain an unlabeled dataset $X^u_{[t_0, t_1]}$ and an unlabeled dataset $X^u_{[t_1, t_2]}$, respectively.
(3) Construct a mixed ransomware dataset $X^{mrs}_{train} \subseteq X_{[t_0, t_1]}$ that contains ransomware addresses of all ransomware families before the ransomware family *rs*.
(4) Sample an unlabeled dataset X^u_{train} randomly from $X^u_{[t_0, t_1]}$, where $|X^u_{train}| = 1{,}000{,}000 \gg |X^{mrs}_{train}|$.
(5) Construct a ransomware dataset $X^{rs}_{test} \subseteq X_{[t_1, t_2]}$ that contains ransomware addresses of the *rs*.
(6) Sample an unlabeled dataset X^u_{test} randomly from $X^u_{[t_1, t_2]}$, where $|X^u_{test}| = 10{,}000$.
(7) Train a classifier using the dataset $\{X^{mrs}_{train} \cup X^u_{train}\}$ to classify the dataset $\{X^{rs}_{test} \cup X^u_{test}\}$.

It is worth noting that we choose to select the DMALocker family with the sixth largest number of addresses, as there are no disclosed ransomware families that appear before CryptoLocker. For each ransomware family, we train one classification model, which learns the cascade features of previous ransomware families and detects the emergence of the target ransomware family.

Table 12 presents the results of timely ransomware family detection on five ransomware families. Compared with the previous two experiments, the performance of XRAD decreases, mainly because, in this scenario, the model can only learn cascade features of ransomware families that occur before the target ransomware family. This may result in the loss of some particular information about the target ransomware family. It is worth noting that XRAD has a relatively poor performance when detecting Cerber. Due to the unique C&C mechanism of the Cerber ransomware family, the transaction behavior of ransomware addresses in Cerber is quite different from ransomware families occurring before it. Our further investigation reveals that the transaction amounts of about 1,000 ransomware addresses in Cerber are below 0.3 bitcoins, which is much less than that of ransomware addresses in other ransomware families.

Table 12. Results of Timely Ransomware Family Detection with XRAD

Ransomware	TP	FP	TN	FN	Acc (%)	Rec (%)	F1 (%)
CryptoWall	654	341	9,659	813	95.38	80.42	72.35
CryptXXX	1,197	411	9,589	545	91.86	68.73	71.46
Locky	5,946	220	9,780	1,148	92.00	83.82	89.68
Cerber	6,162	831	9,169	4,003	76.03	60.62	71.83
DMALocker	156	224	9,776	44	97.37	78.06	53.79



(a) Accuracy (b) Recall (c) F1 score

Fig. 7. Performance comparison of XRAD and the state-of-the-art methods in timely ransomware family detection.

Figure 7 illustrates the superior performance of XRAD over other methods in timely detecting various ransomware families. Similar to the detection results in the previous two scenarios, the performance of XRAD_base is generally about 20% lower than that of XRAD, further highlighting the importance of cascading features. It is evident that the clustering heuristics have zero recall and F1 score, indicating that they are not suitable for timely ransomware family detection. BitcoinHeist exhibits high accuracy but very low recall. In terms of accuracy and recall, BAClassifier is outperformed by XRAD_base. However, when compared to XRAD, BAClassifier demonstrates significantly inferior performance.

XRAD makes great progress in timely detecting ransomware activities possibly occurring in the future. XRAD performs significantly better than other methods in CryptoWall, CryptXXX, and DMALocker, according to the F1 score. However, the accuracy and F1 score of XRAD in Cerber is slightly lower than BitcoinHeist. Despite the relatively low recall due to some ransomware addresses being misidentified as negatives, XRAD can still detect the existence of ransomware families in progress or recently. In cases where some ransomware addresses of an unknown ransomware family are identified, we can use the model trained in Section 5.2 to detect additional ransomware addresses of the same ransomware family.

In our analysis, we identify that *type* is the most significant feature after *nei_num*. This observation can be attributed to the fact that ransomware criminals typically utilize basic transfer functionality to move ransoms, rather than employing multiple individuals to collectively manage the funds or engaging in other complex financial operations. Consequently, the primary ransomware address type is the P2PKH address, which begins with the digit 1, rather than the P2SH address that starts with the digit 3 and supports multi-signature transactions.

5.5 Complete Bitcoin Address Detection

There are still a significant number of ransomware addresses in Bitcoin that remain undisclosed. Specifically, by analyzing ransom notes and encrypted files, it is determined that there are over 600 ransomware families [51]. However, researchers are able to uncover the tracks of a very small portion of ransomware activities in Bitcoin [7, 23, 37, 75]. This underestimation of the serious impact of ransomware activities highlights the need for more effective methods to detect and track ransomware activities.

To comprehensively estimate the impact of ransomware activities, we apply XRAD to complete Bitcoin addresses from January 3, 2009, to December 31, 2023. It is worth noting that the ransomware address detection method is an auxiliary method, and there is a certain false positive rate, so the result cannot be simply used as the final result. As in the previous subsection, we group Bitcoin addresses of all known ransomware families to one set labeled as Multiple Ransomware (abbreviated as *mrs*) and identify the active time period $[t_1, t_2]$ of the *mrs*. Then, we conduct this experiment according to the following procedure:

(1) Construct a dataset $X_{[t_1,t_2]}$ that contains the cascade features and labels of all addresses between t_1 and t_2, including all ransomware addresses $X^{mrs}_{[t_1,t_2]}$ as X^{mrs}_{train} and unlabeled addresses $X^{u}_{[t_1,t_2]}$.
(2) Sample an unlabeled dataset X^{u}_{train} randomly from $X^{u}_{[t_1,t_2]} \subseteq X_{[t_1,t_2]}$, where $|X^{u}_{train}| = 1,000,000 \gg |X^{ars}_{train}|$.
(3) Train a classifier using the dataset $\{X^{mrs} \cup X^{u}_{train}\}$ to classify the other Bitcoin addresses between 2009 and 2023.

Through the above procedure, we detect 5,928,913 Bitcoin addresses related to ransomware activities. Combining these detected addresses with disclosed ransomware addresses, we gain a total of 5,950,399 Bitcoin addresses as *ransomware* addresses out of 1,372,381,215 Bitcoin addresses. To further analyze the detection results, we construct a bipartite transaction graph with complete transactions from January 3, 2009, to December 31, 2023, consisting of 1,372,381,215 address nodes and 973,954,812 transaction nodes. In this graph, nodes representing ransomware addresses and related transaction nodes are labeled as *ransomware*, forming numerous subgraphs containing this label. We assume that each subgraph represents the Bitcoin ransom transfer of an individual ransomware activity.

Considering the sheer volume of Bitcoin addresses and the inherent limitations of the classification model, it is inevitable that there are misclassifications in the detection results (i.e., false positives and false negatives). To make our complete address detection results more reliable, we further process the results in the following three steps:

First, we take into account the possibility that some Bitcoin addresses may be detected as ransomware addresses despite having no association with ransomware activities. To eliminate such false positives, we rely on our prior analysis of ransomware activities to set filter rules. Specifically, we consider that a subgraph comprising solely one address node is not a typical transaction pattern of ransomware activities, as depicted in Subgraph1 of Figure 8. Therefore, we filter out subgraphs that solely comprise one ransomware address node and its associated transaction nodes.

Second, it is plausible that certain Bitcoin addresses associated with ransomware activities may not be identified as such, leading to a fragmented ransom transfer process. Consequently, the

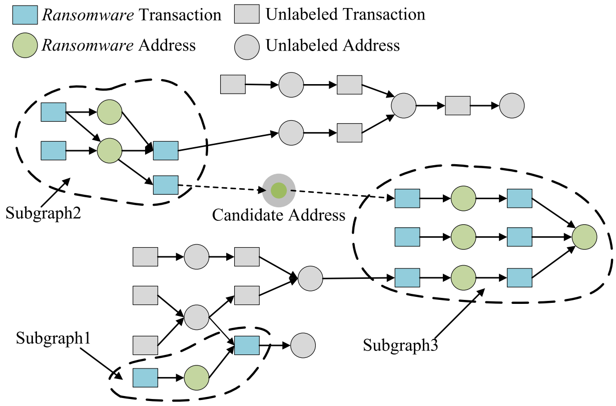


Fig. 8. Three ransomware subgraphs in the transaction graph.

large bipartite transaction graph may contain multiple closely related but disconnected subgraphs that correspond to the same ransomware activity. For instance, as shown in Figure 8, two subgraphs (Subgraph2 and Subgraph3) can be connected if the candidate address is recognized as a *ransomware* address. This implies that these subgraphs jointly represent the complete ransom transfer behavior for an individual ransomware activity, given that ransom transfers between distinct ransomware activities are extremely uncommon. To address this limitation in the detection on complete Bitcoin addresses, we compute the distance between every pair of subgraphs and merge those that are in close proximity. As the large graph is bipartite and all relevant transactions associated with *ransomware* addresses are already labeled as *ransomware*, there is no direct edge linking the two subgraphs. Therefore, the minimum distance between them is two. We then consider two subgraphs belonging to the same ransomware activity if their distance is two, and we merge them by assigning the candidate address node with the *ransomware* label.

Third, after analyzing ransom demands and revenue of ransomware addresses, we find that addresses belonging to the same ransomware activity tend to receive ransom amounts that are similar or identical. Figure 9 depicts the distribution of ransom payments for various ransomware families. Typically, ransomware criminals demand a specific amount of bitcoins or USD, which results in the ransomware address receiving payments that are concentrated around certain values. For example, Cerber demands either 0.75 or 1 bitcoin as a ransom payment, Locky demands varying ransoms at different times, such as 0.5, 1, 1.5, and 2 bitcoins, while CryptoXXX demands a ransom of $500, which is approximately equal to 1.2 bitcoins. Figure 9 shows that the number of transactions paying these specific ransom values is significantly higher, which aligns with the known ransom demands of these ransomware activities. Our analysis shows that victims of the same ransomware activities pay similar amounts in response to the criminals' ransom demands. Therefore, if the ransom amounts charged in a detected ransomware activity vary widely, then it is likely not a ransomware activity. Based on this finding, we further filter the detection results. We first identify the nodes with zero in-degree in each subgraph, which refers to the transactions used for a ransom payment in each ransomware activity. We then extract all the ransom revenue for a ransomware activity based on these transactions. If 70% of the ransom payments are concentrated in an interval with a unit of 0.01 BTC, then we consider the subgraph as a ransomware activity; otherwise, we filter it out.

After applying the filtering and merging operations, we identify 120,335 independent subgraphs containing a total of 1,262,903 Bitcoin addresses associated with ransomware activities. These subgraphs represent 120,335 distinct ransomware activities in Bitcoin. To estimate the total amount of

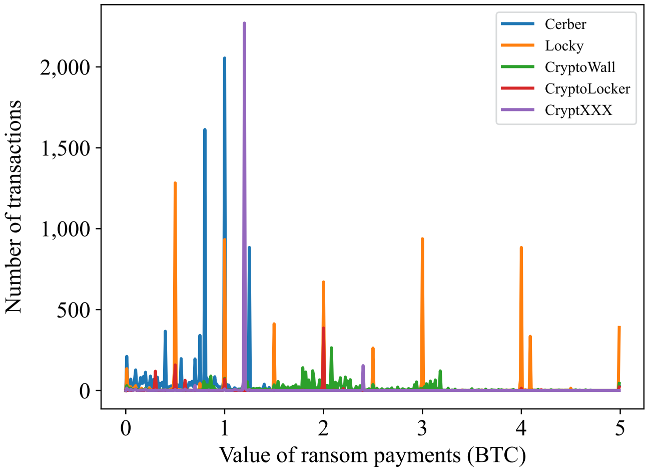


Fig. 9. Distribution of the amount of ransom payments (dividing the ransom amount into intervals of 0.01 BTC).

ransom paid to these ransomware activities, we crawl the historical exchange price of Bitcoin to USD and use the lowest price of each day to calculate the ransom revenue for each activity. In total, our analysis finds that these detected ransomware activities caused a loss of 345,890.56 bitcoins, equivalent to approximately $3,043,136,328.63.

BitcoinAbuse.com is a public database of Bitcoin addresses used by hackers and criminals. We download the dataset that has been collated from this source by other researchers [67]. After analyzing the abuse type and description provided by BitcoinAbuse, we identify a total of 11,712 Bitcoin addresses that are associated with ransomware activities. Out of these addresses, 1,712 Bitcoin addresses have been involved in Bitcoin transactions. Of these addresses, 1,569 addresses appear in our detection results.

We further analyze the detection results to see the trends over time. Table 13 shows the number of ransomware activities, ransom revenue, and average ransom revenue per activity for each year from 2009 to 2023. The CryptoLocker ransomware is the first disclosed ransomware activity, which occurs in 2013. However, there could be earlier ransomware activities that are not disclosed.

We collect analytic reports related to ransom activity from FinCEN [58], Chainalysis [12], and Coveware [24] to corroborate the complete address detection results. The FinCEN [58] is a bureau of the United States Department of the Treasury that collects and analyzes information about financial transactions to combat domestic and international money laundering, terrorist financing, and other financial crimes. Chainalysis [12] is the first start-up company dedicated to the business of Bitcoin tracing. Its customers have included the United States Federal Bureau of Investigation, the Drug Enforcement Administration, and the Internal Revenue Service Criminal Investigation, as well as the United Kingdom’s National Crime Agency. Coveware [24] is a ransomware negotiation company that helps organizations recover from ransomware attacks. Coveware’s ransomware incident response platform plays a critical role in all types and sizes of ransomware infections.

According to the report released by Fincen [30], the total payment of ransomware is about $416 million in 2020 and $590 million in the first half of 2021, which is similar to our result of total ransom revenue. Our result is slightly larger, because the reports and news are based on case information. There may be some undisclosed or unreported ransomware activity not considered.

The annual ransom revenue for 2020, 2021, and 2022 are relatively close to the Chainalysis report [16]. Both the Chainalysis reports and our detection results observe a significant drop in

Table 13. Results of Complete Bitcoin Addresses Detection by Each Year

Year	Number	Ransom (BTC)	Ransom ($)	Average Ransom ($)
2009	0	0	0	/
2010	0	0	0	/
2011	0	0	0	/
2012	854	1,357.46	10,352.69	12.12
2013	10,349	26,342.78	2,532,804.26	244.74
2014	1,817	8,102.09	3,877,467.28	2,133.99
2015	5,831	13,741.05	4,017,636.64	689.01
2016	15,177	31,235.75	18,599,826.20	1,225.53
2017	39,229	96,537.28	286,571,153.94	7,305.09
2018	10,421	41,756.48	252,967,260.37	24,274.76
2019	14,211	39,528.17	513,765,275.32	36,152.64
2020	11,047	58,628.63	597,781,396.87	54,112.56
2021	6,837	14,351.65	821,337,859.71	120,131.32
2022	3,504	9,948.37	415,424,326.89	118,557.17
2023	1,058	4,360.85	126,250,968.46	119,329.84
Total	120,335	345,890.56	3,043,136,328.63	/

annual ransom revenue in 2022. The numbers in these reports increase year-by-year as cases of more ransomware activities are collected and more ransomware addresses are identified, which is also mentioned in the Chainalysis reports [13–16]. Furthermore, Chainalysis emphasizes the true totals can be much higher, as there are Bitcoin addresses controlled by ransomware criminals that have yet to be identified.

In addition, according to the report published by Coveware [24], the average ransom revenue for 2019, 2020, and 2021 is approximately $43,000, $169,000, and $202,000, respectively. Our calculated average ransom revenue is lower than the results reported by Coveware, but it follows the same upward trend. This difference in the calculated averages may be due to the fact that Coveware's cases involve ransomware activities that cause large losses and a smaller number of cases, resulting in a higher calculated average in its report. Table 13 shows an overall increasing trend in annual ransom revenue. However, there are minor fluctuations in 2018 and 2020, which coincides with a drop in Bitcoin's price, making Bitcoin less attractive to ransomware criminals. In contrast, the year 2021 witnesses a significant increase in the average ransom revenue, with Bitcoin's price rising rapidly and staying high, thereby attracting criminals to demand ransoms through Bitcoin.

It is worth noting that the average ransom revenue in 2021, 2022, and 2023 is over $100,000, which is significantly higher than in previous years. For example, CNA Financial pays $40 million in ransom after the Hades ransomware activity in March 2021, and a chemical distributor pays $4.4 million to DarkSide ransomware in May 2021. Furthermore, the FBI confirms that a medical provider in Colorado pays a ransom of $120,000 in Bitcoin after being hacked by the Maui ransomware in April 2022. This also indicates a trend that ransomware activities gradually target large organizations and demand high ransom payments.

6 Discussion

6.1 Feature Engineering vs. Deep Learning in XRAD

We choose to use feature engineering instead of various deep learning algorithms to get features of addresses for several reasons. First, there is an extremely limited number of disclosed ransomware addresses, which makes deep learning prone to overfitting, since they need to learn millions or billions of parameters while building the model [8]. However, combining feature engineering with traditional machine learning has better generalization capabilities than deep learning [20, 80]. Second, interpretability is crucial for the detection results, which deep learning lacks. Deep learning mimics patterns in the data without actually understanding it. Although deep learning eliminates the tedious feature engineering process, it does not negate any data or find hidden biases in the data, making us unaware of the unique characteristics of ransomware activities. Based on our previous work [75], we have a certain level of knowledge about ransomware activities and realize that they have their unique transaction behavior, which we leverage for feature engineering through apparent transaction behavior. Third, extensive experiments in Section 5 show that our method achieves promising results, which is better than the state-of-the-art methods in detecting ransomware addresses.

6.2 Empirical Study of Complete Bitcoin Addresses

To the best of our knowledge, our study is the first to detect ransomware addresses on complete Bitcoin addresses, enabling us to provide a comprehensive overview of the significant impact of ransomware activities. Although it is not feasible to entirely validate our detection results, we are confident in our findings due to the reliable performance of XRAD. Furthermore, we strengthen the credibility of our detection results through two efforts. First, we filter out low-credibility ransomware activities from the perspectives of ransom transfer structure and ransom amount through triple-strict filter rules. Second, we compare our results with some ransomware analysis reports to demonstrate the credibility of the detection results in terms of the annual ransom revenue, average ransom revenue, and their changing trends. The results of the ransomware address detection method are within the scope of academic research and cannot be considered as substantive evidence. Instead, they should be viewed as preliminary findings that require further review and validation before any conclusions or decisions can be made.

6.3 Evolution of Ransomware Activities

In recent years, the number of ransomware activities targeting large organizations is growing. This study [27] highlights the increasing frequency and severity of ransomware activities against large organizations, which can cause significant economic and operational damage. Ransomware criminals often target large organizations with the aim of receiving a high ransom payment, despite the high cost and risk involved. These ransomware activities exhibit different transaction characteristics than previous ones, with criminals using only a few addresses to receive high ransoms, often reaching millions of dollars. To withdraw the ransom more quickly, criminals often use one Bitcoin address to complete all stages of the ransomware activity. However, as there are few disclosed Bitcoin addresses associated with such ransomware activities, the performance of XRAD may decrease when detecting these types of ransomware activities. Therefore, it is essential to continuously update the training data when deploying XRAD to adapt it to these evolving activity patterns.

7 Related Work

In this section, we investigate previous studies closely related to our study from two perspectives. First, we explore a timeline of studies on Bitcoin address classification. Then, we focus on summarizing related studies that detect ransomware addresses in Bitcoin by analyzing ransomware binaries and transaction behavior.

7.1 Bitcoin Address Classification

In the early stages, researchers employ address clustering methods to identify hidden illegal Bitcoin addresses that are associated with disclosed illegal Bitcoin addresses [26]. Classification is widely considered as the predominant method for identifying illegal Bitcoin addresses due to its direct results compared to the clustering method of anomaly detection. Generally, there are two main efforts to improve classification performance: refining feature construction and selecting appropriate model architectures.

In the first category, Lin et al. [50] focus on constructing temporal features and propose four types of transaction moment features to extract the temporal information from transactions. Toyoda et al. [70, 71] construct distinctive features by comparing illegal addresses in HYIP with normal addresses. They find that the two most significant features are the total number of daily transactions and the frequency of the digit 0.3 in USD among received transactions. Hu et al. [35] demonstrate that laundering transactions can be distinguished from regular transactions based on several statistical and network features. These include the in-degree/outdegree ratio, the sum/mean/standard deviation of output values, and the number of weakly connected components, which represents the size of the subgraph to which a transaction belongs. Kanemura et al. [43] propose a method for detecting Bitcoin addresses in darknet markets using a voting-based approach. They extract 73 features for each address and find that the most effective feature for classification is the mean spent PR_{fee}. PR_{fee} refers to the ratio of the difference between the fee of a specific transaction and the average fee within the block containing that transaction, divided by the average fee.

However, some differences among features may be challenging to identify, so effective feature construction alone is insufficient. To address this issue, many classification models are used, including XGBoost, random forest, and artificial neural networks. Bartoletti et al. [4] identify addresses of Ponzi schemes by extracting features such as lifetime, activity days, maximum daily transactions, and the Gini coefficient. Additionally, they compare the performance of three different learning strategies: RIPPER, Bayes network, and random forest. Lin et al. [50] mentioned above employ various machine learning algorithms, including linear regression, SVM, AdaBoost, random forest, XGBoost, LightGBM, and artificial neural network, to train their proposed features. Huang et al. [38] model transaction behavior using a graph neural network and automatically classify Bitcoin addresses into four types: exchange, mining, gambling, and service. Note that they incorporate four features of network centrality: degree centrality, closeness centrality, betweenness centrality, and PageRank centrality. To get better classification results, in this study, we make improvements in both effective feature construction and model construction.

7.2 Analysis of Ransomware Binaries

Many researchers carry out work to detect the emergence of ransomware on users' devices [17, 25, 41, 45, 62] and extract Bitcoin addresses used to collect ransoms [37, 47, 61, 63]. For example, Huang et al. [37] execute each malware sample for up to 20 minutes and then collect the memory dump, created files and screenshots, from which to extract Bitcoin addresses. Kumar et al. [47] find that RTF documents contain the ransom note that gives information about the ransomware address and guides victims for the ransom payment in various languages. Pastrana et al. [61] apply static and dynamic analysis techniques to analyze approximately 4.5 million malware samples, extracting information from the samples, such as Bitcoin addresses and mining pools.

Although these methods can obtain some Bitcoin addresses related to ransomware activities, many of these addresses do not actually receive ransoms, making their analysis of limited value. For instance, WannaCry creates a unique Bitcoin address for each victim to pay the ransom. This means that WannaCry ransomware generates a new Bitcoin address every time it is executed. Consequently, the Bitcoin addresses obtained by executing ransomware binaries in a simulated environment differ from those where victims actually make ransom payments.

7.3 Analysis of Transaction Behavior

Ransomware activities become increasingly rampant as criminals leverage Bitcoin to conceal their real identity when collecting ransom. This practice leads to significant economic losses. So, there are many studies to quantitatively estimate the impact of ransomware activities by analyzing Bitcoin transactions [3, 7, 23, 33, 37, 49, 52, 60]. Specifically, Meiklejohn et al. [52] design the *co-spend* heuristic that regards all addresses in one transaction input controlled by the same user. Androulaki et al. [3] propose the *change address* heuristic that considers the address used to collect back the change belonging to the sender. Liao et al. [49] expand their dataset of ransomware addresses using the above two clustering heuristics and identify 795 ransom payments totaling 1,128.40 bitcoins. Han et al. [33] refine the *change address* heuristic by analyzing peel-chain transactions and locktime transactions. These studies use clustering heuristics to detect hidden ransomware addresses, which may miss many ransomware addresses and cause analysis errors. Wang et al. [74] develop two novel clustering heuristics that utilize unconfirmed transactions in Bitcoin mempool to enhance the clustering effect.

In addition to using the clustering heuristic, Masarah et al. [60] trace outgoing transactions to find more ransomware addresses and estimate the lower-bound direct economic impact of each ransomware family. Wang et al. [75] propose the concept and identification method of the *industry* in Bitcoin and analyze the ransom transfers and victim migrations of disclosed ransomware activities from the industry perspective. Different from References [60, 75], we propose a ransomware address detection method to detect unknown ransomware addresses and ransomware activities in Bitcoin.

There are few studies on extracting address features to build a model to identify ransomware addresses. Akcora et al. [1] introduce the machinery of topological and geometric data analytic tools to ransomware family prediction in Bitcoin. Al-Haija et al. [2] use two supervised machine learning methods to identify transactions related to ransom payments in Bitcoin. These studies focus on transaction information of the address itself or only the network structure information, ignoring transaction relationships between addresses that contain rich activity characteristics.

8 Conclusion and Future Work

In this article, we present XRAD, a novel method for detecting ransomware addresses in Bitcoin. XRAD designs a cascade feature extraction method and incorporates Positive-Unlabeled learning to construct the classification model. Extensive evaluation results demonstrate that XRAD outperforms existing methods in various scenarios, with an average of 15.07% higher accuracy, 19.71% higher recall, and 34.83% higher F1 score. Through the application of XRAD, we detect 120,335 ransomware activities from 2009 to 2023, resulting in a loss of 345,890.56 bitcoins (approximately $3,043,136,328.63). Our study shows that the annual revenue of Ransomware activities in Bitcoin peaks in 2021 and declines annually since then. The average ransom revenue per ransomware activity exhibits an upward trend, also peaking in 2021, and remains relatively stable thereafter. Our study is an essential step toward comprehensively assessing the impact of ransomware activities in the real world. We disclose our code in the repository https://github.com/polarwk/XRAD-in-Bitcoin.

XRAD has an insufficient ability to monitor ransomware addresses in real-time. When a new transaction occurs in Bitcoin, we have to update cascade features of related addresses, which need to recalculate features of the address itself and redefine transaction relationships. To address this limitation, we plan to explore the use of graph neural networks [78] to extract structural information from the transaction graph and enable real-time detection of ransomware addresses. We will also extend our current work to analyze other illegal activities in Bitcoin, such as the money-laundering and Ponzi schemes.

References

[1] Cuneyt Gurcan Akcora, Yitao Li, Yulia R. Gel, and Murat Kantarcioglu. 2020. BitcoinHeist: Topological data analysis for ransomware prediction on the bitcoin blockchain. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI'20)*. ijcai.org, 4439–4445.

[2] Qasem Abu Al-Haija and Abdulaziz A. Alsulami. 2021. High performance classification model to identify ransomware payments for heterogeneous bitcoin networks. *Electronics* 10, 17 (2021), 2113.

[3] Elli Androulaki, Ghassan Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. 2013. Evaluating user privacy in bitcoin. In *Proceedings of the 17th International Conference on Financial Cryptography and Data Security (FC'13)*. Springer, 34–51.

[4] Massimo Bartoletti, Barbara Pes, and Sergio Serusi. 2018. Data mining for detecting bitcoin Ponzi schemes. In *Proceedings of the Crypto Valley Conference on Blockchain Technology (CVCBT'18)*. IEEE, 75–84.

[5] BBC. 2022. Bitcoin Becomes Official Currency in Central African Republic. Retrieved from https://www.bbc.com/news/world-africa-61248809

[6] Liron Bergman and Yedid Hoshen. 2020. Classification-based anomaly detection for general data. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*. OpenReview.net.

[7] Stefano Bistarelli, Matteo Parroccini, and Francesco Santini. 2018. Visualizing bitcoin flows of ransomware: WannaCry one week later. In *Proceedings of the 2nd Italian Conference on CyberSecurity (ITASEC'18)*. CEUR-WS.org.

[8] Lorenzo Brigato and Luca Iocchi. 2020. A close look at deep learning with small data. In *Proceedings of the 25th International Conference on Pattern Recognition (ICPR'20)*. IEEE, 2490–2497.

[9] Jacob Bunge. 2021. JBS Paid $11 Million to Resolve Ransomware Attack. Retrieved from https://www.wsj.com/articles/jbs-paid-11-million-to-resolve-ransomware-attack-11623280781

[10] The United Kingdom's National Cyber Security Centre. 2021. Joint Advisory Highlights Increased Globalised Threat of Ransomware. Retrieved from https://www.ncsc.gov.uk/news/joint-advisory-highlights-increased-globalised-threat-of-ransomware

[11] Yidong Chai, Yonghang Zhou, Weifeng Li, and Yuanchun Jiang. 2022. An explainable multi-modal hierarchical attention model for developing phishing threat intelligence. *IEEE Trans. Depend. Secure Comput.* 19, 2 (2022), 790–803.

[12] Inc. Chainalysis. 2023. Chainalysis: The Blockchain Data Platform. Retrieved from https://www.chainalysis.com

[13] Chainanalysis. 2020. THE 2020 STATE OF CRYPTO CRIME: Everything You Need to Know about Darknet Markets, Exchange Hacks, Money Laundering and More. Retrieved from https://go.chainalysis.com/rs/503-FAP-074/images/2020-Crypto-Crime-Report.pdf

[14] Chainanalysis. 2021. The 2021 Crypto Crime Report: Everything You Need to Know about Ransomware, Darknet Markets, and More. Retrieved from https://go.chainalysis.com/rs/503-FAP-074/images/Chainalysis-Crypto-Crime-2021.pdf

[15] Chainanalysis. 2022. The 2022 Crypto Crime Report: Original Data and Research into Cryptocurrency-based Crime. Retrieved from https://go.chainalysis.com/rs/503-FAP-074/images/Crypto-Crime-Report-2022.pdf

[16] Chainanalysis. 2023. The 2023 Crypto Crime Report: Everything You Need to Know about Cryptocurrency-based Crime. Retrieved from https://go.chainalysis.com/rs/503-FAP-074/images/Crypto_Crime_Report_2023.pdf

[17] Jing Chen, Chiheng Wang, Ziming Zhao, Kai Chen, Ruiying Du, and Gail-Joon Ahn. 2018. Uncovering the face of Android ransomware: Characterization and real-time detection. *IEEE Trans. Inf. Forens. Secur.* 13, 5 (2018), 1286–1300.

[18] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining (KDD'16)*. ACM, 785–794.

[19] Weili Chen, Xiongfeng Guo, Zhiguang Chen, Zibin Zheng, and Yutong Lu. 2020. Phishing scam detection on ethereum: Towards financial security for blockchain ecosystem. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI'20)*. ijcai.org, 4506–4512.

[20] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS'16)*. ACM, 7–10.

[21] Fabrizio Cicala and Elisa Bertino. 2022. Analysis of encryption key generation in modern crypto ransomware. *IEEE Trans. Depend. Secure Comput.* 19, 2 (2022), 1239–1253.

[22] Mauro Conti. 2019. On the Economic Significance of Ransomware Campaigns: A Bitcoin Transactions Perspective. Retrieved from https://spritz.math.unipd.it/projects/btcransomware/

[23] Mauro Conti, Ankit Gangwal, and Sushmita Ruj. 2018. On the economic significance of ransomware campaigns: A Bitcoin transactions perspective. *Comput. Secur.* 79 (2018), 162–189.

[24] Coveware. 2021. 2018–2022 Ransomware Statistics and Facts. Retrieved from https://www.comparitech.com/antivirus/ransomware-statistics/

[25] Simon R. Davies, Richard Macfarlane, and William J. Buchanan. 2021. Differential area analysis for ransomware attack detection within mixed file datasets. *Comput. Secur.* 108 (2021), 102377.

[26] Dominic Deuber, Viktoria Ronge, and Christian Rückert. 2022. SoK: Assumptions underlying cryptocurrency deanonymizations—A taxonomy for scientific experts and legal practitioners. *IACR Cryptol. ePrint Arch.* (2022), 763.

[27] Sudipti Dhawan and Bhawna Narwal. 2019. Unfolding the mystery of ransomware. In *Proceedings of the International Conference on Innovative Computing and Communications (ICICC'19).* Springer, 25–32.

[28] Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alexander J. Smola, and Vladimir Vapnik. 1996. Support vector regression machines. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS'96).* MIT Press, 155–161.

[29] Charles Elkan and Keith Noto. 2008. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th International Conference on Knowledge Discovery and Data Mining (KDD'08).* ACM, 213–220.

[30] FinCEN. 2021. Ransomware Trends in Bank Secrecy Act Data between July 2021 and December 2021. Retrieved from https://www.fincen.gov/sites/default/files/2022-11/Financial%20Trend%20Analysis_Ransomware%20FTA%202_508%20FINAL.pdf

[31] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* 29 (2001), 1189–1232.

[32] GraphSense. 2023. GraphSense Public TagPacks. Retrieved from https://github.com/graphsense/graphsense-tagpacks

[33] Weili Han, Dingjie Chen, Jun Pang, Kai Wang, Chen Chen, Dapeng Huang, and Zhijie Fan. 2021. Temporal networks based industry identification for bitcoin users. In *Proceedings of the 16th International Conference on Wireless Algorithms, Systems, and Applications (WASA'21)*, Vol. 12937. Springer, 108–120.

[34] Julio C. Hernandez-Castro, Eerke Albert Boiten, and Magali F. L. Barnoux. 2014. Second Kent Cyber Security Survey. Retrieved from https://kar.kent.ac.uk/52891/

[35] Yining Hu, Suranga Seneviratne, Kanchana Thilakarathna, Kensuke Fukuda, and Aruna Seneviratne. 2019. Characterizing and detecting money laundering activities on the bitcoin network. *CoRR* abs/1912.12060 (2019).

[36] Danny Yuxing Huang. 2018. Ransomware-public-data. Retrieved from https://hdanny.org/ransomware-public-data/

[37] Danny Yuxing Huang, Maxwell Matthaios Aliapoulios, Vector Guo Li, Luca Invernizzi, Elie Bursztein, Kylie McRoberts, Jonathan Levin, Kirill Levchenko, Alex C. Snoeren, and Damon McCoy. 2018. Tracking ransomware end-to-end. In *Proceedings of the Symposium on Security and Privacy (S&P'18).* IEEE, 618–631.

[38] Zhengjie Huang, Yunyang Huang, Peng Qian, Jianhai Chen, and Qinming He. 2023. Demystifying bitcoin address behavior via graph neural networks. In *Proceedings of the 39th IEEE International Conference on Data Engineering (ICDE'23).* IEEE, 1747–1760.

[39] Newegg Inc. 2022. Using Cryptocurrencies on Newegg. Retrieved from https://kb.newegg.com/knowledge-base/using-crypto-on-newegg/

[40] Aleš Janda. 2021. Bitcoin Block Explorer with Address Grouping and Wallet Labeling. Retrieved from https://www.walletexplorer.com

[41] Manel Jerbi, Zaineb Chelly Dagdia, Slim Bechikh, and Lamjed Ben Said. 2020. On the use of artificial malicious patterns for Android malware detection. *Comput. Secur.* 92 (2020), 101743.

[42] Harry A. Kalodner, Malte Möser, Kevin Lee, Steven Goldfeder, Martin Plattner, Alishah Chator, and Arvind Narayanan. 2020. BlockSci: Design and applications of a blockchain analysis platform. In *Proceedings of the 29th USENIX Security Symposium (USENIX Security'20).* USENIX Association, 2721–2738.

[43] Kota Kanemura, Kentaroh Toyoda, and Tomoaki Ohtsuki. 2019. Identification of darknet markets' bitcoin addresses by voting per-address classification results. In *Proceedings of IEEE International Conference on Blockchain and Cryptocurrency (ICBC'19).* IEEE, 154–158.

[44] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Proceedings of Annual Conference on Neural Information Processing Systems (NeurIPS'17).* 3146–3154.

[45] Amin Kharaz, Sajjad Arshad, Collin Mulliner, William Robertson, and Engin Kirda. 2016. UNVEIL: A large-scale, automated approach to detecting ransomware. In *Proceedings of the 25th USENIX Security Symposium (USENIX Security'16).* USENIX Association, 757–772.

[46] Mark Kolakowski. 2021. El Salvador Becomes Bitcoin Laboratory as First Nation to Adopt It as Legal Tender. Retrieved from https://www.investopedia.com/el-salvador-accepts-bitcoin-as-legal-tender-5200470

[47] M. Satheesh Kumar, Jalel Ben-Othman, and K. G. Srinivasagan. 2018. An investigation on WannaCry ransomware and its detection. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC'18).* IEEE, 1–6.

[48] A. O. Kaspersky Lab. 2021. WannaCry: Are you Safe? Retrieved from https://www.kaspersky.com/blog/wannacry-ransomware/16518

[49] Kevin Liao, Ziming Zhao, Adam Doupé, and Gail-Joon Ahn. 2016. Behind closed doors: Measurement and analysis of CryptoLocker ransoms in Bitcoin. In *Proceedings of the APWG Symposium on Electronic Crime Research (eCrime'16)*. IEEE, 1–13.

[50] Yu-Jing Lin, Po-Wei Wu, Cheng-Han Hsu, I-Ping Tu, and ShihWei Liao. 2019. An evaluation of bitcoin address classification based on transaction history summarization. In *Proceedings of IEEE International Conference on Blockchain and Cryptocurrency (ICBC'19)*. IEEE, 302–310.

[51] MalwareHunterTeam. 2022. MalwareHunterTeam. Retrieved from https://malwarehunterteam.com/

[52] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. 2013. A fistful of bitcoins: Characterizing payments among men with no names. In *Proceedings of the Internet Measurement Conference (IMC'13)*. ACM, 127–140.

[53] Nicole Perlroth Michael D. Shear and Clifford Krauss. 2021. Colonial Pipeline Paid Roughly $5 Million in Ransom to Hackers. Retrieved from https://www.nytimes.com/2021/05/13/us/politics/biden-colonial-pipeline-ransomware.html

[54] Carina Mood. 2009. Logistic regression: Why we cannot do what we think we can do, and what we can do about it. *Eur. Sociol. Rev.* 26, 1 (2009), 67–82.

[55] Fantine Mordelet and Jean-Philippe Vert. 2014. A bagging SVM to learn from positive and unlabeled examples. *Pattern Recog. Lett.* 37 (2014), 201–209.

[56] Satoshi Nakamoto. 2009. Bitcoin: A Peer-to-peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf

[57] nopara73. 2020. Ransomware-public-data. Retrieved from https://github.com/nopara73/WasabiVsSamourai/

[58] The United States Department of the Treasury. 2023. Finanical Crimes Enforcement Network. Retrieved from https://www.fincen.gov

[59] Masarah Paquet-Clouston. 2018. Ransomware in the Bitcoin Ecosystem. Retrieved from https://github.com/behas/ransomware-dataset/

[60] Masarah Paquet-Clouston, Bernhard Haslhofer, and Benoit Dupont. 2019. Ransomware payments in the Bitcoin ecosystem. *J. Cybersec.* 5, 1 (2019), tyz003.

[61] Sergio Pastrana and Guillermo Suarez-Tangil. 2019. A first look at the crypto-mining malware ecosystem: A decade of unrestricted wealth. In *Proceedings of the Internet Measurement Conference (IMC'19)*. ACM, 73–86.

[62] Michal Piskozub, Fabio De Gaspari, Freddie Barr-Smith, Luigi V. Mancini, and Ivan Martinovic. 2021. MalPhase: Fine-grained malware detection using network flow data. In *Proceedings of the ACM Asia Conference on Computer and Communications Security (AsiaCCS'21)*. ACM, 774–786.

[63] Stijn Pletinckx, Cyril Trap, and Christian Doerr. 2018. Malware coordination using the blockchain: An analysis of the cerber ransomware. In *Proceedings of IEEE Conference on Communications and Network Security (CNS'18)*. IEEE, 1–9.

[64] J. Ross Quinlan. 1986. Induction of decision trees. *Mach. Learn.* 1, 1 (1986), 81–106.

[65] Fergal Reid and Martin Harrigan. 2011. An analysis of anonymity in the bitcoin system. In *Proceedings of the 3rd International Conference on Social Computing (SocialCom'11)*. IEEE, 1318–1326.

[66] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alexander J. Smola, and Robert C. Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural Comput.* 13, 7 (2001), 1443–1471.

[67] Sergio Serusi. 2020. BitcoinAbuse Dataset. Retrieved from https://doi.org/10.7910/DVN/SMPQBQ

[68] Guosong Sun and Quan Qian. 2021. Deep learning and visualization for identifying malware families. *IEEE Trans. Depend. Secure Comput.* 18, 1 (2021), 283–295.

[69] David M. J. Tax and Robert P. W. Duin. 2004. Support vector data description. *Mach. Learn.* 54, 1 (2004), 45–66.

[70] Kentaroh Toyoda, P. Takis Mathiopoulos, and Tomoaki Ohtsuki. 2019. A novel methodology for HYIP operators' bitcoin addresses identification. *IEEE Access* 7 (2019), 74835–74848.

[71] Kentaroh Toyoda, Tomoaki Ohtsuki, and P. Takis Mathiopoulos. 2018. Multi-class bitcoin-enabled service identification based on transaction history summarization. In *Proceedings of the International Conference on Internet of Things (iThings'18) and Green Computing and Communications (GreenCom'18) and Cyber, Physical and Social Computing (CPSCom'18) and Smart Data (SmartData'18)*. IEEE, 1153–1160.

[72] Rohit Valecha, Pranali Mandaokar, and H. Raghav Rao. 2022. Phishing email detection using persuasion cues. *IEEE Trans. Depend. Secure Comput.* 19, 2 (2022), 747–756.

[73] Cheng Wang and Hangyu Zhu. 2022. Representing fine-grained co-occurrences for behavior-based fraud detection in online payment services. *IEEE Trans. Depend. Secure Comput.* 19, 1 (2022), 301–315.

[74] Kai Wang, Yakun Cheng, Michael Wen Tong, Zhenghao Niu, Jun Pang, and Weili Han. 2024. Exploring unconfirmed transactions for effective bitcoin address clustering. In *Proceedings of the ACM Web Conference (WWW'24)*. ACM, 1880–1891.

[75] Kai Wang, Jun Pang, Dingjie Chen, Yu Zhao, Dapeng Huang, Chen Chen, and Weili Han. 2022. A large-scale empirical analysis of ransomware activities in bitcoin. *ACM Trans. Web* 16, 2 (2022), 1–29.

[76] Alexandra Winter. 2021. Use Crypto to Buy Dad a Father's Day Gift This Year. Retrieved from https://www.jomashop.com/blog/articles/you-can-use-bitcoin

[77] Lei Wu, Yufeng Hu, Yajin Zhou, Haoyu Wang, Xiapu Luo, Zhi Wang, Fan Zhang, and Kui Ren. 2021. Towards understanding and demystifying bitcoin mixing services. In *Proceedings of the Web Conference (WWW'21).* ACM/IW3C2, 33–44.

[78] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 32, 1 (2021), 4–24.

[79] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. 2021. Generalized out-of-distribution detection: A survey. *CoRR* abs/2110.11334 (2021).

[80] Yang Zhao, Hao Zhang, and Xiuyuan Hu. 2022. Penalizing gradient norm for efficiently improving generalization in deep learning. In *Proceedings of the International Conference on Machine Learning (ICML'22).* PMLR, 26982–26992.

Received 1 August 2023; revised 20 July 2024; accepted 22 July 2024