SAPIENZA
UNIVERSITÀ DI ROMA

Uncovering Threats in Digital Systems: A Deep Dive into BGP, the Blockchain, and Telegram

Department of Computer Science
Computer Science (XXXVI cycle)

Francesco Sassi
ID number 1661522

Advisor
Prof. Alessandro Mei

Academic Year 2023/2024

Thesis defended on May 28th, 2024
in front of a Board of Examiners composed by:

Lamberto Ballan, Associate Professor, University of Padova (chairman)
Giovanni Petri, Professor, Northeastern University London
Alessandro Raganato, Assistant Professor, University of Milano-Bicocca

Reviewers:

Giuseppe Bianchi, Full Professor, University of Roma Tor Vergata
Bernhard Haslhofer, Senior Scientist, Complexity Science Hub Vienna

Uncovering Threats in Digital Systems: A Deep Dive into BGP, the Blockchain, and Telegram

PhD thesis. Sapienza University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: sassi@di.uniroma1.it

Contents

Chapter 1

Introduction

Nowadays, digital systems are becoming increasingly integrated into our everyday lives. This transformation arguably began in the late 90s [203], with the widespread adoption of the Internet. This global platform revolutionized the exchange of information, allowing people to share and access knowledge like never before. [238]. Moreover, it paved the way for other innovations like messaging platforms and social networks that have revolutionized communication by enabling instant global connectivity and interaction. Another more recent innovation is the blockchain technology, which allows transferring funds between people without the need to trust traditionally centralized systems like banks [269]. This technology is already impacting several domains [265, 25, 22, 397, 383, 72] and is slowly gaining recognition by governments, with El Salvador becoming the first country to adopt Bitcoin as a legal tender [151]. These technologies share a common thread: they collectively simplify our daily lives, making information easily accessible, communication instantaneous, and transactions transparent and secure. In doing so, they have become not just tools but integral components of modern life.

However, the profound impact of these innovations introduced novel risks. The Internet is vulnerable to attacks compromising communication and redirecting users to malicious websites that hinder their privacy and personal information [90, 321]. Messaging platforms introduce an additional threat dimension, serving as a medium for manipulating people through disseminating fake news and misinformation [340, 295, 138, 347]. Lastly, blockchain technology opens the door to financial frauds already present in the stock market [391, 240, 368, 47] or brand new ones [253, 241, 114]. In this setting, it is critical to ensure the resilience and safety of these digital platforms to protect the functionality they provide to people and organizations. This thesis is a step in this direction, aiming to analyze, measure, and ultimately propose potential solutions to vulnerabilities of these critical systems.

In Chapter 3, we focus on the security of the Border Gateway Protocol (BGP), the de-facto routing protocol of the Internet. We propose a novel technique incorporating Internet control plane data and administrative information to provide a novel lens to study Internet traffic hijacks and routing misconfigurations. In the first part of our work, we build administrative lifetimes of ASes, collecting, restorating, and polishing RIRs' publicly provided information about ASN allocations. Then, we build the operational lifetimes of ASes by collecting, analyzing, and aggregating over 17 years of BGP data. Finally, we perform a joint analysis of the two lifetimes, creating a taxonomy of the possible behaviors. We find that this combined lens can be used to detect hijack events and router misconfigurations. The work presented in this chapter has been published in a scientific paper titled *"The Parallel Lives of Autonomous Systems: ASN Allocations vs. BGP"* [274], accepted at the Internet Measurement Conference 2021 (IMC 2021).

In Chapter 4, we move to the analysis of frauds that exploit some vulnerabilities of the blockchain ecosystem. In particular, we perform an in-depth analysis of two market manipulations organized by online communities: The pump and dump and the crowd pump. First, we focus on studying pump and dump schemes, a fraud born in the stock market that gained new popularity in the loosely regulated market of cryptocurrencies. We monitor more than 20 Telegram channels for over 3 years, detecting around 900 pump and dumps events. We leverage our unique dataset to build a machine learning model to detect pump and dumps. Then, we characterize crowd pumps. This new phenomenon hit the news in the first months of 2021 when a Reddit community inflated the GameStop stock (GME) price by over 1,900% on Wall Street, the world's largest stock exchange. The operation was replicated on the cryptocurrency market, targeting the DogeCoin (DOGE) and Ripple (XRP) cryptocurrencies. We reconstruct how these operations developed and discuss differences and analogies with the standard pump and dump. We also validate that the machine learning model to detect pump and dumps can effectively detect these events. The work presented in this chapter has been published in a scientific paper titled *"The Doge of Wall Street: Analysis and Detection of Pump and Dump Cryptocurrency Manipulations"* [233], accepted in the Transactions on Internet Technology (TOIT) in 2023.

In Chapter 5, we deepen our study of cryptocurrency markets, focusing on the emerging phenomenon of Decentralized Finance (DeFi). We perform a longitudinal analysis of the BNB Smart Chain and Ethereum blockchain from their inception to March 2022. We study the ecosystem of the tokens and liquidity pools, highlighting analogies and differences between the two blockchains. To characterize tokens, we define and study their lifetime, defined as the time between their creation and the last time they are active in the blockchain. Moreover, we also find that a small group of addresses creates an anomalous number of tokens. Analyzing these tokens, we find that they are often used to perform a particular type of fraud called *1-day rug pull.* We quantify the presence of this operation on both blockchains discovering its prevalence in the BNB Smart Chain. Finally, we present sniper bots, a new kind of trader bot involved in these activities, and we detect their presence and quantify their activity in the rug pull operations. The work presented in this chapter has been published in a scientific paper titled *"Token Spammers, Rug Pulls, and Sniper Bots: An Analysis of the Ecosystem of Tokens in Ethereum and in the Binance Smart Chain (BNB)"* [80], accepted at the USENIX Security Symposium in 2023 (USENIX 2023).

In Chapter 6, we delve into the analysis of sniper bots, finding that they are automated tools designed to buy tokens as soon as they are listed on the market. We leverage GitHub open-source repositories to study them in depth by analyzing their features and how they are implemented. Then, we build a dataset of Ethereum and BNB Smart Chain (BSC) liquidity pools to identify addresses that serially take advantage of sniper bots. We estimate the number of operations they perform, their success rate, and their gains. Finally, we analyze token smart contracts to identify mechanisms that can hinder sniper bots. The work presented in this chapter has been published in the Companion Proceedings of the ACM Web Conference 2023 as *"Ready, Aim, Snipe! Analysis of Sniper Bots and their Impact on the DeFi Ecosystem"* [79].

Finally, in Chapter 7, we examine the risks for users associated with the proliferation of conspiracy theories on social media platforms. To perform our study, we focus on Telegram, a popular instant messaging platform with fewer content limitations than major social media. In this work, we propose an approach to detect conspiracy channels. Then, we discover that conspiracy channels can be clustered into four distinct communities comprising over 17,000 channels. Then, we uncover the "Conspiracy Money Machine," revealing how most conspiracy channels seek to profit from their subscribers. We find conspiracy theorists leverage e-commerce platforms to sell questionable products or lucratively promote them through affiliate links. Moreover, we observe that conspiracy channels use donation and crowdfunding platforms to raise funds. We determine that this business involves hundreds of donors and generates a turnover of over $90 million. The work presented in this chapter is part of the scientific work: *"The Conspiracy Money Machine: Uncovering Telegram's Conspiracy Channels and their Profit Model"* [199], currently under review in an international security conference.

Chapter 2

Background

2.1 Border Gateway Protocol (BGP)

The Internet is a network of more than 70 thousand smaller interconnected networks called Autonomous Systems (AS) [274]. ASes are independent administrative entities that manage a collection of IP prefixes and present a clearly defined routing policy to the Internet [189]. This is possible thanks to the BGP protocol [307], the de facto standard inter-AS routing protocol in today's Internet [225]. In the following, we describe the two building blocks of the BGP protocol, IP prefixes (§ 2.1.1) and Autonomous Systems Numbers (ASN) (§ 2.1.2). Then, we report how the protocol works (§ 2.1.3) and a possible attack on its security (§ 2.1.4).

2.1.1 IP prefixes

Since the release of BGP-4 (RFC4271 [307]), BGP propagates IP reachability information using the Classless Inter-Domain Routing (CIDR) notation [150]. CIDR is a compact way to represent blocks of contiguous IP addresses using *IP prefixes*. An IP prefix is composed of two parts separated by a '/' character. Given the prefix:

$$x.y.z.w/n$$

The IP x.y.z.w is called *network address*, and the number n is used to build a 32 bitmask, called the *network mask* with n leading ones and $32 - n$ trailing zeroes. The two components represent a continuous range of IP prefixes. The first IP of the range can be obtained by applying the bitwise AND operation between the network mask and the IP represented in binary. The number of *host addresses* is computed by applying 2 to the number of zeroes in the mask. A concrete example of an IP prefix is:

$$192.168.1.0/24$$

In this case, the first IP of the prefix is 192.168.1.0, and the number of addresses is 256. Thus, this mask represents all the IP addresses from 192.168.1.0 to 192.168.1.255. A CIDR prefix can strictly contain another. We will refer to the contained prefix as *more specific* and the containing prefix as *less specific*. The CIDR notation also works in the same way for IPv6 prefixes, with the only difference being that the mask has a size of 64 bits.

2.1.2 Autonomous System Number (ASN)

Autonomous Systems (AS) are identified in BGP by a unique identifier known as Autonomous System Number (ASN). Initially, AS numbers were 16 bits long, allowing the creation of 65,536 distinct ASNs. However, due to the expansion of the Internet and the increasing need for ASNs by various organizations, the 16-bit ASN pool started exhausting [182]. The Internet Assigned Numbers Authority (IANA) introduced 32-bit ASNs to tackle this problem [375]. This extended the range significantly, allowing the creation of approximately 4.3 billion unique ASNs.

IANA reserved some ASNs for special use [194], meaning that they cannot be assigned to organizations to be used in BGP. Here is the list of ASNs reserved for special use:

- **Reserved ASNs**: The first and last ASNs of the original 16-bit integers (0 and 65,535) and the last ASN of the 32-bit numbers (4,294,967,295) are reserved and should not be used by operators [183, 224].

- **AS 112**: ASN 112 is reserved for a project that handle reverse DNS lookup queries for private-only use addresses that should never appear in the public DNS system [11].

- **AS_TRANS**: AS 23456 (AS_TRANS) is reserved to facilitate the transition to 32-bit ASNs without causing compatibility issues with routers that only support 16-bit ASNs. When a router advertises a path to a neighbor that does not support 32-bits ASNs it can add the AS_TRANS ASN instead of adding its own 32-bit ASN. [375]

- **Documentation ASNs:** ASNs 64,496-64,511 of the original 16-bit AS range and 65,536-65,551 of the 32-bit range are reserved for use in documentation and sample code [160].

- **Private use ASNs** ASNs 64,512-65,534 of the original 16-bit AS range, and 4,200,000,000-4,294,967,294 of the 32-bit range are reserved for Private Use [263]. Private ASNs are particularly useful when an organization operates multiple autonomous systems or when they need to segregate routing information within their network. Organizations can control their internal routing by using private ASNs while using globally unique ASNs for external BGP peering with other organizations and networks.

The IANA distributes available ASNs to Regional Internet Registries (RIRs), organizations responsible for Internet resource (IPs and ASN) allocation within specific geographic regions. There are five RIRs worldwide divided as follows:

- **AfriNIC** is the regional Internet registry for Africa.

- **APNIC** is the regional Internet registry for the Asia-Pacific region.

- **ARIN** is the regional Internet registry for the United States, Canada, and many Caribbean and North Atlantic islands.

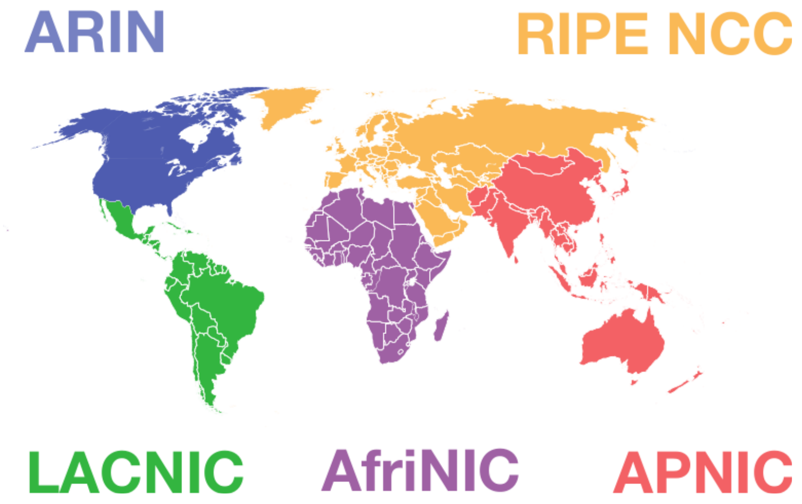


Figure 2.1. The five Regional Internet Registries (RIRs) manage Internet number resources in their respective regions of competence.

- **LACNIC** is the regional Internet registry for the Latin American and Caribbean regions.
- **RIPE NCC** is the regional Internet registry for Europe, the Middle East, and parts of Central Asia.

Fig. 2.1 depicts the different geographic areas covered by each RIR.

2.1.3 The BGP protocol

The BGP protocol enables the exchange of reachability information between Autonomous Systems. Indeed, ASes can use BGP to announce the CIDR IP they manage to other ASes on the Internet. BGP is a path vector protocol, meaning that routing is handled keeping the sequence of Autonomous System Numbers (ASNs) that must be traversed to reach each prefix.In the following, we report some of the core steps of the BGP protocol:

- When an AS wants to be connected to the network its BGP routers establish TCP connections on port 179 with the router of neighbors Autonomous Systems (ASes). There is a phase where routers agree on parameters, such as the version of BGP to use and the IP address for peering.
- Once the BGP neighbors are established, routers exchange BGP routing updates. Each AS can advertise the IP address prefixes they are responsible for to their peers. These advertisements are known as BGP update messages. When a route reaches the router of an AS, it checks if its corresponding ASN is present in the path. If it is present, it rejects the route to prevent routing loops.
- BGP routers use a decision process to determine the best path to reach a specific destination. One of the main factors is the length of each path. Indeed, when two paths are available to a given prefix, it is preferred the shortest AS

path. However, other policies can vary according to the preferences of each AS.

- The BGP router selects the best path based on the decision process and installs it in its routing tables. When an AS is required to forward traffic toward another AS it makes a decision based on this routing table. If there is more than one valid route for a given IP, it should forward the traffic to the route with the shortest path.

- BGP routers continuously exchange updates as the network topology changes. The BGP router advertises the best path to its BGP neighbors.

If an AS advertises one or more IP prefixes, we will refer as an *origin* AS. If it announces routes in which it is not the origin of the path, we will refer to it as *transit* AS. An AS can be used as a transit for one or more routes and as the origin for others.

2.1.4 BGP Hijacking

BGP does not have by-design mechanisms of origin or path validation. Consequently, it assumes an implicit mutual trust among ASes, meaning that each AS announces legitimate paths and prefixes [207, 329]. This mechanism has facilitated various attacks on the protocol, particularly the so-called BGP prefix hijacks [90, 328]. Prefix hijacking is the act of diverting traffic directed to another AS through the propagation of BGP routes [328]. When an AS announces a route to IP prefixes that it does not control, this announcement, if not filtered, can spread and be added to routing tables in BGP routers across the Internet. Until somebody notices and corrects the routes, traffic to those IPs will be routed to that AS. Prefix hijacking can be used to create black holes [166], and perform spam campaigns or phishing [371, 204]. A notorious example is an attack performed in April 2018 where a malicious actor performed a BGP hijack by announcing IP prefixes belonging to AS16509, the host of Amazon Web Services' (AWS) DNS service. The attackers aimed to redirect traffic to a malicious DNS server, directing users to a counterfeit version of the *myetherwallet.com* website. Consequently, users attempting to access a cryptocurrency site were unknowingly redirected to a fraudulent version, leading to over $160,000 worth of cryptocurrencies being transferred to the hackers' wallets [369].

In the following, we present two of the most common categories of BGP hijacks:

- **MOAS:** A MOAS occurs when two or more different ASes originate the same prefix. In this case, the attacker can try to provide a shorter route to certain blocks of IP addresses.

- **SubMOAS:** In the SubMOAS attack, an AS announces a most specific prefix of another AS. In this case, the attacker tries to exploit that a BGP router always forwards the traffic toward the most specific prefix.

Figure 2.2 shows an example of these two attacks. A MOAS occurs when two or more different ASes originate the same prefix. In this example, AS4 announces the same prefix as AS3, the legitimate origin of the prefix 192.168.100/22. AS1 knows

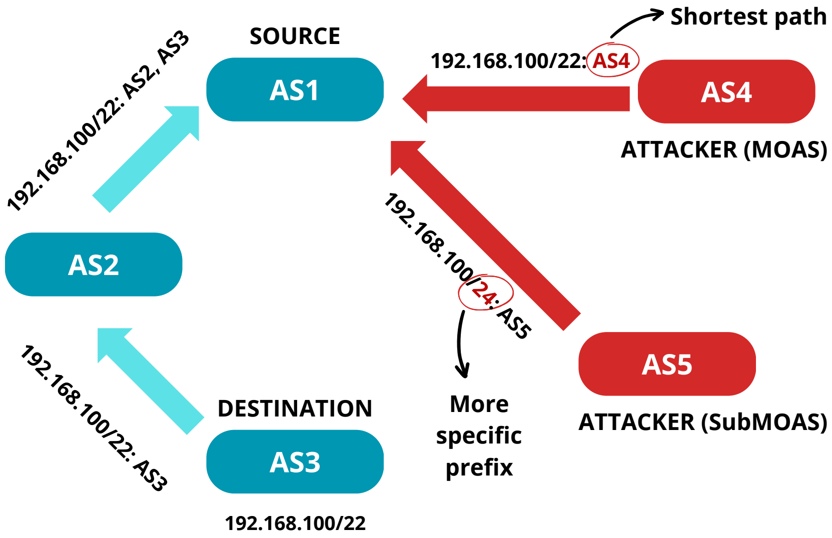


Figure 2.2. In this figure, AS4 performs a MOAS hijack by announcing the same prefix as AS3 (the legitimate origin). Instead, AS5 performs a subMOAS hijack by announcing a more specific prefix than the one originated by AS3.

the route 192.168.100/22:AS2,AS3, that was received from AS2. During the attack, AS1 receives a new route from AS4 (192.168.100/22:AS4) which is the shortest. Thus, AS1 will route its traffic directed to 192.168.100/22 to AS4. Since BGP does not natively support a mechanism of origin validation, AS1 cannot know that AS3 is the legitimate origin and will forward the traffic coming toward 192.168.100/22 to the shortest path, which is the one toward AS4.

The Figure also shows a variation of this attack, called the SubMOAS attack. In this case, AS5 is the attacker and announces a prefix more specific than the one advertised by AS3. Since the BGP router always forwards the traffic toward the most specific route, a part of the traffic that should be directed to AS3 is routed toward AS4.

2.2 Blockchain

The blockchain was proposed by Satoshi Nakamoto in 2008 to execute and record the transactions of the Bitcoin cryptocurrency [269]. The core idea behind the blockchain is to create a digital ledger or record-keeping system, where the information is spread across many computers, often referred to as nodes, forming a decentralized network. Since then, its applications have expanded far beyond. It's now used in various industries for secure and transparent record-keeping, supply chain management, and more. This was possible because the design of the blockchain provides some good properties (open, distributed, immutable, and trustless) but also thanks to the introduction of programmable blockchains, like Ethereum [73], that provide smart contracts. Indeed, using smart contracts allows the creation of decentralized applications that operate on trustless, transparent, and automated protocols. This facilitated the development of various applications, most notably Decentralized Finance (DeFi), which represents a financial system operating without traditional intermediaries like banks. Leveraging smart contracts, a range of financial services, including lending, borrowing, trading, and yield farming, can be automated and executed on blockchain platforms. This transformation significantly impacted the cryptocurrency trading landscape, introducing Decentralized Exchanges (DEXes) powered by smart contracts. These platforms are challenging Centralized Exchanges (CEXes), which are historically the primary platforms for cryptocurrency trading.

In the following subsections, we will explore these aspects, describing Ethereum and BNB Smart Chain 2.2.1 and their smart contracts. Then, we will focus on cryptocurrency trading, analyzing Centralized Exchanges 2.2.3 and Decentralized Exchanges 2.2.4.

2.2.1 Ethereum

Ethereum, created by Vitalik Buterin in 2014 [73], is one of the most popular blockchains. Its native cryptocurrency, Ether (ETH) is the second cryptocurrency by market capitalization, with more than 210 billion US dollars. Ethereum has become very popular as it is one of the first blockchains to be "programmable". Indeed, unlike Bitcoin, Ethereum is a distributed state machine, meaning that it not only stores accounts and balances but also a machine state. There is only a "canonical" state that is shared between all the participants of the Ethereum protocol. The state can change from block to block, and the change is ruled by the Ethereum Virtual Machine (EVM). The Ethereum Virtual Machine (EVM) enables the creation of smart contracts, and programs that are stored and executed on the Ethereum blockchain. Through smart contracts, it is possible to create new digital assets like (fungible) tokens and NFTs (not fungible tokens).

Tokens. Tokens, like coins, are cryptocurrencies that can be exchanged or traded. The main difference is that a coin is the native asset of the blockchain, whereas tokens are created on top of the blockchain, and their mechanisms are defined using smart contracts. In Ethereum, the ERC-20 [136] standard defines the main properties of tokens. ERC-20 was proposed in late 2015 to establish the standard interface for tokens. An ERC-20-compliant smart contract must implement a set of functions and events specified in the standard. These functions are reported in Table 5.2. Some of them are optional, in particular the *name()*, the *symbol()*, and the *decimal()* functions. In Ethereum, tokens and digital assets are held in accounts.

Ethereum accounts. There are two kinds of accounts in Ethereum: Externally owned accounts (EOA) and contract accounts. EOAs consist of a pair of public and private keys generated with the Elliptic Curve Digital Signature Algorithm (ECDSA) [209]. An account is represented by its public address, a 42-character hexadecimal string obtained concatenating "0x" to the last 20 bytes of the Keccak-256 [127] hash of the public key. Generally, users interact with an account using applications called wallets. Example of wallets are MetaMask [261], TrustWallet [379] or MyEtherWallet [267]. A contract account, instead, is an account tied to a smart contract, and it is represented with an address in the same format as an EOA. A contract account is generated when a smart contract is deployed to the Ethereum blockchain. Both accounts can hold and send Ether. However, contract accounts can only send transactions in response to receiving a transaction.

Transactions and fee. A transaction is an action that updates the whole Ethereum network. It can be used to move digital assets, deploy a smart contract, or invoke a smart contract. Executing a transaction has a cost, commonly called a transaction fee. The fee is variable and depends on two main factors: The state of the network (if the network is heavily loaded, the fee is usually higher), and the complexity of the operation that the transaction triggers. For instance, moving Ether from one EAO to another is the cheapest kind of transaction, while interacting with a smart contract could be very expensive. For the sake of simplicity, we can say that the transaction fee is composed of two parts: the *gas limit* and the *gas price*. Gas refers to the unit that measures the computational effort required to execute specific operations. The gas limit represents the maximum amount of gas a user is willing to pay for the operation, and it has to be high enough to pay the computational effort; otherwise, the transaction will fail. Instead, the gas price is the amount of Gwei (10^{-9} Ether) the user is willing to pay for each gas unit.

Smart contract deployment. As mentioned before, smart contracts are programs that run on the Ethereum blockchain. They are written in a high-level programming language (*e.g.,* Solidity [118]) and compiled into bytecode that runs on the Ethereum Virtual Machine (EVM) [134]. A smart contract can be deployed by sending a contract creation transaction from an EOA to the zero address[1]. The transaction contains the bytecode of the smart contract. A smart contract can also create new smart contracts. In this case, the bytecode of the new smart contract has to be embedded in the bytecode of the smart contract that generates the new one. Since a smart contract can not start a transaction by itself but only in response to a transaction that triggers it, an EOA must trigger the generation of a new smart contract.

Events and logs. A smart contract has data associated with it, such as its Ether balance and the value of its variables. Transactions, by calling the smart contract methods, can modify those values, hence the state of the smart contract itself. Knowing the internal state of a smart contract can be crucial, especially in cases where it serves as a backend to distributed applications (dApps). Ethereum provides Events and a Logs register to track the internal states of smart contracts. Each time an action changes the internal state of a smart contract, it can fire an Event that will notify the change. All the events are written on an Event log so that users and developers can easily track the state of the smart contracts in the blockchain.

EVM and EVM compliant. Ethereum is a distributed state machine that changes its state at each new block according to a predefined set of rules. The EVM is the entity that computes these changes in states. Specifications of the EVM are described in the Ethereum Yellowpaper [388]. There are several standard implementations of the EVM in different programming languages (*e.g.,* Python, JavaScript, C++). In addition to Ethereum, other blockchains rely on the EVM (to name a few: BNB Smart Chain [57], Avalanche [324], Fantom [145], Cronos [110]), and they use one of the standard EVM or a complete custom one. These blockchains are called EVM-compliant. They run the same (or with minimal change) smart

[1] 0x0000000000000000000000000000000000000000

contract written in Ethereum, use the same convention for the address, and handle states the same way as Ethereum.

2.2.2 Binance Smart Chain (BSC)

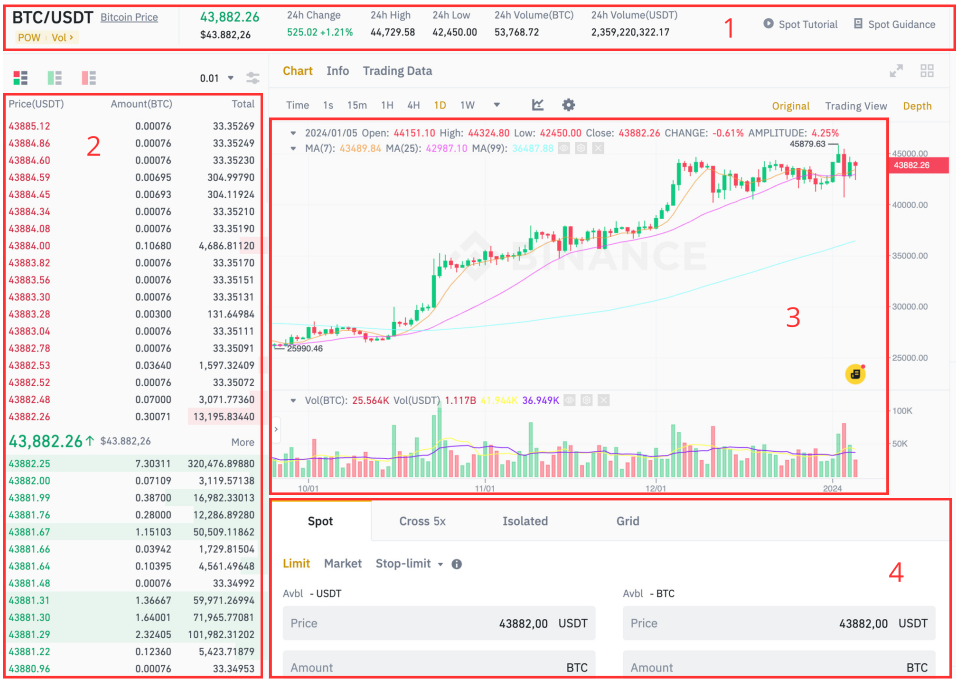
The BNB Smart Chain [57] (previously Binance Smart Chain) or BSC is a blockchain that was born in 2020 as a parallel to the Beacon Chain (previously Binance Chain), and together they form the BNB Chain. The BNB Smart Chain aims to provide a fast and low-cost alternative to other smart contract platforms, especially addressing some of the scalability issues faced by Ethereum. Its consensus is based on the PoSA [59] (Proof of Stake and Authority). While the Beacon chain handles the staking and the governance of the blockchain, the BSC manages the consensus layer and provides EVM compatibility.

The coin of both chains is the BNB (Build and Build, previously Binance Coin)—the third coin by market cap with over 46 billion of capitalization. As Ether on Ethereum, the BNB coin fuels the transactions in the BNB chain. Developers can build decentralized applications and deploy smart contracts on the Binance Smart Chain migrating existing Ethereum-based projects thanks to the EVM compatibility. This allows the creation of tokens on the BSC in a similar way as in Ethereum. The main difference is that BSC tokens follow the BEP-20 standard instead of the ERC-20. The compatibility with Ethereum and the lower transaction fees make the BSC an appealing platform for developers seeking to launch token projects without substantial financial barriers.

2.2.3 Centralized Cryptocurrency Exchanges (CEXes)

A Centralized Cryptocurrency Exchange (CEX) is an online platform for cryptocurrency trading. In general, centralized exchanges work similarly to traditional stock markets, where the users trade cryptocurrencies instead of stocks. Many exchanges focus on facilitating crypto-to-crypto trades, enabling users to exchange one cryptocurrency for another. Instead, platforms such as Coinbase specialize in fiat-to-crypto trading, allowing users to trade traditional currencies (*e.g.,* , USD) for cryptocurrencies. In Centralized Exchanges (CEXes), users can trade cryptocurrencies with the assistance of a centralized entity. Indeed, users typically deposit their digital assets into a custodial wallet managed by the CEX before executing trades. The exchange manages trades on behalf of the users. It provides a user-friendly interface and a wide range of trading pairs.

Fig. 2.3 shows the typical interface of an exchange platform, *i.e.,* the ETH/USDT market in the Binance exchange. We can divide the interface into three core parts, marked in the figure by three numbered rectangles. The figure's top bar (1) shows the pair of cryptocurrencies a user can trade on this market (BTC/USDT). The first currency in the pair is the transaction currency, and the second is the base currency. In this example, the user can buy BTC in exchange for USDT or vice versa. The top bar also shows some statistics about the pair, like the percentage change in price, the 24h high, and 24h low, representing the maximum price and the minimum price reached by the currency in the last 24 hours. Moreover, it also shows the 24-hour volume, representing the total quantity of currency traded in the last 24 hours in



BTC/USDT Bitcoin Price POW Vol ›

43,882.26	24h Change	24h High	24h Low	24h Volume(BTC)	24h Volume(USDT)
$43.882,26	525.02 +1.21%	44,729.58	42,450.00	53,768.72	2,359,220,322.17

Price(USDT)	Amount(BTC)	Total
43885.12	0.00076	33.35269
43884.86	0.00076	33.35249
43884.60	0.00076	33.35230
43884.59	0.00695	304.99790
43884.45	0.00693	304.11924
43884.34	0.00076	33.35210
43884.08	0.00076	33.35190
43884.00	0.10680	4,686.81120
43883.82	0.00076	33.35170
43883.56	0.00076	33.35151
43883.30	0.00076	33.35131
43883.28	0.00300	131.64984
43883.04	0.00076	33.35111
43882.78	0.00076	33.35091
43882.53	0.03640	1,597.32409
43882.52	0.00076	33.35072
43882.48	0.07000	3,071.77360
43882.26	0.30071	13,195.83440
43,882.26↑ $43.882,26		More
43882.25	7.30311	320,476.89880
43882.00	0.07109	3,119.57138
43881.99	0.38700	16,982.33013
43881.76	0.28000	12,286.89280
43881.67	1.15103	50,509.11862
43881.66	0.03942	1,729.81504
43881.64	0.10395	4,561.49648
43881.48	0.00076	33.34992
43881.31	1.36667	59,971.26994
43881.30	1.64001	71,965.77081
43881.29	2.32405	101,982.31202
43881.22	0.12360	5,423.71879
43880.96	0.00076	33.34953

2024/01/05 Open: 44151.10 High: 44324.80 Low: 42450.00 Close: 43882.26 CHANGE: -0.61% AMPLITUDE: 4.25%

MA(7): 43489.84 MA(25): 42987.10 MA(99): 36487.88

Vol(BTC): 25.564K Vol(USDT) 1.117B 41.944K 36.949K

Avbl - USDT

Price 43882,00 USDT

Amount BTC

Avbl - BTC

Price 43882,00 USDT

Amount BTC

Figure 2.3. Key components of the Binance exchange interface, highlighting features such as the statistics section, trading chart, order book, and order placement section.

terms of the BTC and USDT. The volume traded on a cryptocurrency pair is one of the simplest indicators of the health of a market. In general, a high volume of trades is a good indicator that a market is active. In contrast, a low trading volume indicates that people have little interest in cryptocurrency trading and that the market may be exposed to manipulations.

Order book. The left part of the figure (2) shows the order-book. An order book is a ledger that summarizes all the orders of the users on the market. An order to buy is called a bid, while an order to sell is called an ask. The red part of the ledgers contains the ask orders. Each row contains the amount of BTC the users want to sell and the price they are willing to pay. The orders of more users at the same prices are aggregated to make the order book more readable. The green part of the ledger contains the bid orders. Each row contains the amount of BTC users are willing to buy at a specific price. In this case, if more users want to buy BTC at the same price, their orders are aggregated inside the order book. Ask orders and bid orders are ordered by price, the former ascending and the latter descending. There is always a gap between the bid order part and the ask order part of the ledger called the bid-ask spread. When the price of a buy order is equal to the price of a sell order, there is a match, and a trade occurs. Typically, the users who want to buy or sell place an order on the order book and wait for it to be filled to maximize the profit. However, it is also possible to buy a specific quantity of cryptocurrencies regardless of their price. This operation is not economically advantageous, and it is usually used to enter or exit a market quickly.

Candlestick chart. The central part of the interface (3) shows a candlestick chart. The candlestick chart, invented in Japan by Munehisa Homma in the 18th century [247], is a financial chart used to show the history of the movements of assets. A candlestick comprises two main parts: the body and the upper and lower shadows. These two parts are used as a summary of the asset's movement in a time frame, which can usually vary from 1 minute to 6 months or more. The body is the rectangular part of the candlestick and changes meaning according to its color. If the body is green, the lower part of the body shows the opening price of the asset, and the upper part the closing price of the asset. Conversely, if the body is red, the upper part shows the opening price of the asset, and the lower part shows the closing price of the asset. This implies that if the body of a candlestick is green, the asset has closed in gain, while if the body is red, the asset has closed at loss. The two shadows illustrate the higher and the lower price of the asset. Candlestick charts are one of the main visual instruments traders use to decide whether to invest in an asset. Indeed, inexpert users can, at a glance, have simple information on the trend of the coin and expert users can look for some refined patterns. Under the candlestick chart, there is a simple bar chart showing the trading volume of the trading pair in the selected time frame.

Buy/sell interface. The last part of the exchange shows the interface to buy or sell cryptocurrencies (4). Two common kinds of orders can be placed to buy/sell cryptocurrencies: limit orders and market orders. A *limit order* is an order to buy or sell a cryptocurrency at a specified price or better. When placing a limit order to buy, the trader sets a maximum price they are willing to pay. Conversely, when placing a limit order to sell, the trader specifies a minimum acceptable price. Thus, limit orders allow traders to control the price at which they buy or sell a cryptocurrency but may not be immediately filled if the market does not reach the specified price. Conversely, a *market order* allows traders to buy or sell a cryptocurrency immediately. However, the trade is executed at the best available market price, meaning that the user has limited control over the price. Indeed, the trade may potentially be executed at a different price than expected, especially in volatile or low-liquidity markets.

2.2.4 Decentralized Cryptocurrency Exchange (DEXes)

Decentralized exchanges (DEXs) are cryptocurrency exchanges that allow the trade of cryptocurrency without the need for an intermediary. The user interacts with smart contracts deployed on the blockchain, and the user's cryptocurrencies leave their private wallet only when traded. DEXs can be divided into two categories depending on their order matching system *i.e.,* how they match buy and sell orders. The first category of DEXs performs order matching by leveraging a decentralized order book. The order book contains a record of all open buy and sell orders and matches them accordingly. Some examples of this category are dYdX [128], IDEX [196], and EtherDelta [132]. However, the most popular follow the Automated Market Maker model. In this model, trade matching is performed using liquidity pools, and the price of assets is determined using a mathematical formula.

Uniswap [12] is the first decentralized application (DEX) to use the AMM model successfully. According to DefiLlama [245], a popular DeFi statistics aggregator, Uniswap is the 5^{th} dApp by TVL (Total Value Locked, amount of money locked into smart contracts) with over 6 billion USD, while it is the 1^{th} among the AMMs. Uniswap was launched on Ethereum, but now it is also present on the Ethereum

Layer 2 solutions Arbitrum and Optimism and the Ethereum side chain Polygon Matic. Because of its popularity, its open-source smart contracts, and the copyleft license [364], more than 50 protocols were born on several blockchains by forking Uniswap smart contracts in the last years. Uniswap is on its third version, but all its forks belong to the second version since the third one is under a Business Source License [365]. For this reason, in this work, we focus on Uniswap V2 and its forks. One of the most popular forks of Uniswap is PancakeSwap, which lives in BSC, and it is the 1^{st} dApp by TVL on this blockchain with over 4 billion USD locked in its smart contracts.

Fig. 2.4 shows, at a high level, how liquidity pools work in Uniswap. A *liquidity pool* is a smart contract that contains a pair of ERC-20 tokens (A, B) that users can swap. Users that want to invest in the liquidity pool provide both tokens (A, B) to the smart contract, becoming *liquidity providers.* To keep track of the share of the liquidity owned by each investor, liquidity pools use an ERC-20 token called LP-token. When a liquidity provider adds liquidity to the liquidity pool, the smart contract mints LP tokens and transfers them to the liquidity provider. Conversely, a liquidity provider that wants to remove its liquidity can transfer the LP tokens to the liquidity pool smart contracts. The smart contract burns the LP tokens and returns the tokens (A, B) back to the investor. Any user can interact with the liquidity pool to swap token A with token B and vice versa. Suppose a pool consists of x token A and y token B. The price of assets is ruled by the constant product formula, meaning that, at each swap, the pool preserves $x * y$. When a user swaps a token A for token B (the user adds token A to the pool and takes token B from the pool), x increases by a and y decreases by b, where b is computed so that $x * y$ is constant. Thus, token A's value decreases while token B's value increases, and the two parts maintain the same value.

Uniswap implements the AMM model using mainly three smart contracts.

- The *Factory* contract is used to create the smart contract that handles liquidity pools. It is responsible for creating one and only one liquidity pool for each token pair. Each time a new liquidity Pool is created, the Factory contract emits a *PairCreated* event.

- The *Pair* contract implements the AMM logic and keeps track of the pool's status, including the token balances. The Pair contract emits three Events that notify the changes in the status of the liquidity Pool which are the *Mint*, *Burn*, and *Swap.* The Pair contract emits a Mint (or Burn) Event each time an LP-token is minted (or burned) and a Swap event each time a user swaps tokens in a liquidity pool. All liquidity pool created by the Factory smart contract implements these Events.

- The *Router* offers an entry point to interact easily with the other Uniswap smart contracts. Interacting with the Router, it is possible to create liquidity pools, add and remove liquidity, and swap tokens.

Usually, liquidity pools apply a trading fee to each swap operation and distribute a portion of the fees to the liquidity providers according to their LP-tokens.

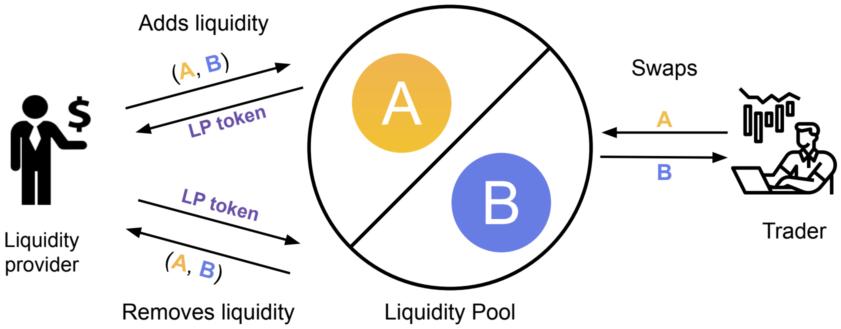


Figure 2.4. Liquidity pool and its main operations.

2.3 Telegram

Telegram is one of the most prominent instant messaging application platforms, with over 700 million active users in 2023 [352]. In subsection 2.3.1, we will give an overview of Telegram and its main features, while, in Section 2.3.2 we will describe the TGDataset, one of the most extensive Telegram datasets.

2.3.1 An overview of Telegram

Telegram provides one-to-one messaging, allowing users to easily engage in conversations by exchanging text messages, multimedia content, and files. Moreover, users can also create and join group chats where any member can post content. This feature allows users to create communities around shared interests for discussions, event planning, and coordination.

Channels. Channels are one of Telegram's core features. They provide one-to-many messaging, as the channel administrator is the only user allowd to send messages into it. Other Telegram users can freely join a channel and read its posts but cannot send messages. This feature allows the admin to share content with a huge number of subscribers, making Telegram channels a prime broadcasting medium for disseminating news and announcements. Channels on Telegram are identified by unique usernames, have a title, and may include a description and a chat picture. Moreover, while group members can see which users are in their group, only a channel admin can access the list of subscribers.

Message forwarding Another core functionality for distributing content within Telegram is message forwarding. Indeed, users (or admins of a channel) can easily forward a message from one chat to another. The forwarded message displays the original message's author, serving as a bridge between groups, channels, and private chats.

The combination of channels and message forwarding makes Telegram a popular choice among numerous public figures, institutions, and businesses to quickly spread information [2]. However, Telegram's feature also exposed the platform to the rapid proliferation of illegal or questionable content, such as revenge porn [333], pedo-pornography [323], animal torture [208], and unregulated gun sales [380].

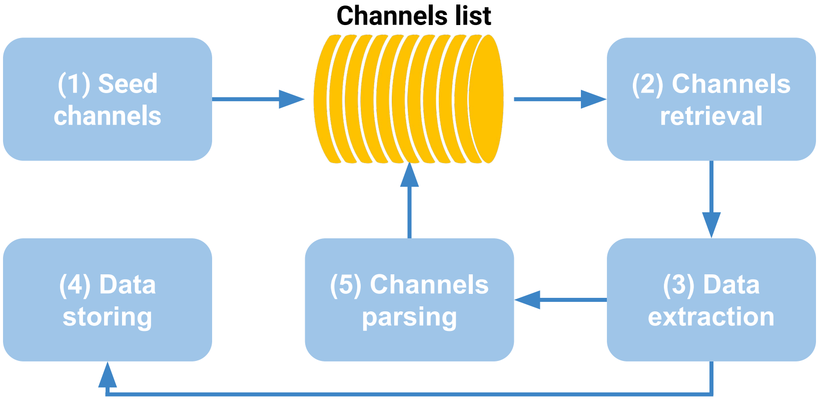


Figure 2.5. The flowchart diagram of the data collection process [229]

2.3.2 TGDataset

The TGDataset [229] is the largest collection of public Telegram channels, with over 120,000 channels and 400 million messages, for a total size of 460GB. The dataset is publicly available at [228]. The TGDataset was created using a snowball approach. The idea is to start with a small set of Telegram channels and gradually expand it leveraging forwarded messages. Figure 2.5 summarizes the steps of the TGDataset creation. The process starts with a small set of seed channels (Step 1). The initial set is gathered through Tgstat [3], a freemium service that collects statistics about over 150,000 Telegram channels. In particular, the authors extract the categories of the top 100 channels by the number of users, finding 18 categories: *Sales*, *Humor and entertainment*, *News and Mass media*, *Video & Movies*, *Business & Startups*, *Cryptocurrencies*, *Politics*, *Technologies*, *Sport*, *Marketing*, *Economics*, *Games*, *Religion*, *Software and Applications*, *Lifehacks*, *Fashion & Beauty*, *Medicine*, *Psychology*, and *Adults*. From each category, they select the ten channels with the highest number of subscribers resulting in 180 seed channels that cover a wide range of topics. In steps 2 and 3, all the relevant information of seed channels (ID, creation date, username, title, description, etc.) and their messages (text, timestamp, author, etc.) are extracted using the Telethon APIs [4], an open-source Python tool that provides access to the official Telegram APIs. Media files are ignored to avoid storing copyrighted or illegal content. Then, the extracted data is stored (step 4) and messages are parsed to identify forwarded messages and their original authors (step 5). If the author of a forwarded message is channels unseen before it is added to a new list of channels to explore. Finally, the TGDataset is expanded by iteratively repeating steps 2,3 and 4 using the newly discovered channels. The collection of the dataset began on 4 January 2021 and ended on 31 July 2022.

Chapter 3

The parallel lives of Autonomous Systems: ASN Allocations vs. BGP

The Internet is a network of independent networks called Autonomous Systems (ASes) that use the Border Gateway Protocol (BGP) [307] to exchange reachability information and effectively interconnect. The number of ASes operating on the Internet has been steadily increasing since its inception, with currently some 70 thousand ASes exchanging routing information in BGP. Autonomous systems are uniquely identified in BGP by their AS number (ASN), which is delegated to ASes by Regional Internet Registries (RIRs).

The link between a given network and the ASN it uses on BGP is key to the proper functioning of the routing infrastructure. However, other than common practices [189, 263, 183] and anecdotal evidence of abuses [174, 172, 355], little is known about the actual relation between the administrative delegation of an AS number and its related announcements in BGP. In this chapter, we develop and apply an analysis methodology to investigate this relation in terms of the actual behaviors observed *in the wild* and extract novel insights.

We perform the first joint longitudinal analysis of ASN delegation records and ASNs' BGP activity. To this end, we restore and build datasets—over a 17-years time frame—that we use as a dual-lens to examine the life cycle of ASNs. We show that this combined perspective can reveal insight into various operational phenomena impacting the security and stability of inter-domain routing—including malicious behavior, misconfiguration, administrative delays, and failed deployments—and potentially inform discussion on best practices and policy.

Our key contributions are:

- We propose a method enabling a novel bi-dimensional lens to look at BGP activity across time, which puts into focus important behaviors by RIRs, operators, and malicious actors.

- We carry out a meticulous restoration of 17 years of delegation files from all five RIRs, learning about errors and inconsistencies present in this precious public source of data. We make available the restored data (on top of which we build our datasets).

- We perform a longitudinal analysis comparing per-RIR behavior and highlighting historical and present trends related to infrastructural growth and (re-)allocation policies.

- Through a taxonomization based on our joint (admin-operational) perspective, we perform an in-depth analysis of the life of Autonomous System numbers. Our analysis reveals a long list of patterns and behaviors that improve our understanding of current practices and anomalies and can inform the discussion around policy and best practices. Although in this work we do not develop a specific detection methodology, our results highlight the potential and practical relevance of ASN delegation data for identifying misconfigurations and malicious behavior.

- We publish our code and datasets for other works to leverage data on the administrative and operational lifetimes of ASNs in the Internet.[1] We will continue updating and publishing our datasets in order to facilitate near-realtime analysis and insight.

Roadmap. The diagram in Figure 3.1 illustrates the pipeline of this work. After providing background on AS number assignments in §3.1, in §3.2 we describe the ASN delegation and BGP datasets we use and our data sanitization and cleaning methods. In particular, we undertake a careful—and to the best of our knowledge, unprecedented—effort to verify and improve the consistency of the data provided in RIR delegation records in order to support our longitudinal analysis. In §3.3 we describe the methodology we use to build administrative and operational lifetimes out of these data. In §3.4 we present a first analysis of what we can learn by jointly looking at the administrative and operational dimensions at a broad (RIR-wide) scale. In §3.5 we delve into an in-depth joint analysis of the parallel lives of ASNs, highlighting insights about usual ASN behaviors, operational practices, inconsistencies, malicious activities, and misconfigurations.

The work presented in this chapter was accepted at the Internet Measurement Conference in 2021. (IMC 2021). In this project, I worked with my supervisor Alessandro Mei, the professor Massimo La Morgia and post-doc Eugenio Nerio Nemmi from Sapienza University of Rome; professor Alberto Dainotti University of California, San Diego (UCSD) and Cecilia Testart from Massachusetts Institute of Technology (MIT).

3.1 Autonomous Systems and the Internet

From the moment the Internet became large enough to have “separate domains” in the early ‘80s, Autonomous Systems (ASes) needed to be identified by a specific number in routing protocols [314, 250]. Even though there is no verification step included in these routing protocols, the management of allocations of AS numbers

[1] Datasets and code available at `https://github.com/SystemsLab-Sapienza/ParallelLives`.

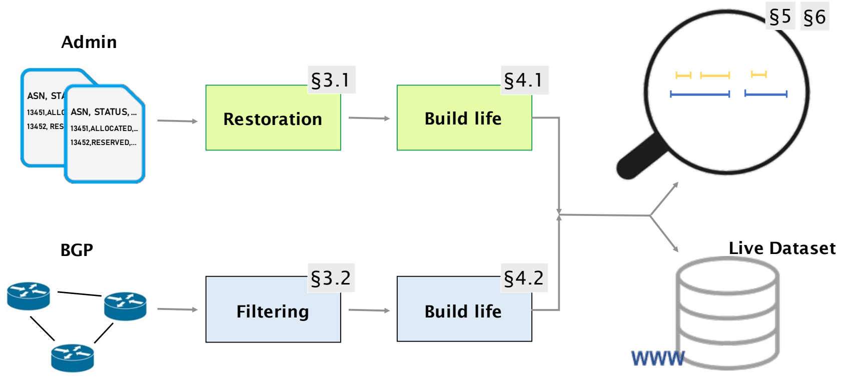


Figure 3.1. Representation of the pipeline of our work and the workflow of the paper.

and other Internet resources is required for the operation of the Internet [77]. From the first delegation in 1983 [78] until now, the management and delegation of ASN has undergone substantial changes.

The early years. In the ‘80s, Jon Postel and the Internet Registry function of the Internet Assigned Numbers Authority (IANA) kept track of the assignations of AS numbers in RFCs [298, 216]. By 1990, 612 AS numbers had already been delegated. In the early ‘90s, following a recommendation by the IETF, the first Regional Internet Registries (RIRs) were created to manage Internet number resources—including AS number delegations—at a regional level, and leaving the IANA as the ultimate central authority, delegating large blocks of resources to the RIRs as needed [213]. Only in the early 2000s, RIPE NCC, ARIN, APNIC, and LACNIC, the registries for Europe, North America, Asia-Pacific, and Latin America and the Caribbean regions respectively, did start periodically publishing and archiving files with the status of Internet resource delegations. AfriNIC, the RIR for Africa, followed shortly after.

The initial daily tracking. While originally each RIR had its own format for keeping track of Internet number resource allocations in files—providing different information and published with different frequency—in 2004, the RIRs [2] unified the format and content [143] of the daily *“delegations files”*. Table 3.1 lists the dates of the first delegation file for each RIR. These files include information about AS numbers delegated, the registry that made the delegation, the country code of the organization to which the resource was allocated, and the date of the allocation.

The current delegation tracking. Between 2008 and 2010, the RIRs started using a new, *“extended”*, Internet resources delegation file format [144] initially developed by APNIC. This new format lists all the resources that are in the pool of each registry, including *(i)* the *available* resources that each RIR has—*i.e.,* resources that have been delegated by the IANA to each RIR to then allocate to organizations in its region—and *(ii) reserved* resources, which are resources in-between states: before either being delegated or returning to the pool of available resources. In addition, the extended format includes an opaque identification value in each line, the `Opaque_id`, to identify an organization within a file, so that resources allocated to the same organization all share the same Opaque_id. This new format provides

[2] At that time the RIRs were APNIC, ARIN, LACNIC, and RIPE NCC. AfriNIC was recognized as an RIR only in April 2005 [20].

a comprehensive picture of all the resources each RIR is responsible for and their respective status. There should be no overlap in resources between delegation files from different registries. All the registries but ARIN produce both the standard and the extended delegation files.

The administrative life of an AS. The administrative life of an AS starts when a registry allocates a specific AS number to the given organization, removing that number resource from the available pool. The ASN will appear in the (extended) delegation file as *allocated*, with a corresponding registration date. The end of the administrative life happens when an ASN is either returned by the holder organization or reclaimed by the respective RIR, in accordance with RIR internal resources allocation policies. The ASN is then quarantined for some time in reserved status before going back to the available pool and being allocated again.

RIR-specific ASN allocation policies and reporting practices. RIRs have different approaches to handle ASN allocations, the eligibility criteria, the recovery of unused resources, the reuse of resources, and special cases (*e.g.,* ASNs reclaimed for a short time or ASN transfers), which impact ASNs' administrative lives. Section 3.6.2 describes in more detail the policies and how they have changed over time. For instance, since 2010, ARIN has been requesting number resources back from organizations that are out of compliance (*e.g.,* did not pay the annual fee), whereas other RIRs only actively reclaim unused resources or just reuse the ones given back to them or when the organization holding an ASN ceases to exist [42]. In addition, tracking in delegation files varies between RIRs for certain cases. For example, if an ASN held by a company is switched from *allocated* to *reserved*, and then it is allocated again to the same company, all RIRs except AfriNIC keep the registration date from the first allocation. Moreover, RIPE NCC and APNIC, do not modify the registration date of an ASN when it is transferred internally (inside the registry). Finally, APNIC allocates ASNs also to NIRs (National Internet Registries), thus introducing more uncertainty to when the NIRs allocate these resources to the end-users.

3.2 Data Collection & Preparation

This section describes our process to collect, restore, and sanitize the delegated files and BGP data we use in this study.

3.2.1 Restoring 17 years of ASN delegations

We collect all (regular/extended) delegation files from the RIRs' FTP sites [19, 36, 38, 234, 312], from the first file available (see Table 3.1 for details), until Mar 1, 2021; the RIRs FTP sites are publicly accessible. Across all RIRs, in less than 1% of the days in our observation time frame it happens that a (regular/extended) delegation file is missing from the site or the available file is corrupted. The longest count of consecutive days missing delegation files is 7 (RIPE). When both regular and extended delegation files are available[3] for the same day, we consider the information

[3] Only ARIN completely stopped publishing the delegated files after Aug. 12, 2013; all the other registries decided to keep publishing both file types.

Table 3.1. Overview of the delegation files we collected from their inception until March 1, 2021 (between 16 and 17 years of data per RIR).

RIR	First regular	First extended	Number of files
AfriNIC	2005-02-18	2012-10-02	5,791
APNIC	2003-10-09	2008-02-14	6,345
ARIN	2003-11-20	2013-03-05	6,303
LACNIC	2004-01-01	2012-06-28	6,257
RIPE NCC	2003-11-26	2010-04-22	6,249

from the extended delegation file. The last column of Table 3.1 lists the total number of files collected per RIR, spanning a period of more than 17 years.

To be able to study the administrative lifetime of ASes through the lens of delegation files, we try to restore missing or potentially corrupted information. We make the restored data publicly available.[4] Our restoration process consists of the following steps.

***(i)* Filling the gap of missing files:** If an AS appears in both the day before and the day after an empty or missing file (157 occurrences), we assume that the AS is also allocated in the missing day. Otherwise, we use as reference for its starting (ending) date, the first (last) day it shows in the delegated files.

***(ii)* Filling missing records:** When comparing consecutive files, we find instances of large ASN count drops, although normally, the count monotonically increases. After careful investigation of large decrements, we find that in most cases when a group of ASes (from few hundreds to few thousands) disappears for one or a few days from the extended delegation file(s), we can recover information by leveraging the data still present in the corresponding regular delegation file(s).

***(iii)* Same day file update:** When comparing extended and regular delegation files from the same day, we find differences in 1.8% of the days—this happens for all RIRs except AfriNIC. We use the newest of the delegation files (based on the start and end times in the headers) to interpret the status of the ASNs accordingly. However, when an ASN disappears from the newest files for a few days but is *always* in the (corresponding) older files, we consider the ASN information in the old ones.

***(iv)* Cleaning invalid duplicate records:** In the AfriNIC files, we find duplicate records with inconsistent information (*e.g.,* allocated and reserved) persisting over periods of up to 6 months, with 16 ASNs affected in total. By manually looking at the history of each ASN, and sometimes their BGP behavior, we gather strong evidence disambiguating the inconsistent information.

***(v)* Restoring registration dates:** Some ASN delegation records show inconsistent registration dates, such as a registration date that is in the future with respect to the file date, that travels back in time across files, or that is filled with a placeholder value. We examine carefully each phenomenon and recover the registration date with the earliest date found in files when possible. For example, we find a few records in AfriNIC files for which the registration date is in the future when compared with the file date. As the difference is of a few days only, we use the date the ASNs first appeared in the delegation files (*i.e.,* the file date) as registration date.

[4] https://github.com/SystemsLab-Sapienza/ParallelLives

We also find ASN delegations for which the registration date travels back in time (when only forward changes are expected, *i.e.,* new allocation). This type of phenomenon affects only few records in all RIRs except RIPE NCC, where more than 800 go backward in time to what we find is a "placeholder" registration date (1993-09-01). Most of these ASes are old ASes delegated in the '90s before the creation of most RIRs. Upon further inspection of these ASNs and contacting the respective RIRs, we trace back and confirm that these ASN allocations are all related to the ERX project: "early registration" ASN transfers from ARIN to the other RIRs [310, 39, 35, 311]. ARIN was formed in December 1997, and it inherited the database of existing address-block and ASN resources from InterNIC. In 2002, the RIRs agreed to have ARIN transfer the management of these resources to the respective RIRs accordingly to the region in which the holder of the resource resided. As a result, 5,026 ASNs were moved to APNIC, LACNIC, and RIPE NCC. We recover and restore the original registration dates leveraging delegation information published by ARIN before the delegation files era [39]. In a second phase of the ERX project, in 2005, once AfriNIC was created, it received 204 ASes in total from ARIN and RIPE NCC. However, in this case, the transfer did not alter the original registration dates.

***(vi)* Cleaning inter-RIR inconsistencies:** We find some 450 ASNs that—at different points in time—are simultaneously being allocated or reserved in multiple RIRs. We identify various overlaps, some affecting many ASes at once and lasting more than 250 days. After careful investigation, we find that the two main reasons for the multiple allocation of the same resources among RIRs are: *(i)* (regular or ERX) transfers where the "origin" RIR temporarily maintains stale data for ASNs that fails to remove from its delegation files and *(ii)* mistaken (apparent) allocations, some by RIRs who have not been assigned those ASN blocks from IANA. In all these cases, we are able to identify the cause and remove the evidently erroneous records from our data.

3.2.2 17 years of BGP data

To find operational ASN activity, we process historical BGP data from all available RIPE RIS [273] and RouteViews [316] collectors, using CAIDA BGPStream's Python library [283], starting on October 9, 2003 and ending on March 1, 2021. To track ASNs that appear in BGP paths, for each day, we process one full RIB dump per collector and *all* update dumps available.

Sanitizing BGP data: We sanitize the data discarding all paths to prefixes either longer than /24 or shorter than /8 for IPv4 and longer than /64 or shorter than /8 for IPv6, since they should not be globally propagated (except for specific cases such as *e.g.,* DDoS protection with BGP blackholing [125]). We also discard paths with loops since they are often related to misconfigurations [191]. A challenge when looking for all ASNs active in BGP is to distinguish low visibility ASNs from ASNs appearing because of errors in the BGP announcements a peer might share with a collector. In our long observation period the probability to incur into spurious data from 1 collector's peer is high. For this reason, we only consider an ASN to be active in BGP in a given day if in that day its visibility is strictly more than 1 peer, *i.e.,* two or more distinct ASes that peer with the collector infrastructure share BGP announcements with that ASN in the path that day.

In total we process **more than 930 billions RIB dump records and 2.3 trillion updates** over 17 years of data. We find a total of 96,391 unique ASNs being routed in BGP in the 17 years of our dataset, from 16,234 on October 9, 2003 to 73,143 on March 1, 2021.

3.3 Building lenses for ASN lifetimes

This section describes our methodology to build ASN lifetimes in terms of administrative allocations (§3.3.1) and BGP operations (§3.3.2). We show a snippet of the datasets in Listing 3.1. We make the datasets resulting from this process also publicly available, together with the code to generate them.[5]

3.3.1 Inferring ASN allocation lifetimes

Our method to infer *administrative ASN lifetimes* is based on two key fields in the delegations files—the allocation status and the registration date—in addition to the policies and practices followed by RIRs. As a general rule, we consider as the start of a new lifetime of an ASN the date of when it first appears or reappears (after deallocation) in the delegated files—or, in the case of extended delegated files, when it is labeled as *allocated*. Typically, this date is close to the registration date. Between 90.1% (AfriNIC) and 99.35% (ARIN) of the cases, the ASN appears in the delegation files the same day or the day after its registration. However, APNIC can allocate AS numbers in blocks to each of its National Internet Registries (NIR), which in turn allocate these resources to end-users. This characteristic introduces more uncertainty over the start of the actual administrative life.

We consider the end of a lifetime when it either becomes *available*, *reserved* or it disappears from delegation files. Specifically, we apply the following rules, which take into account different policies adopted by RIRs, either as documented or based on what we have learned in private conversations:

- ASN appearing allocated after being in *reserved* status or disappeared from the file.
  - An ASN is moved to the *reserved* status (extended delegated files) either if there are administrative issues with the organization that is holding the ASN or for quarantine, before the ASN becomes ready to be reallocated. We use as discriminating factor the registration date: if the ASN returns in the delegated files with the same registration date, it means it was not returned to the free pool, so we can assume it was returned to the previous owner and we merge the two allocation spans in one. Otherwise we infer it was reallocated to someone else.
  - Similarly, in the case in which the ASN disappears from the delegated files (when only regular delegated files are present), we consider the registration date the discriminant between reallocation (new date) and same owner/life (same date).

[5] `https://github.com/SystemsLab-Sapienza/ParallelLives`

  - AfriNIC exception: for AfriNIC, if an ASN has been reserved for any period of time and becomes allocated without first being available, it means they re-allocated the resource to the previous owner even if it gets a new registration date. In this case, we merge the two allocation spans.

- Allocated ASN suddenly changing registration date: An ASN cannot be reallocated before being in quarantine. Thus changes in registration dates without ASNs being deallocated, are explainable by administrative corrections to the same current allocation.

- Inter-RIR transfers (342 in total): if an ASN is transferred across two RIRs, we consider the ASN allocation only one lifetime *iff* there are no gaps between the allocation in each RIR.

By applying these criteria, we identify 126,953 lifetimes, for a total of 106,873 ASNs, that have existed throughout our 17-year time frame of analysis.

3.3.2 Establishing BGP lifetimes

We aggregate BGP data (§3.2.2) at daily granularity, consistently with the resolution available for administrative lifetimes. For each ASN, we consider the start of a *BGP lifetime* the first day we see it in BGP AS paths. However, differently from the administrative dimension, there is no reference concept to leverage to separate periods of BGP activity of an ASN into distinct lifespans. In addition, establishing the end of an ASN lifespan when such ASN is not seen in BGP for only 1 day would be misleading, since it is normal for a BGP speaker to temporarily stop originating prefixes or transiently disappear as a transit in preferred routes (*e.g.,* during an outage). Therefore, in order to introduce the concept of ASN "activity" in BGP for juxtaposition against the administrative dimension, we establish a timeout threshold.

We observe the distribution of per-ASN, activity time gaps with a daily granularity (Figure 3.2, red line) and select an arbitrary *activity timeout* threshold of 30 days, which is approximately where the "knee" of the CDF of activity time gaps starts and corresponds to 70.1% of the distribution. That is, we consider an ASN to start a new operational lifespan only if it reappears in BGP after > 30 days of inactivity. To further understand the implications of picking this threshold, we also look at the number of operational lives that a timeout value would cause to exist within the same administrative lifespan. We consider the "canonical" case for an administrative lifetime to contain at most 1 operational life and we thus compute the distribution of administrative lives that contain one or less operational lives (blue dotted line in Figure 3.2). Our 30 days threshold well fits the area where this CDF starts flattening and corresponds to 83% of the administrative lifetimes having only one or less operational lives. We obtain 152,926 BGP lifetimes for 96,391 ASNs, compared to 126,953 administrative lifetimes for 106,873 ASNs in the delegated files. In Section 3.6.3 we show the (minimal) impact on the rest of our analysis of varying this activity timeout.

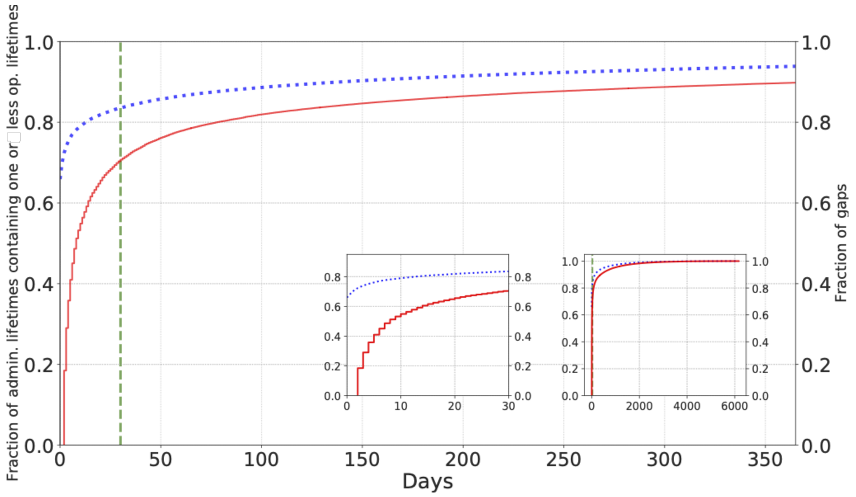


Figure 3.2. Sensitivity to different BGP activity timeout values: Distribution of per-ASN BGP activity gaps (red line) and fraction of administrative lives that contain one or no operational life (blue dotted line) as the timeout threshold changes (x-axis). We choose a BGP inactivity timeout of 30 days (vertical line).

```
# Administrative Dataset
{
    "ASN":205334,
    "regDate":"2017-09-20",
    "startdate":"2017-09-20",
    "enddate":"2021-02-11",
    "status":"allocated",
    "registry":"ripencc"
},
# Operational Dataset
{
    "ASN":205334,
    "startdate":"2017-10-05",
    "enddate":"2017-10-23"
}
```

Listing 3.1. Examples from our Administrative and Operational datasets. The snippets show the records for ASN 205334. The first one represents its administrative life: the AS has been registered and allocated by RIPE NCC in 2017-09-20 and deallocated on 2021-02-11. During that period, it was active in BGP from 2017-10-05 to 2017-10-23.

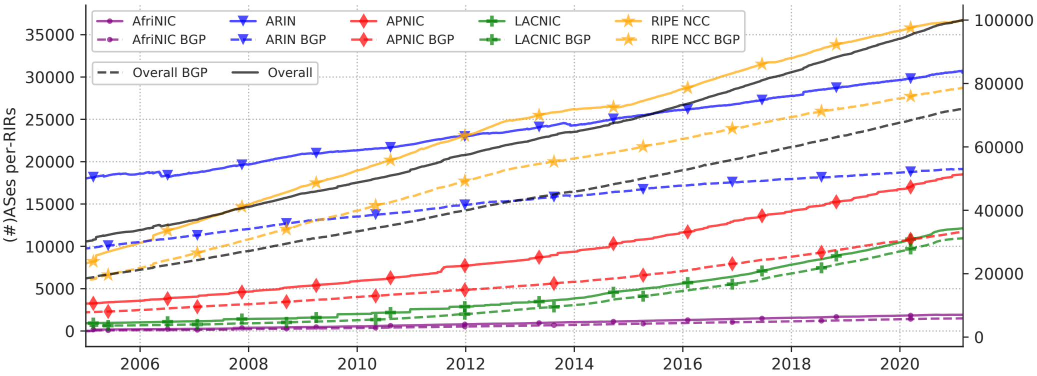


Figure 3.3. Administrative vs BGP lives: number of ASNs per day that are *administratively* (solid lines) and *operationally* (dashed lines) "alive", per RIR (colored) and overall (black). There is a significant and increasing gap between the number of ASNs in the two dimensions, with many ASNs allocated that are not alive in BGP. The growth of individual regions and the change in their proportions over time is visible (*e.g.,* RIPE NCC surpasses ARIN in terms of alive ASNs on BGP in 2009, and only in 2012 for allocations).

3.4 A Bird's Eye View

In this section, we take a look at global and per-RIR trends. We present insights that emerge from a bird's eye view of the data, such as a large number of ASNs never used; in Section 3.6.1 we provide further insight into historical trends. In the next section (§3.5), we instead delve into an in-depth analysis.

A better understanding of regional trends. We find that by using our newly-built administrative and operational lifetime lenses we can better estimate trends (*e.g.,* compared to [162]). Figure 3.3 shows the count of alive ASNs per day, per RIR and overall: administrative and operational data are respectively depicted with solid and dashed lines; for the overall lines, we use the y-axis on the right side. While all RIRs show a growing trend, RIPE NCC exhibits a much faster growth than the other RIRs since the very beginning of our observation period in 2004. At that time RIPE NCC had ten thousand less ASNs than ARIN, but in 2012 it surpassed ARIN, becoming the registry with the largest number of alive ASNs. Note that in public reports at [162] this overtaking is estimated to happen 4 years later, around 2016, since their methodology counts all ASNs ever allocated, including those that were later de-allocated (*i.e.,* returned to the pool of available resources or in transition (reserved) status). Moreover, when comparing the administrative and operational lives, the graph reveals that, in the operational perspective, RIPE NCC surpassed ARIN much earlier: in 2009 compared to 2012. In Section 3.6.1, we show how this data, when broken down by country, provides insight into the expansion of Internet infrastructure in different countries and regions of the world over the years.

Many allocated ASNs are not operationally alive. The graph in Figure 3.3 also highlights that there is a significant gap between the two *overall* (BGP and administrative) lines, *i.e.,* many allocated ASNs that are allocated but are not used in BGP. In March 2021, this gap consisted of more than 27,800 ASNs, meaning that almost 28% of all allocated ASNs are not active in BGP (*i.e.,* . have not appeared

Table 3.2. Number of administrative and operational lifetimes per ASN.

	1 life		2 lives		>2 lives	
RIR	Adm.	Op.	Adm.	Op.	Adm.	Op.
AfriNIC	96.7%	78.6%	3%	12.5%	0.3%	8.9%
APNIC	93.2%	76.9%	6.1%	14.5%	0.7%	8.6%
ARIN	71.9%	65.8%	21.9%	22.4%	6.2%	11.8%
LACNIC	98.4%	88.4%	1.5%	7.9%	0.1%	3.7%
RIPE NCC	84.4%	76.2%	14%	15.0%	1.6%	8.8%
Total	84.1%	74.3%	13.4%	15.8%	2.5%	9.9%

in BGP announcements for at least 30 days). In §3.5.3 we analyze this phenomenon in detail and identify a set of causes.

RIRs still make ASN re-allocations. Most (84.1%) ASNs are never re-allocated. However, RIRs exhibit substantially different behaviors with respect to the reuse of ASNs: Table 3.2 ("Adm." columns) shows, for each RIR, how many ASNs have been allocated once, twice, or more. ARIN and RIPE NCC, re-allocate significantly more than the other RIRs, especially for ASNs that are re-allocated more than once: intuitively, being the two oldest and largest (by total ASNs) RIRs, there is a higher probability their ASNs are re-used. In addition, RIPE NCC and ARIN have more aggressive resource reuse policies [42], which can impact the reuse rate of those RIRs (see Section 3.6.2 for more details). However, as 32-bit ASNs became available in 2007—thus making AS numbers an extremely abundant resource—re-assigning previously used numbers would seem unnecessary and potentially at risk of creating conflicts with stale router configurations or routing policies that operators fail to update—a phenomenon we characterize in §3.5.2. Nevertheless, we observe this practice in all RIRs. A possible explanation is that 16-bit numbers are still a precious resource; we provide more insight about possible issues with 32-bit AS numbers in §3.5.3.

Many ASN allocations are short-lived. A large fraction of ASNs have a long life (CDF in Figure 3.4): more than *5 years* between 65% (ARIN) and 44% (LACNIC) and more than *10 years* between 42% (ARIN) and 19% (LACNIC). However, more interestingly, a significant portion of ASNs do not last more than 1 year. This fraction is higher in the 3 smaller RIRs (LACNIC 13%, APNIC 11%, AfriNIC 9%, versus RIPE NCC 8%, and ARIN 6%). However, when we break down the life duration by the birth year (Figure 3.12 in Section 3.6.1 shows a detailed sequence of boxplots), we find that, starting from around 2010, the life expectancy becomes similar across all RIRs, suggesting that in the last decade it has reached a certain stability in all RIRs. We also find that some short-lived ASNs are likely due to operational issues with 32-bit ASNs experienced by network operators (see §3.5.3 for more details). As RIRs started delegating 32-bits ASNs in 2010-2011, from then on they all have a significant share of ASNs with short administrative lifetimes.

The deployment of 32-bit ASNs is highly diverse across RIRs. Separating the allocations of 16- and 32-bit ASNs we can see how the registries managed the 16-bit ASN exhaustion and the transition to 32-bits. (Figure 3.11 in Section 3.6.1

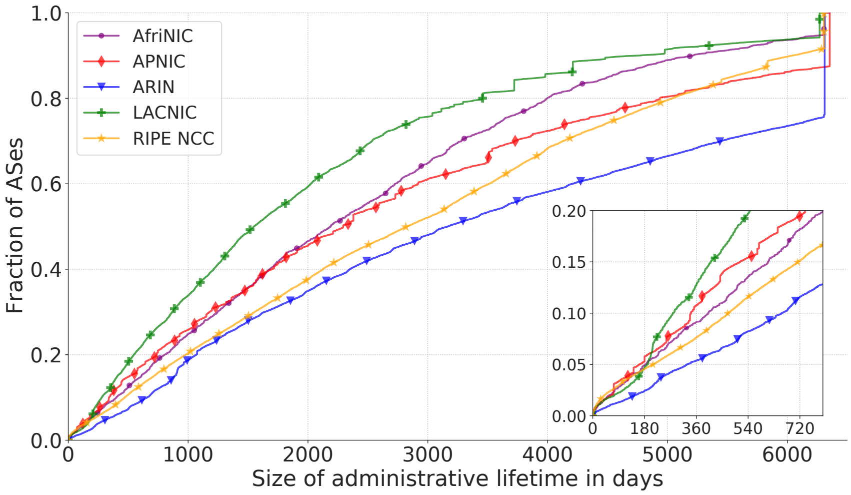


Figure 3.4. CDF of the duration of the administrative lifetimes per RIR. In the bottom right corner: zoom of the CDF focused on the fraction of ASes with shorter life (between 0 and 2 years).

Table 3.3. Distribution of the 4 categories in our taxonomy illustrated in Figure 3.5.

Category	Adm. lives	Op. lives
§6.1 - Complete overlap	99,790	130,397
§6.2 - Partial overlap	4,434	5,434
§6.3 - Unused administrative lives	22,729	0
§6.4 - Op. lives outside delegation	0	2,382
Total	126,953	138,213

shows per-day allocation status of 16- and 32-bit ASNs over time for each RIR). Unexpectedly—despite still being the 2nd largest RIR—ARIN is currently the fourth registry by 32-bit allocations and it only ramps up allocating these resources around 2014, several years after RIPE NCC, APNIC, and LACNIC. Still, in 2020, around 30% of ARIN's new allocations were 16-bit numbers—a completely different behavior compared to the younger registries (APNIC, LACNIC, AfriNIC) where 16-bit ASNs represented only between 1% and 1.7% of all the allocations each of them made in 2020. In Section 3.6.1, we analyze the behaviors related to the 16-bit ASNs exhaustion in more detail.

3.5 Joint analysis of administrative and operational lives

We now align the two lenses we have built in §3.3.1 and §3.3.2 in order to look at individual ASNs when bringing into focus both the administrative and the operational perspectives across time. Jointly looking at them provides an opportunity to better understand operational practices and identify anomalies. We first present a taxonomy of behaviors that it is possible to observe for each ASN when looked through our compound lenses. We then discuss representative examples and novel findings for

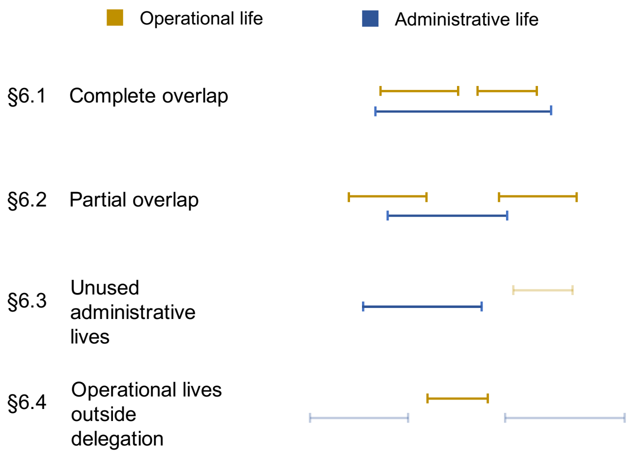


Figure 3.5. Taxonomy of behaviors that it is possible to observe when looking at individual ASNs through our compound lenses. The golden (blue) lines represent administrative (operational) lifetimes. The third and fourth cases show faded out lines representing lifetimes whose presence does not alter the specific case.

each of these categories.

We classify ASNs into four different categories depending on how the administrative and operational lives compare, taking the administrative life as the primary reference. Figure 3.5 provides a graphical representation of the four categories and Table 3.3 shows the count and percentage of ASNs in them. The fourth category of ASNs that have an operational life in BGP without being allocated for the duration of that operational activity (*i.e.,* the operational life is outside any administrative life) may have a disjoint administrative life at another point in time that would fall in one of the 3 categories concerning administrative lives. The four categories in our taxonomy are the following:

1. **Complete overlap:** This is the canonical case, where an operational lifetime happens entirely *within* the time that an ASN is in an allocated state. 78.6% (99,790) of the administrative lives fall in this category. However, we observe large variations *(i)* in the ratio between an operational lifespan and its corresponding administrative lifespan, and *(ii)* in the number of operational lifetimes within the same administrative lifetime. In §3.5.1 we dive into the range of behaviors that we observe in this category and the anomalies linked to malicious behavior that we find.

2. **Partial overlap:** In this case, for a given ASN, we see an operational lifetime overlapping with an administrative lifetime but starting before and/or ending after it. 3.4% (4,434) of the administrative lives present this behavior. In most cases the operational life beginnings and end are close to the related administrative delegation indicating just a slow synchronization of the two dimensions. In §3.5.2 we describe more in detail our findings related to partial overlap.

3. **Unused administrative lives:** These are administrative lifetimes with no BGP activity overlapping with them. Overall almost 18% (22,729) of

administrative lives fall in this category. This behavior is partially explained by the limited visibility of ASNs in the BGP activity captured by the RouteViews and RIPE RIS collecting infrastructure, especially for the China region, the utilization of sibling ASNs, and issues in the deployment of 32-bit ASNs. We analyze and provide more detail on this category in §3.5.3.

4. **Operational lives outside delegation:** We find a total of 1,667 ASNs in this last category. In particular, we discover 799 ASNs that appear in BGP entirely outside of administrative lifetimes and 868 ASNs that are used in BGP for which there is no record of administrative delegation at all by any RIR in the entire 17-years period of examination. We find cases of malicious behavior in the first category, and we identify some reasons for the second one. In §3.5.4 we describe each of these behaviors in detail.

3.5.1 Complete overlap

This is the most common case, accounting for 78.6% (99,790) of all the administrative lifetimes.

Lack of full utilization

Figure 3.6 shows the CDF of the utilization of each administrative life, computed as the ratio between the sum of the operational lifetimes an administrative lifetime contains and its duration. The majority of the administrative lives (70%) are heavily used (more than 75% of their duration) but a close to full usage happens in less than half of the cases (only 45% have a usage greater than 95%). On the contrary, many allocations are heavily under-utilized (*e.g.,* 10% are less than 30% utilized). We analyzed the causes of under-utilization, and found evidence of *(i) late deallocation*, *(ii) sporadic/intermittent use*, and *(iii) largely spaced operational lives.* Below we characterize and provide examples of each of these three behaviors.

Late deallocations. One of the main reasons for the lack of full operational utilization of delegated ASNs is the significant delay in the deallocation of ASNs when they are not operationally active. We find that it often takes months[6] for an ASN to be deallocated since its last day of BGP life: the median for APNIC ASNs is more than 6 months, and more than 10 for all the other RIRs, with AfriNIC's median value being almost a year and a half (530 days). This behavior highlights a potential security problem, which we discuss later, since these resources can be vulnerable to squatting attacks. Delays are also common, though less significant, in the start of operational activity in BGP after an ASN has been allocated: the median is greater than a month for all RIRs.

Sporadic/intermittent use. Another cause of lightly-used administrative lives is the intermittent behavior of BGP activity of some ASNs. The vast majority (84.1%) of the administrative lives that fully overlap with BGP activity actually contain only one operational life. Another significant fraction (10.4%) contains only two operational lives and—despite our 30-days threshold—5.4% has two or more lives.

[6] We perform this analysis only on the administrative lives that end before the last day of our time frame of analysis, March 1, 2021.

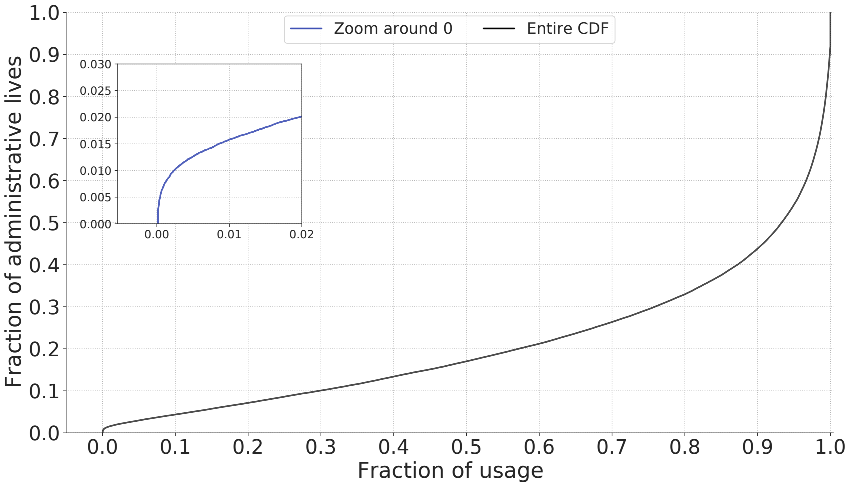


Figure 3.6. CDF of the usage of administrative lifetimes fully containing operational lifetimes, computed as the ratio between the sum the operational lifetimes an administrative lifetime contains and its duration.

Surprisingly, 287 ASNs have more than 10 operational lives. We further investigate these ASNs and find that the majority of them (153 out of 287) have sibling ASNs, *i.e.,* they are part of an organization that manages multiple (sibling) ASes. This suggests that routing policies of large operators (*e.g.,* the same routes might be propagated using their siblings' ASNs, depending on internal routing adopted by the operator) are a possible explanation for sporadic BGP activity. In addition, we manually verify that other ASNs in this category are intermittent "by design": For example, AS37095 (African Network Operators' Group - AFNOG) and AS24555 (Asia Pacific Network Operators Group) are only used by the two network operator groups during their conferences or other events.

Largely spaced operational lives. A third reason causing under-utilization of ASN administrative lives are ASNs having very distant operational lives within the same ASN administrative allocation. Specifically, looking at administrative lives with more than one operational life, we see that 3,789 (23.9%) of them have operational lives more than 365 days apart. While this behavior might be due to organizational or operational changes within a company (*e.g.,* an AS going through changes of providers or in the arrangements with its provider—such as letting a provider announce its space in BGP on its behalf—we find several episodes of malicious activity within this behavior, which we discuss in the next paragraph.

Squatting of dormant ASNs

In *ASN squatting*, an attacker originates BGP announcements of prefixes using an ASN that it does not hold. The squatted ASN is either *(i) dormant*, *i.e.,* allocated but not used to advertise prefixes for long periods, or *(ii)* not allocated at all. This behavior is often associated with malicious purposes, such as announcing squatted prefixes[7] (*e.g.,* for spamming from non-blacklisted address blocks) or hijacking

[7] Prefixes advertised by a malicious actor that were allocated to other organizations that were not advertising them in BGP.

prefixes[8] (which enables various types of attacks). By originating from a different ASN than its own, the attacker tries to disguise their "BGP identity" [174]. For the same attacks, the attacker could also use its own ASN or one it *hijacked* from another organization that was allocated and active on BGP. However, using a dormant/unallocated ASN offers the advantage that potentially there is no owner to notice the event (similarly with property squatting).

We conjecture that, by leveraging the lens of combined administrative-operational lifetimes, squatting of dormant ASNs would result evident in extreme cases. The intuition, is that such attacks should happen after a long time of inactivity and for a short period of time compared to the whole administrative lifespan (*i.e.,* the operational life related to these squatting events will be very short compared to the administrative life of the ASN and far in time from the previous operational lifetime). To test our hypothesis, we set two parameters to detect possible malicious activity of dormant ASNs:

- A *period of inactivity* (while allocated) longer than 1000 days, either since the start of the administrative allocation or between operational lives.
- A *"relative duration"* of the post-dormant operational life (after being inactive in BGP for 1000 or more days and computed as its lifespan divided by the lifespan of the corresponding administrative lives) set to 5%.

Note that these thresholds are arbitrary *by design*, since here we are interested in simply testing our intuition through manual investigation. We find 3,051 operational lives matching our simple filter. We semi-automatically inspect them by counting the daily number of prefixes originated by BGP announcements of those ASNs, and checking their upstream to look for well known malicious actors. We successfully identify many suspicious cases, some of which we are able to cross-validate through external sources, finding at least 76 confirmed cases using information collected from network operators' mailing lists such as NANOG [272], Twitter alerts by network security groups such as Spamhaus [300], routing monitors such as BGPmon [32], and previous work [353]. Unfortunately, broad ground truth about hijacks is not available, thus we cannot quantify in detail how many of these cases are malicious. We confirm as many cases as possible using the sources cited above.

To illustrate this phenomenon, Figure 3.7 shows the number of prefixes originated over time by a subset of these ASNs, providing a visualization of the concept of the awakening of dormant ASNs (*i.e.,* not previously announcing prefixes and not seen in BGP for a long period of time). Furthermore, the figure shows that some hijacks happen simultaneously and we verify those prefix announcements share the same upstream provider (next hop in BGP), suggesting coordination of these attacks. For example, the second spike of AS10512 in the figure, represents a prefix hijacking event disclosed on the NANOG mailing list (the mailing list of North American operators) where one of the victims was Spectrum, a major broadband provider in the U.S. [271]. Even if AS10512 was allocated for more than 17 years (from 2003-11-20 to 2021-03-01), in BGP it was active for only 31 days, from 2017-12-08

[8] Prefixes advertised by a malicious actor that were allocated to and are covered by BGP announcements of other organizations.

to 2017-12-16 and from 2017-12-18 to 2018-01-09. Both periods match the spikes visible in Figure 3.7. In the second one, AS10512 suddenly originated 60 /16 prefixes for a short period, also causing (Sub)MOAS conflicts[9] for some of them, including prefixes originated by Spectrum (AS11426). In other words, AS10512 was squatted and used to perform BGP prefix hijacking attacks. The other ASNs in Figure 3.7 show similar behavior in terms of number of prefixes announced and in some cases also generate (Sub)MOAS events. We find that 2 of these ASNs are in the dataset of potential "serial" BGP hijackers created by Testart et al. in [353].

Some of the ASNs we pinpoint (including AS28071 and AS7449 in Figure 3.7), to the best of our knowledge, have not been previously identified as involved in these type of activities. Interestingly, we find that AS7449, which is unusually active in the same period AS10512 is, shares with it—in the BGP announcements of these events—the same direct upstream, AS203040, an ASN notoriously known as a "BGP Hijack Factory" [270]. It is thus most likely that AS203040 generated and shared with its neighbors forged BGP announcements with these (squatted) ASNs as origins and itself as the first hop, disguising itself as their transit. We identify a similar attack pattern for AS28071 and AS262916 (a well known spammer, reported in 2014 by BGPmon [54]), visible in Fig. 3.7 to be suddenly alive in BGP between 2013 and 2014: through inspection of the AS path in related BGP announcements, we learn they appear to share the same direct upstream—AS52302—during these activity spikes. Searching for this ASN, we find validation of its malicious behavior in the Latin America operators mailing list [235].

However, not all of these malicious events show a sudden increase in the number of prefixes originated per day, making it more challenging to detect them by solely studying their BGP activity without the allocation context. For example, between April and July 2020, 31 ASNs woke up almost simultaneously after several years of inactivity and started announcing each a few /20 prefixes that they never had announced before. We verified these announcements were also malicious, as they involved upstream ASNs known for this type of attacks [348].

Summing up on squatting of dormant ASNs. These case studies show that by using detection parameters that combine the administrative and operational perspectives it is relatively easy to put into focus malicious activity. Our newly-constructed lens could for example provide additional "classification features" for machine-learning based detection approaches. However, our study does not show to which extent and with which accuracy detection would be possible. As previous work on detecting BGP hijacking activity shows [353], it is hard to disambiguate legitimate operations exhibiting irregular/unusual behavior—explainable with traffic engineering, BGP blackholing, *etc.*—from malicious activity. Future work specifically focused on detection would need to rely on ground truth for all the events related to previously dormant ASNs, which is currently not available.

3.5.2 Partial Overlap

This category (second from the top in Figure 3.5), includes all administrative lives that have an operational life starting before and/or ending after it. They represent

[9] Events in which two ASNs originate the same (MOAS) or overlapping (SubMOAS) prefixes.

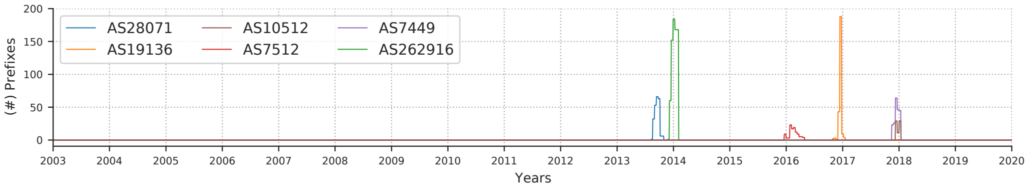


Figure 3.7. Number of prefixes originated by ASNs that suddenly "wake up" on BGP after years of inactivity (while staying allocated for the entire time). Our findings provide evidence of these events being related to malicious ASN squatting perpetrated in the context of BGP prefix hijacking attacks.

only 3.4% (4,434) of all administrative lives that we observe in 17 years of data. We find two benign reasons that explain most of the cases in this category and are described below.

Operators' dangling announcements. Most cases, (2,840, *i.e.,* 64% of all the administrative lives in this category) of partial overlap are due to operational lives continuing beyond the deallocation of their ASNs. The most probable explanation for these cases is the lack of reconfiguration of the routers (*e.g.,* by a provider of the AS). We study the size of ASes exhibiting this behavior using CAIDA ASRank historical snapshots [142] to retrieve their customer cone [251]—the set of ASes that can be reached from them following the customer links in their BGP paths. These ASNs are predominantly small: 95% of them have no customers. Thus, these dangling announcement likely come from manual router configurations that were not updated. Another possible cause of this behavior are stuck routes, where one of the ASNs in the path, does not record a withdraw update, therefore continuing seeing a path that should not exists anymore [108, 141]. While dangling announcements are a phenomenon known by registries, they constitute strong evidence against re-use of ASNs. In our exchange with RIRs, we learned about cases where an RIR had to keep a deallocated ASNs in *reserved* status instead of putting it back in the available pool because of remaining BGP announcements with that ASN. An example is ASN 43268, which was allocated from 2007-07-05 to 2014-12-29 but appears in BGP announcements for almost 2 years after being deallocated (until 2016-09-01), prompting RIPE NCC to keep the ASN out of the available pool during that time.

Late allocations by RIRs. 1,594 ASNs start announcing prefixes in BGP before being allocated by an RIR. However, only 631 of them start announcing before the registration date shown in their respective allocation data. We find these mismatches only last a few days, suggesting their cause is due to a lack of synchronization between when RIRs communicate to the operator the assigned ASN and when they publish the allocation in their delegation files.[10] While this behavior seems of negligible importance, it has significant implications when hypothesizing to use delegation files as reference data for detecting potential misconfiguration and malicious behavior, which we discuss later in §3.9.

[10] RIPE NCC stands out from other RIR with an extremely large median value of 518 days between the start of ASN operation in BGP and the ASN appearance in delegation files. We find this is due to very old ASN resources (*i.e.,* from 1984-03-05 to 2002-09-06), which RIPE NCC added to its delegation files in bulk much later than the date appearing in their "registration date" field.

3.5.3 Allocated but unused administrative lives

No BGP activity is globally observed for a sizable fraction of administrative lifetimes. In total, for 22,729 (17.9%) administrative lives we do not find any BGP activity in our data during their lifespan. This phenomenon happens for 21,431 delegated ASNs, which is 20.7% of the total. Furthermore, 63% (13,407) of ASNs in this category have been allocated but are *never* seen in our BGP data in the entire 17-years period. We note that APNIC allocates entire blocks to National Internet Registries (NIRs), who perform individual allocations that we cannot track (*i.e.,* we consider all ASNs in the allocated block to have an administrative life). However, even if we do not count APNIC allocations, there are still 18,211 lives, allocated by the other 4 RIRs and never globally seen on BGP. This is surprising given that, according to RFC 1930, which provides the baseline guidelines RIRs follow for creating and delegating ASNs (see Section 3.6.2 for more details), *"an AS must be used for exchanging external routing information with other ASes"* [189].

To characterize unused administrative lives, we start by inspecting their duration. Figure 3.8 shows the CDF of the duration of unused administrative lives by RIR. Interestingly, only a short portion of these lives are short-lived: depending on the RIR, only between 14.9% (ARIN) and 45% (LACNIC) of these ASNs had an administrative life lasting less than 1 year. We instead find that the majority of unused lives last multiple years, with a significant fraction being allocated for the entire observation period (the spikes at the end of each distribution)

Our further analysis of unused administrative lives suggests that *(i)* some of those ASNs might be used but are not globally observable in BGP, while others *(ii)* are actually unutilized for various reasons, including the use on the public Internet of sibling ASNs and the failed deployment of 32-bit ASNs. We discuss this analysis in the next paragraphs.

Disproportionate fraction of allocated-but-unobserved ASNs from China. China has a disproportionate fraction of its delegated ASNs that we do not observe in our BGP data. The BGP data collection infrastructure we use has varying levels of visibility depending on the topological and geographical location of ASes that share their BGP announcements with collectors. Nonetheless, we would expect only a small number of (likely transit) ASes impacted by limited visibility, but not such a large-scale phenomenon as the case with Chinese ASNs: Among the top-10 countries by number of unused administrative lives, China is by far the country with the largest fraction of its administrative lives being "allocated-but-unobserved", with 50.6% of all allocated ASNs being unobserved in BGP during the allocation lifetime compared to values below 15% for the runner up countries. Moreover, Chinese allocated-but-unobserved administrative lives represent more than 27% of all the allocated-but-unobserved lives in the APNIC region, even if China has only 10% of APNIC ASN allocations. The other top-10 countries exhibit a much smaller contrast. The next largest is France (14.5% of allocated-but-unobserved), holding—of all administrative lives in the RIPE NNC region—7.9% of allocated-but-unobserved lives but only 4.85% of the allocated (either observed or unobserved). Most other countries have comparable shares of allocated-but-unobserved and all delegations in their respective region. However, Russia stands out for the opposite reason, with a far smaller percentage of allocated-but-unobserved (8.12%) administrative lives

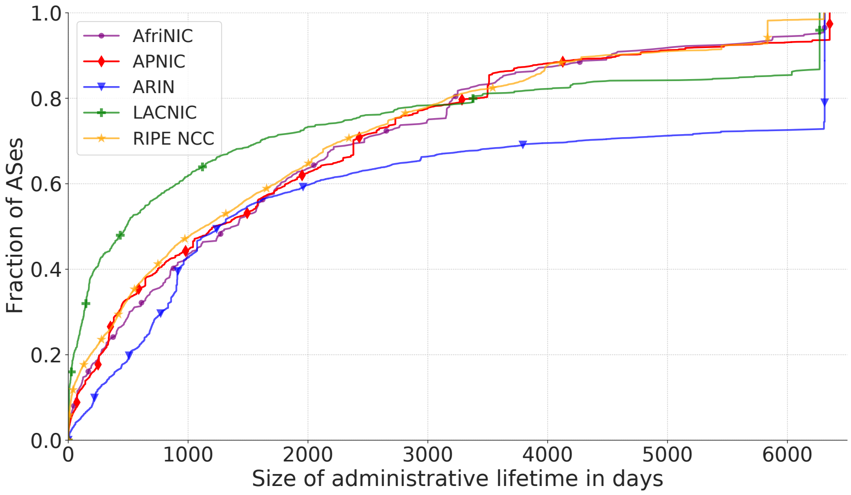


Figure 3.8. Distribution of lifetime for the never used ASNs.

compared to all allocated ones (16%), respectively in the RIPE NCC region. We conjecture that the large fraction of unused ASNs from China is due to how routing is managed in the country: it is possible that several ASNs within the Chinese national AS-level topology are stripped from the AS-paths (*e.g.,* through route aggregation) by their upstream providers before being propagated to the rest of the Internet (where the RouteViews and RIS vantage points are located).

Unused ASN with sibling ASNs in use. Several organizations appear to keep their ASN allocations (and paying the negligible fee) even if they do not use an ASN in BGP—thus either not using it at all or using it only internally. We observe that a large fraction of allocated-but-unobserved ASNs have sibling ASNs, that is, the organization owning them owns also other ASNs. Organizations presenting such behavior include government organizations, such as the US Department of Defense and Air Force—for which we observe only around 40% and 45% of their allocated ASNs respectively—and companies that received large blocks of ASN allocations in the early years, such as Verisign and France Telecom (currently Orange)— which use only 24% and 20% of allocated ASNs respectively.

Challenging deployments of 32-bit ASNs. We examine short-lived unused administrative lives and find that the vast majority of them are 32-bit ASN allocations. Among the unused administrative lives shorter than a month (31 days), 32-bit ASNs represent 92.6% for APNIC, 81% for AfriNIC, 87.3% for RIPE NCC, 65.2% for ARIN, and 38% for LACNIC. By leveraging ARIN's WhoWas service [40], which provides historical information about expired allocations made by ARIN, we investigate if these short-lived allocations are linked to operational issues: We check which organizations were responsible for a random half of the 101 ARIN short lifespans. We then search for the organization names in the list of currently allocated ASNs, and we find that 86% of these organizations have been assigned 16-bit ASNs right after the end of the previous (short-lived) 32-bit ASNs allocation. This finding suggests that short administrative lives that we do not observed in BGP might potentially be caused by operational issues with the deployment of 32-bit ASNs.

3.5.4 Operational lives without allocation

We identify 1,667 ASNs announcing in BGP without an overlapping administrative lifetime. Within this category, we find more evidence of abuse of unused resources (similar to §3.5.1) and ample evidence of misconfigurations. We split them in two sub-categories: 799 ASNs that at a certain point in time were allocated but had at least one BGP life entirely outside of any administrative life and 868 ASNs that have *never* been allocated. Note that we exclude from our analysis "bogon" ASNs normally filtered by operators, *i.e.,* ASNs reserved for special use [195, 160, 375, 263, 11, 224].

More BGP hijacking. Examining ASNs in the first sub-category, which are used in BGP outside their administrative allocation (*i.e.,* after being deallocated), we identify 9 prefix hijacking events where these ASNs were used as origins. We were able to corroborate these events through the same data sources mentioned in §3.5.1. Interestingly, we find that these events are not necessarily far from the closest administrative life but they are always far in time from the last (if ever) seen BGP life. E.g., we see AS12391 originating two /16 blocks and a /18 block (with AS197426 (Bitcanal) as upstream) 3 days after the deallocation of its ASN but 3,898 days after its previous operational life. Note that, differently from the cases we discover and highlight in §3.5.1, these ASNs were not allocated at the moment they were abused. This means that checking the status of these resources on the delegation files could have helped in identifying and preventing these squatting events.

"Fat-finger" misconfigurations that last months. When investigating the 868 ASNs that show BGP activity despite *never* being allocated in our entire 17-years observation period, we identify significant instances of misconfiguration events. Of the ASNs never allocated that appear in BGP, only 427 are active for more than 1 day, 186 more than 1 month, and 15 more than 1 year. We manually investigate more than half of these ASNs and find 258 (29.7%) evident cases of misconfiguration. 76% of these misconfigurations involve an origin ASN similar to an ASN in the AS Path of BGP announcements usually the first hop (*i.e.,* the ASN after the origin): these errors are typically caused by a failed attempt of AS path prepending [90]. For example, in 42 cases we find in the AS path an ASN that is an exact repetition of the origin ASN, such as AS3202632026, where the first hop is AS32026. In the remaining 24% cases, we observe Multiple Origin AS (MOAS) conflicts involving ASNs that differ by 1 digit. Surprisingly these events can last several months. For example, AS419333 appears in BGP for almost 10 months (between Nov, 2017 and Sep, 2018) causing a MOAS with AS41933, IPRAGAZ-AS.

Another example is AS363690 causing a MOAS with AS393690 for almost 7 months (between Nov, 2018 and Jun, 2019).[11]

Unallocated ASNs used internally leak to the global Internet. Among the "never allocated" ASNs, we also observed (unallocated) ASNs with very large numbers. We found that 472 (54.4% of the 868 never allocated) have more digits than the highest allocated ASN, which is 6 digits long. The majority of the events we could manually investigate appear to be the unintended consequence of benign

[11] Note that an attacker might be able to carefully choose an ASN to squat that looks like a mistyped ASN of the victim. In the cases we investigated, we verified that the upstream ASNs in the AS paths match the upstreams of the corresponding legitimate ASN (*i.e.,* strongly suggesting that these are actual fat-finger mistakes).

behavior and often last months, if not years. For example, AS290012147 announced a /24 prefix for more than 2 years (between 2015 and 2017), which is covered by a /12 announced by AS701, held by Verizon. We collect all AS paths from BGP announcements including that ASN for a day (while it was announced) and find that they all have the ASN triplet {AS290012147, AS7046, AS701}. Since both AS701 and AS7046 are held by Verizon, and AS701 announces the covering /12 prefix, it is very likely that such announcements are due to a misconfiguration "leaking" routes used internally by Verizon. Similarly, we find events associated with other large unallocated ASNs (such as AS499981773, AS3489671207, and AS12845938). Note that these are not "bogon" ASNs defined in RFCs for internal use but are actual valid ASNs that RIRs might allocate.

3.6 Further Insights

3.6.1 Administrative Lifetime Analysis

In this Section, we extend the analysis of §3.4 based on the administrative and operational ASN lives we build (see §3.3), providing insights into the expansion of Internet infrastructure in different countries and regions of the world over the years.

Registries growth. When studying ASNs' administrative lives, through the ASN registration date field, we can observe allocations dating back to 1992. In Figure 3.9, we compare the (quarterly) birth rate of administrative lives across RIRs over time. The graph clearly shows a spike in allocations around year 2000, explainable with the so-called "Internet bubble" [384], and highlights the explosion of LACNIC and APNIC starting from 2014. Looking at the (quarterly) balance between births and deaths over time (Figure 3.10) helps us to further capture the infrastructural Internet expansion of these two regions: In the last three years, APNIC and LACNIC have gained more than 1000 ASN net allocations more than ARIN ($\approx$ 4,000 for APNIC and LACNIC and $\approx$ 3,000 for ARIN). RIPE NCC, still slightly leads, with more than 4,400 ASNs than it had at the beginning of 2018.

Countries infrastructural expansion. The analysis of the ASNs allocations by country, reveals which countries have had faster growth in ASNs allocation in recent years. Brazil is by far the leading country in its region, with an increment in allocations of the total LACNIC ASNs from 64% in 2015 to more than 70% in March 2021 (Argentina is the second country, with only 9.5% of LACNIC ASN allocations). Interestingly, within APNIC, India has climbed to the top (In March 2021, India had more than 15% of all APNIC ASN allocations, while in 2010 it was not even in the top-5!) surpassing Australia, which had been leading in the region since 2006 (Table 3.4). The third most represented country in the APNIC region is now Indonesia, which recently surpassed China (11.1% and 10.6%, respectively). The ARIN region is dominated by the U.S., with more than 92% of all the allocated resources. In AfriNIC, South Africa is the leading country (with more than 32% of ASN allocations). Finally, in the RIPE region, resources have been distributed more evenly across several countries. Russia largely leads the region with 16.6% of allocated ASNs, more than twice the number of allocated ASNs of the UK, the second largest country.

16-bit exhaustion. Using our data, we also analyze how close to exhaustion

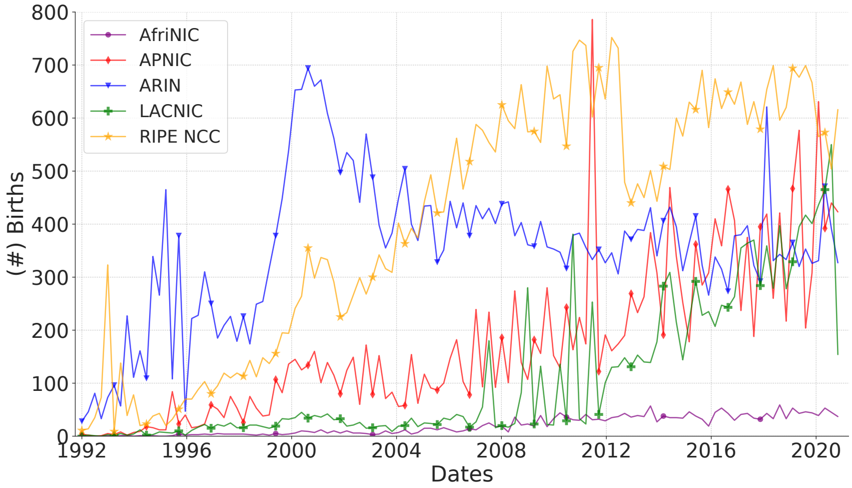


Figure 3.9. Per-RIR ASN administrative birth rate (3-month bins). It shows the 2000's Internet Bubble and the change in pace of RIPE (2003) and APNIC and LACNIC around 2014

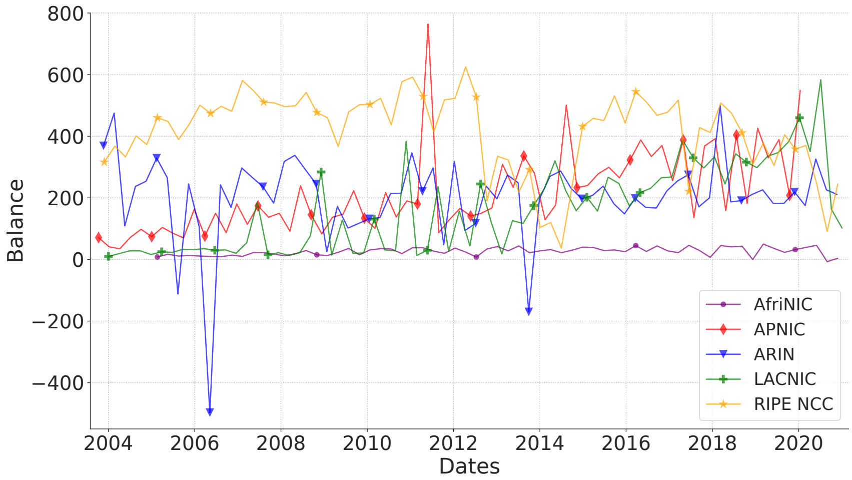


Figure 3.10. Balance between new ASN allocations and deaths. The volume of RIPE's ASN allocations from 2005 to 2013 is massive. Around 2017, APNIC and LACNIC's ASN allocations exceed ARIN's.

Table 3.4. APNIC countries evolution.

Pos.	2010	2015	2021
1°	AU: 1038 - 17.6%	AU: 1697 - 16.1%	IN: 2917 - 15.7%
2°	KR: 863 - 14.6%	CN: 1202 - 11.4%	AU: 2681 - 14.5%
3°	JP: 762 - 12.9%	JP: 1103 - 10.4%	ID: 2059 - 11.1%
4°	CN: 449 - 7.6%	IN: 1070 - 10.1%	CN: 1967 - 10.6%
5°	ID: 417 - 7.1%	KR: 1019 - 9.6%	JP: 1127 - 6.1%

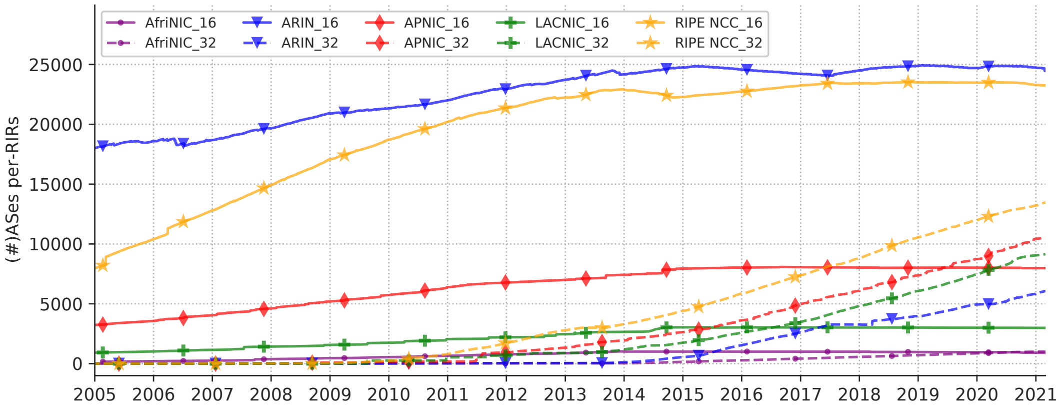


Figure 3.11. Administrative lives: count of 16-bit (solid lines) and 32-bit (dashed lines) ASNs allocated per day. We can clearly see that the growth of 32-bit allocations is different between the registries. In particular, ARIN 32-bit allocations (dashed blue line) ramp up late (mid-2014) when compared to RIPE NCC, APNIC and LACNIC, despite ARIN being the second-largest registry.

of 16-bit ASNs were the different registries. Looking at the availability of 16-bit numbers, we discover that none of the registries actually used every 16-bit they could allocate. Studying the daily number of 16-bit ASN allocations, the registries reach their maximum in different periods: end of 2013 for AfriNIC, mid-2016 for APNIC, beginning of 2019 for ARIN, mid-2015 for LACNIC, and end of 2018 for RIPE NCC. The global largest number of 16-bit allocations was reached on January 23, 2019, with 60,455 ASNs and globally only 4,039 16-bit available, removing the ones private or reserved by RFC [160, 263, 183].

3.6.2 RIR policies

The Regional Internet Registries (RIRs) were created in the '90s to manage the delegation of Internet number resources, *i.e.,* Internet Protocol (IP) addresses (IPv4 and IPv6) and AS numbers, at a regional level. Regarding the delegation of Autonomous System Numbers (ASNs), RFC 1930 (also Best Current Practice (BCP) 6) [189] has provided guidelines for the creation and registration of ASNs since it was published in 1996. RFC 1930 has indeed been the baseline of RIR policies for delegating ASNs ever since. The Number Resource Organization (NRO), created in 2003 to coordinate the work of RIRs, has tracked and compared RIR policies—including the ones for allocating AS numbers—since 2004. It publishes the RIR Comparative Policy Overview [280] a few times per year, providing a valuable source about RIR policies and their changes. The next paragraphs describe RIR policies and practices related to the allocation of ASNs and how they have changed over time. When possible, we link the allocation process to the delegation files and describe practices related to the tracking of ASN allocations that we infer from our datasets (see §3.2 for dataset descriptions).

Eligibility Requirements. RIRs have policies that describe which organizations are eligible to be allocated an ASN. In 2004 (the first year with historical policy

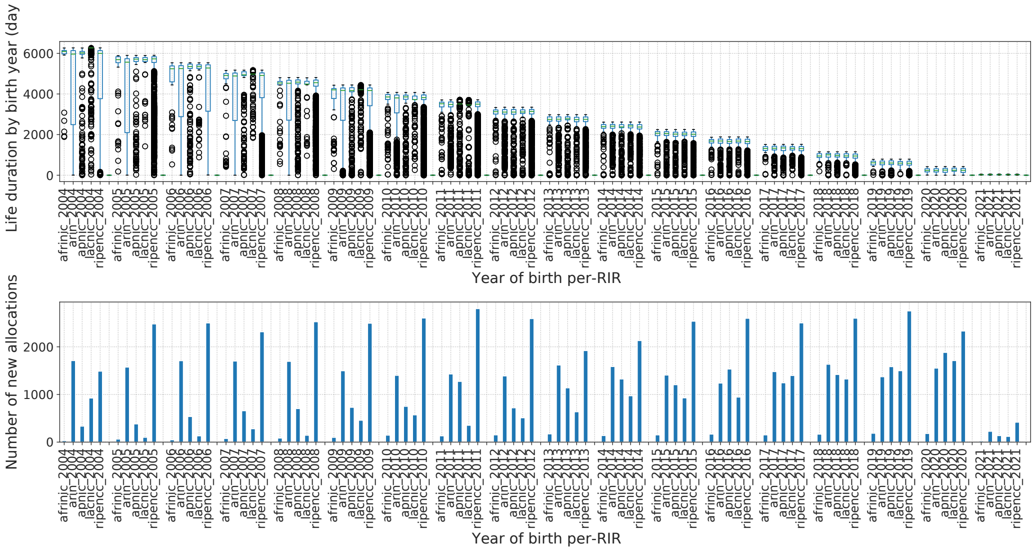


Figure 3.12. Life expectancy based on year of allocation: The upper sequence of boxplots represents the administrative life duration per registry based on the year of birth (allocation). The bottom image represents the number of new allocations per-RIR for each year. Starting around 2010, the life expectancy becomes similar for all the RIRs, suggesting a kind of life stability.

documents available), ARIN, LACNIC, RIPE NCC and APNIC[12] explicitly cited RFC 1930 in their eligibility criteria, stressing two main conditions:

1. The organization has a unique routing policy, distinct from its provider (*i.e.,* the provider could not advertise the organization prefixes itself), or

2. The organization is multihomed.

APNIC used a stricter criteria in 2004, requiring both conditions described above. In addition, APNIC is the only RIR delegating blocks of ASNs to National Internet Registries (NIRs) for further distribution between their members.

Over the years, RIRs have slightly updated the eligibility criteria, mainly to allow organizations to comply with the requirements within 6 months and replacing multihomed with any setting needing to interconnect with an ASN. Starting in 2015, LACNIC also requests applicants a detailed routing policy, including the list of prefixes they will advertise.

ASN deallocation and reuse. Policies and practices relating to the deallocation of ASNs are only succinctly touched upon in RIRs' policy documents when describing reuse policies. In general, as long as the delegation criteria remains valid, all RIRs will keep a delegation active. However, initially only APNIC had a policy to actively recover unused resources (for the ones it delegated directly, not through a NIR), although all RIRs would put an ASN back in the available pool should the organization the ASN was delegated to cease operations. In 2010, LACNIC and RIPE NCC adopted the policy to actively recover unused resources and ARIN

[12] AfriNIC was being created at the time and defined its policies a few months later.

Table 3.5. The table shows how the choice of the inactivity timeout impacts the distribution of cases in our taxonomy.

Timeout	Complete overlap	Partial overlap	Op. lives outside delegation
15	99,834 (+ 0.04%)	4,390 (- 0.99%)	1,750 (+ 4.9%)
30	99,790	4,434	1,667
50	99,713 (- 0.08%)	4,511 (+ 1.74%)	1,592 (- 4.4%)

included a policy requesting organizations found to be “materially out of compliance" (*e.g.,* owing the annual fee) to return their resources [42]. Nonetheless, through our exchange with RIRs, we learned that the enforcement of these policies has varied over time. In particular, when 16-bit ASNs became scarce in the mid 2010s, RIPE NCC made the reuse of ASN easier and faster (*e.g.,* not waiting until all dangling announcements of de-allocated ASNs disappear from BGP before putting the ASN back in the available pool). Analyzing the reallocation of AS numbers (reported in Table 3.2), we identify that indeed ARIN and RIPE have reallocated more resources than the other registries. These practices also impact the deallocation of ASNs and thus the end of the administrative lives we compute in our analysis. We find that it often takes months for AS numbers to be deallocated after their last activity on BGP: the median for APNIC ASNs is more than 6 months, and more than 10 for all the other RIRs.

32-bit ASNs. RIRs started allocating 32-bit ASNs in 2007. At that time, RIRs would delegate 32-bit numbers only if applicants requested 32-bit ASNs. Then, in 2009, RIRs started to delegate 32-bit numbers unless the applicants specifically requested 16-bit ASNs. After this point, RIRs took different paths in the allocation of 16- and 32-bit ASNs. Starting mid 2009, APNIC only allocated 16-bit ASNs if the applicant could “demonstrate that a 32-bit only AS Number is unsuitable". Similarly, in 2010, LACNIC started requesting applicants for 16-bit ASNs to “duly justify the technical reasons” for not using a 32-bit ASN. However, also in 2010, RIPE NCC, ARIN and AfriNIC simply ceased to make any distinction between 16-bit and 32-bit AS Numbers and started assigning them from an undifferentiated 32-bit AS Number allocation pool. In the delegation files dataset, we confirm that RIRs started allocating 32-bit ASN in 2007.[13] Figure 3.11 shows the number of allocated 16- and 32-bit ASNs per RIR per day. We also notice that the share of allocations 32-bit ASNs represent per RIR evolves differently over time, with the share for APNIC and LACNIC growing much faster than for AfriNIC, RIPE NCC and specially ARIN.

Tracking allocations in delegation files. RIRs use the delegations files to track and make publicly available the allocation records of number resources, including ASNs (for details about the content of these files, see §3.1). Using these files and the methodology described in §3.3.1, we infer the administrative lifetimes of the ASNs. However, while analyzing the delegation files, we realize that RIRs have different practices when it comes to updating and handling the delegation files. For instance, after allocating an ASN, the precise timing of when the record is added to

[13] The one exception is RIPE NCC, which delegated a first 32-bit ASN in December 2006.

the file varies. We found that between 90.1% (AfriNIC) and 99.35% (ARIN) of ASN allocations, the ASN appears in the delegation files the same day or the day after its registration. In addition, the outliers that we encountered in our analysis prompted us to exchange emails with RIRs. In §3.2.1, we describe these phenomena, including the drop of allocated AS numbers, invalid duplicate records, and registration dates that travel back in time. From our exchanges with RIRs, we also learned about challenges they faced in keeping up-to-date the files and dealing with corner cases of resource allocations, both of which sometimes lead to resources disappearing from the files a few days while the issues are being sorted.

3.6.3 Inactivity threshold

In §3.3.2 we set a timeout threshold to introduce the concept of operational life of an ASN in BGP. After careful consideration and based on the sensitivity analysis of the distribution of key variables (per-ASN BGP activity gaps and fraction of administrative lives that contain only one or no operational life, shown in Figure 3.2), we choose a 30 days threshold. Here, to further explore the implications of our choice, we extend our sensitivity analysis to determine how the four categories from our proposed taxonomy (§3.5) change using either a smaller (15 days) or a larger (50 days) threshold.

In Table 3.5 we report the impact of 3 different thresholds on the distribution of ASN lives in the 3 categories of our taxonomy that consider operational lives. The highlighted row—the middle row—shows the distribution with a 30-day timeout, the threshold we use in the chapter (baseline). The two other rows show numbers for the same categories with the associated threshold (15 and 50), highlighting the delta (in percent) with respect to our 30-day choice. We do not report in the table the never-used category (§3.5.3) since it is not impacted at all by the choice of the threshold as those ASNs are never seen in BGP.

Table 3.5 shows that changing the value of the threshold does not have a significant impact on the number of ASNs that completely overlap (§3.5.1) and partially overlap (§3.5.2). The most affected category is the "Operational lives without allocation" (§3.5.4). However, it is a small fluctuation of less than 5%, that is almost symmetric around the threshold we picked. These changes are not significant and do not alter the substance of our findings.

3.7 Related Work

The allocation of Internet resources has been studied for a long time, however the focus has been on IP block allocations. Huston [163, 157, 159] has produced information on the total number of allocations of IPs along with per RIR allocation analysis: How many resources are allocated in the delegated files and how many of them are routed. With this analysis, Huston shows the increased rate of IPs allocation and gives insights on IPv4 address exhaustion. In [308], Richter et al. study IPv4 addresses exhaustion and how the evolution and management ecosystem created diverse realities in different regions. In [309], Richter et al. analyze the operational use of IPv4 addresses from the point of view of a large CDN and characterize behaviors revealing under-utilization in some regions and complete utilization in others. Starting from the delegated files, Meng et al. analyze the correlation between the allocation of IP blocks and their usage in BGP, discovering that most of the prefixes allocated between 1997 and 2004 appear as routed after 75 days and that 8% have not been used at all [260]. Sriraman et al. [336] analyze the fragmentation of the IP address space contrasting allocated blocks with block routed on BGP for a period of five years, finding that almost 90% of ASes with a provider-customer relationship do not share an address delegation relationship. Similarly, Heidemann et al. [190] use allocation data of IPs to assess that only 3.6% of these addresses are actually visible hosts. More recently, Dainotti et al. [115] proposed a taxonomy and a new method combining active and passive measurements to understand address utilization. They discovered that only 37% of the total number of IPv4 usable addresses are actually used, and that most of the unused blocks are in the US. Other work focuses on the effectiveness of bogon lists and on how to improve their use [139, 124, 41]. In particular, the most common problem is that these lists are usually not updated as soon as new allocations are made, and therefore valid routes can be filtered out. Vaidyanathan et al. [366] introduced in the bogon lists the semi-dark space, addresses that are not in operational use. All these works focus on IP allocation rather than ASes.

Concerning ASes, many works have studied specific aspects of AS behavior in BGP without considering ASN delegations and their administrative lives. Chang et al. [82] built AS-TRUST, a scheme to quantify the reputation of an AS based on BGP updates, showing that it is possible to improve BGP operations. Konte et al. build ASwatch, a system to find bulletproof hosting ASes based on network and connectivity features of ASes inferred from BGP data [218]. Since these works do not take into account ASN delegations, they do not evaluate AS behavior depending on allocation status, which would allow to discern ASNs that were previously delegated to another organization. In [353], Testart et al. build a supervised machine learning system to find ASes that persistently hijack BGP prefixes. In our work, we provide evidence that using both the administrative and the operational dimensions, it is possible to separate behaviors from different allocations (*i.e.,* different administrative lives of the same ASN), thus possibly better characterizing the overall AS behavior. We believe this approach can improve detection methods solely based on the operational activity. Huston [161, 158] has published analyses on ASN consumption and aggregated allocation. Other works on ASes analyze their connectivity structures [331, 315, 281]. In summary, most of the works on ASes are based on BGP data and their interconnections rather than the life of these resource allocations in the Internet and their effective use in BGP.

In 2005, Wilhelm and Uijterwaal correlated ASN delegations and their activity in BGP [386]. However, in 2005 AfriNIC was just born and we are now able to analyze 17 years of data. Policies changed and extended delegated files carrying more information have been introduced, allowing us to better characterize what invalid resources are being advertised. Moreover, we introduce new concepts such as ASN delegated life, ASN BGP life and ASN usage and perform a longitudinal analysis on the correlation between administrative and BGP lives.

3.8 Limitations

ASN-level granularity. In our study, we work with ASN-level data. We do not look at the individual prefixes advertised by ASNs, except in few manual analyses to better understand and characterize our findings (as in §3.5.1 on ASN squatting). However, information about the announced prefixes may help to further build and characterize BGP lifetimes, *e.g.,* identifying different BGP lifetimes of the same ASN based on different sets of announced prefixes. E.g., in §3.3.2 we pick an arbitrary 30-days inactivity threshold to separate two operational lives. Using prefixes, we could consider both the inactivity period and the prefixes announced by the ASN to decide whether to start a new operational lifespan or not.

Visibility limitations. We can only infer the use of an ASN in BGP if the BGP announcements from that ASN reach a peer of the collecting infrastructure we use. The existing collecting infrastructures have several vantage points, but they are not uniformly distributed around the globe. Indeed there are jurisdictions such as China, that heavily control the local interconnection with the global Internet and where such measurement infrastructure is not present. This is a factor that can limit the inference of operational activities of ASNs in some specific geographical areas.

Collectors. There are other BGP data collection infrastructures available, such as *e.g.,* from the Packet Clearing House project (PCH) [193]. However, adding further collectors is unlikely to significantly alter our findings, since—differently from BGP prefixes, which might not propagate far in the topology, or might be shared in private peerings, or might end up aggregated—the operational information we are interested in (AS numbers from BGP announcements) does propagate in the topology. An exception would be if *e.g.,* PCH or another BGP collecting infrastructure had a presence in China, where (see previous paragraph) we find a limitation due to likely a filtering of AS numbers; in that case, we might be able to observe Chinese ASNs that are never propagated to the rest of the Internet. We are not aware of public BGP data collection infrastructure with such coverage.

Private peering. Another issue we might encounter is ASNs not visible in BGP because used for private peering. However, in the majority of such cases, we would expect the owning organizations to also use a second ASN publicly. If this was a significant phenomenon, we would find many unobserved ASNs to have siblings. In §3.5.3 we show that sibling ASNs are not significant in number and are not enough to explain the extremely large number of unseen ASNs we find.

3.9 Discussion

In this chaper we align two dimensions along which ASNs are visible across time: their administrative allocation by registries and their operational use in BGP. ASNs are a key Internet infrastructural resource and this link is crucial for the operation and security of inter-domain routing but has received little attention in the research community. The combination of the administrative and operational lenses that we build through our datasets allows us to characterize the different behaviors that stem from the *interaction* between ASN delegation and BGP, the policies set by Internet Registries, misconfigurations, and malicious behavior.

Contrasting the administrative and operational dimensions of an ASN, we find that even though most organizations receive an ASN allocation and then start operating in BGP, there is a large breadth of different behaviors. At the two extremes, we find ASNs that are delegated (for many years) that never appear in BGP, and ASNs that operate in BGP without being allocated at that time. In between we have BGP operation fully or partially covering the ASN allocation. These behaviors are shaped by 3 distinct aspects:

- **RIRs policies and management of ASN delegations:** Whether RIRs delegate in block or mainly single ASNs, the internal delegation process (and when ASN are included in delegation files), the reuse policies and re-allocation process of previously allocated ASNs, and the choice of delegating 16-bits vs. 32-bit ASNs, they all impact the usage of allocated ASNs in BGP. Therefore, further study of our dataset can help elucidate best practices for both the delegation and use of ASN resources and the broader impact of these policies in the Internet infrastructure and ecosystem.
- **Misconfigurations and mistakes in operational setting and in RIRs delegation process:** Many operational and administrative errors quickly show up as anomalous behavior when combining these lenses. Indeed we find that fat-finger errors are the largest contributor of ASNs seen in BGP that have never ever been allocated to an organization. When these fat-finger errors and other misconfigurations relate to the origin AS, access to authoritative records of the correct ASN as origin of a given prefix would allow to verify the information in BGP and limit the spread of invalid announcements. Thus, if ASes have properly issued Route Origin Authorizations (ROAs) in the Resource Public Key Infrastructure (RPKI) for their prefixes, the spread of errors and misconfigurations would be limited when networks in the path drop RPKI-invalid announcements, *i.e.,* implement RPKI filtering.
- **Malicious behavior:** By studying the usage of ASNs in BGP during and after administrative allocation we are able to spot many indications of malicious behavior. There is much further work to do to characterize all the malicious behavior that is detected with these combined lenses. However, as a high-level conclusion from our manual analysis, hijackers are ahead of us: they carefully pick dormant or previously allocated ASNs to make their attacks stealthier (*i.e.,* mostly avoiding picking never-allocated ASNs, which we instead see in misconfigurations). Similarly to the case of misconfigurations though, when unallocated ASes are used as origin, if the victims of attacks had properly registered ROAs providing an authoritative record of the ASN authorized to announce as origin a given prefix, networks dropping RPKI-invalid would limit the spread of this type of attacks.

Practical relevance: We argue that this dual-lens has operational value to reduce the spread of misconfigurations in BGP (*e.g.,* by filtering all ASNs that are not delegated) and make malicious behavior, as well as operational problems (*e.g.,* the challenge with 32-bit ASNs), more visible. However, our study also highlights inconsistencies and behaviors—*e.g.,* mistakes and delays in the delegation files, dangling announcements after deallocation, large AS numbers “illegitimately“ used internally and sometimes leaking—that should be addressed through policy and best practices in order to make delegation information more useful for operational purposes.

As a future work, we expect to extend our dataset to integrate other information shaping ASNs behavior: (1) information about sibling organizations in order to prune our correlation between administrative lifetime and BGP lifetime; (2) data from IP address delegations with the purpose of better characterizing the administrative dimension of a network; and (3) distinguishing between origination and transit BGP activity of an ASN to differentiate the role(s) an ASN has at different times of its BGP lifetime.

Chapter 4

The Doge of Wall Street: Analysis and Detection of Pump and Dump Cryptocurrency Manipulations

Pump and dump is a market manipulation fraud that consists in artificially inflating the price of an owned security and then selling it at a much higher price to other investors [226, 222]. This fraud is as old as the stock market. One of the most famous pump and dumps in Wall Street history happened in the late '20. The security was the RCA Corporation. RCA was the manufacturer of the first all-electric phonograph, one of the hottest pieces of technology at that time. The fraud was organized by the "Radio Pool", a group of investors that artificially pumped RCA to the incredible price of $549, and then dumped the shares making the price plummet to under $10. A large number of investors lost all of their savings in this operation. At that time, communication was done through the radio, tabloids, and word of mouth.

With the advent of the hectic and almost non-regulated markets of cryptocurrencies, pump and dumps are more vital than ever. There are now hundreds of cryptocurrencies, the market is not strictly regulated, and prices are easy to manipulate. Thus, pump and dump schemes are incredibly common, with public groups on the Internet, rules, and precise and complex organization. Now, pump and dumps are led by a large number of self-organized groups over the Internet, and the phenomenon is viral though still not very well known. As of January 2021, a new kind of operation has been in the global spotlight. A group of people active on a Reddit group called r\wallstreetbets started an operation against a few hedge funds shorting GameStop stocks (GME). The group was able to attract other people and managed to raise the stock price of GME by more than 1,900% [243]. The event got worldwide attention, and several celebrities, including Elon Musk, rock star Gene Simmons, and rapper Snoop Dogg [389] commented on the event and contributed to making it even more popular. Following the success of this operation, people collaborated into buying other stocks such as AMC (AMC Entertainment Holdings), BB (BlackBerry Ltd.), and NIO (NIO Inc.) and later the Ripple (XRP) and DogeCoin (DOGE) cryptocurrencies. Also in these cases, prices increase rapidly in a few days [292].

In this work, we describe the pump and dump phenomenon in the cryptocurrency ecosystem, focusing on the organization of the groups and the frauds. We perform a 3 years longitudinal analysis of the pump and dump operations on 4 different exchanges. Then, we analyze the events arranged by Big Pump Signal, a pump and dump group that moved 5,176 BTC (around $300M as of today) in a single operation. Lastly, we introduce a novel detection algorithm that works in real-time. The algorithm is not just based on the detection of the abrupt rise of the price. The fundamental idea is to leverage the abnormal growth of so-called *market buy orders*, buy orders that are used when the investor wants to buy extremely quickly and whatever is the price. Just like the colluding members of a pump and dump group when the pump starts. Moreover, we describe a new kind of pump operation—that we refer to as crowd pump to distinguish it from the standard pump and dump, discussing the differences in the organization and aim between the standard pump and dump and the crowd pump.

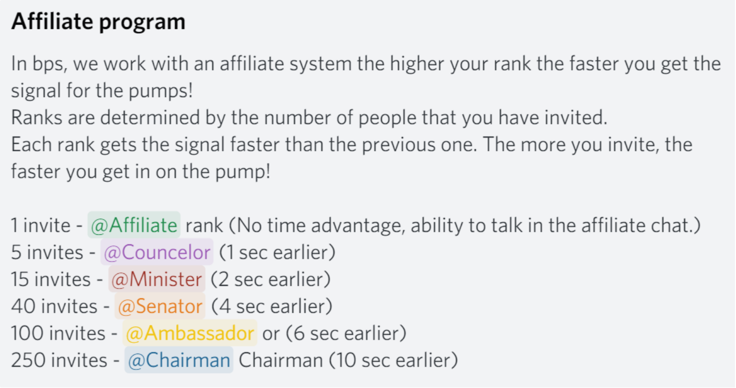
Our main contribution are:

- **Pump and dump dataset.** We publicly released our dataset [343] containing more than 1,000 confirmed pump and dump events arranged by 20 different Telegram groups.
- **Pump and dump detection model.** We propose a novel real-time machine learning model, showing that it outperforms the current state of the art [212], improving the expected speed of the detection from 30 minutes to 25 seconds and, at the same time, the F1-score from 62.7% to 94.5%.
- **Crowd pump analysis.** We conduct an in-depth analysis of the crowd pump events carried out on the DogeCoin and Ripple cryptocurrencies. Collecting and analyzing the messages on Reddit, we reconstruct the way these events occurred and how they started. Lastly, we show that it is possible to use the proposed machine learning model to detect when a crowd pump is in action.

The work presented in this chapter was published in the ACM Transactions on Internet Technology (TOIT) in 2023. In this project, I worked with my supervisor Alessandro Mei and professor Julinda Stefa and Massimo La Morgia from Sapienza University of Rome.

4.1 Pump and dump groups

Pump and dump schemes are performed by self-organized groups of people over the Internet. These groups arrange the frauds out in the open on the Telegram [246] instant messaging platform or Discord server [200]. Thus everyone can join the groups without prior authorization. Along our longitudinal research, from July 2017 to January 2021, we joined and followed all the activities performed by more than 100 groups daily. Being members of the groups allowed us to retrieve and collect one-of-a-kind information such as internal group organization, the phases of pump and dump arrangement, and how the groups attract outside investors inside the



Affiliate program

In bps, we work with an affiliate system the higher your rank the faster you get the signal for the pumps!
Ranks are determined by the number of people that you have invited.
Each rank gets the signal faster than the previous one. The more you invite, the faster you get in on the pump!

1 invite - @Affiliate rank (No time advantage, ability to talk in the affiliate chat.)
5 invites - @Councelor (1 sec earlier)
15 invites - @Minister (2 sec earlier)
40 invites - @Senator (4 sec earlier)
100 invites - @Ambassador or (6 sec earlier)
250 invites - @Chairman Chairman (10 sec earlier)

Figure 4.1. Affiliate program and benefits of the Big Pump Signal group.

market. In the following section, we report on the findings we discovered about these communities.

4.1.1 Group organization

Pump and dump groups have leaders (or admins) that administrate the group, and a hierarchy of members. If a member is higher in the hierarchy, he gets the message that starts the pump by revealing the target cryptocurrency a few moments earlier than lower ranked people. This way, the member has a higher probability of buying at a lower price and make more money from the pump and dump operation. The advantage in terms of time of being at a higher level is usually between 0.5 and 1 second with respect to the next level, and the maximum advantage is in the interval between 1 and 10 seconds. Most groups are organized as an affiliation system —climbing the hierarchy is possible by bringing new people into the group. The larger is the number of new members brought to the group, the higher the ranking. Fig. 4.1 shows the affiliation system of the Big Pump Signal group and the rank's benefits.

Some groups have a simple hierarchy with only two levels: Common members and VIP members. In these groups, to become a VIP the user has to pay a fee, usually in Bitcoins, in the range of 0.01 to 0.1 Bitcoins (from approximately $310 to $3,100 at current exchange rates[1]). In the pump and dump groups, the admins are the only people that make decisions. We saw only in rare cases the admins running polls to decide the hour of the pump or the exchange to use but never to decide the target cryptocurrency.

4.1.2 Group communication

The groups typically use Discord servers and Telegram channels to communicate and organize the pump. Telegram [246] is an instant messaging service, and a Telegram channel is a special kind of chat in which only the owner of the channel can broadcast public messages to all the members. Discord [200] is a VoIP and text chat service. It was originally designed for video gaming communities, but nowadays it is widely

[1] Data retrieved on January 10, 2021

used by communities not related to video games [230]. Discord offers the possibility to create macro sections and host multiple chat rooms. Each section has its own topic or scope. In our analysis, we have found that all the pump and dump Discord servers are organized in roughly the same fashion, with the following sections:

- **Info & How-Tos**: These two sections are like an electronic bulletin board with pinned messages. Both sections are composed of several rooms that contain only one or very few messages. The rooms of the Info section usually contain the rules of the group, the news about the group, how the affiliation system works (Fig. 4.1), and the F.A.Q.. The rooms of the How-Tos section contain manuals related to the cryptocurrency world or the best practices to participate in a pump and dump operation.

- **Invite**: This section contains rooms where the bots of the server live. Here, the users can query the bots to generate invite links to bring new members or to know the number of people that joined the server by using their invite links.

- **Signal**: This is the core section of the group, in which only the admins can write. Usually, there are two rooms in this section: The pump-signal and the trading-signal. In the first room, the admins share info about the next pump and dump operation. In the second, they share trading advice.

- **Discussion**: In this section, there are rooms covering different topics where the group members can freely chat.

Usually, the messages written in the news and in the pump-signal rooms are also broadcasted to the Telegram channel.

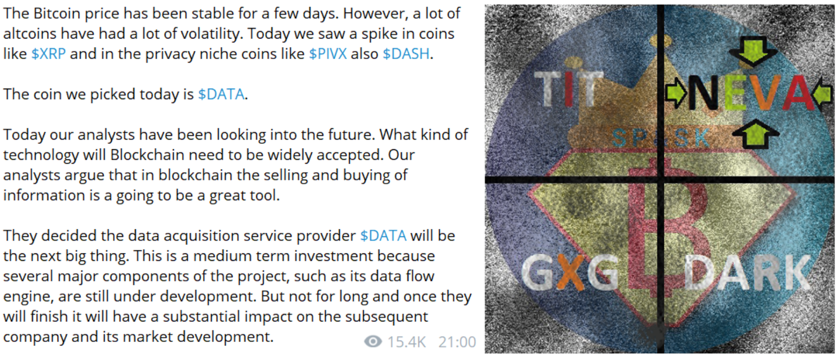
4.1.3 Organization of the pump and dump operations

The levels of activity of the many pump and dump groups on the Internet differ considerably. The most active ones perform roughly one pump and dump operation a day. Less active groups perform one operation a week. Other groups perform operations only when they believe the market conditions are good. The steps during the operation are typically as follows:

- A few days or hours before the operation the admins announce that the pump and dump will happen and communicate which is the exchange that will be used, the exact starting time of the operation, and whether the operation will be FFA (Free for All—everybody gets the message at the same time) or Ranked (VIPs and members of higher levels in the hierarchy get the starting message before the other members).

- The announcement is repeated several times, more frequently as the starting time of the operation gets closer.

- A few minutes before the start, the admins share some simple tips and best practices: Check your Internet connection, buy low and sell high, disconnect all the other Internet activities to get low latency on your network, hold the currency as much as possible waiting for an external investor. At this



The Bitcoin price has been stable for a few days. However, a lot of altcoins have had a lot of volatility. Today we saw a spike in coins like $XRP and in the privacy niche coins like $PIVX also $DASH.

The coin we picked today is $DATA.

Today our analysts have been looking into the future. What kind of technology will Blockchain need to be widely accepted. Our analysts argue that in blockchain the selling and buying of information is a going to be a great tool.

They decided the data acquisition service provider $DATA will be the next big thing. This is a medium term investment because several major components of the project, such as its data flow engine, are still under development. But not for long and once they will finish it will have a substantial impact on the subsequent company and its market development.

15.4K 21:00

Figure 4.2. Messages that indicate the start of a pump and dump operation on the Streamr DATAcoin (on the left) and the NevaCoin (on the right).

point, the free chat rooms are closed in order to avoid so-called FUD (Fear, Uncertainty and Doubt)—sometimes due to actual human anxiety of losing money, sometimes due to activities of disinformation done by people that have the goal of sabotaging the operation, make people panic and make the panic spread in the group. This is also useful to avoid any possible overload on the communication server.

- At the established time the targeted cryptocurrency is revealed, the exact time depends on the position in the hierarchy of the group. Usually, the name of the cryptocurrency is contained in an image that is obfuscated in such a way that only humans can read it correctly. Fig. 4.2 shows an example, a message that instructs to start a pump and dump operation on the NevaCoin. The idea behind the obfuscation is to make it hard for bots to parse the message with OCR techniques and start the operation faster than humans.
- A few seconds after the start of the operation, the admins share a piece of news and invite all the group members to spread the information that the price of the cryptocurrency is rising. This is done in dedicated chat boxes, forums, and Twitter. This activity aims to attract external investors by creating FOMO—Fear of Missing Out a unique investment opportunity.
- Finally, when the operation ends, the admins reopen the free chat rooms and share some statistics about the pump with the members.

4.2 Case study

In this section, we present three case studies. In the first, we perform an analysis of the pump and dump groups, the targeted exchange, and the cryptocurrencies. In the second, we focus on Big Pump Signal, arguably the biggest pump and dump group, able to generate a volume of transactions of 5,176 BTC in a single operation. Lastly, we present the case study of the Yobit exchange that organized 3 pump and dump operations in 2018. We analyze these frauds and leverage the users' comments on Twitter regarding these events to understand the feeling of the crypto-community about the phenomenon.

Table 4.1. Metrics of Telegram pump and dump channels.

Channel name	Members	Hierarchy	Main Exchange	PnD (#)	avg. Volume ($)
BigPumpSignal	72, 097	affiliation	Binance	41	7,245,437
Trading Crypto Guide	91, 725	vip	Binance	22	2,442,923
Crypto Coin B	166, 689	vip	Binance	12	5,733,637
Crypto4Pumps	11, 716	vip	Bittrex	45	491,395
Pump King Community	7, 771	vip	Bittrex	14	931,960
Luxurious Crypto	6, 020	free	YoBit	17	4,997
AltTheWay	7, 333	free	YoBit	253	700

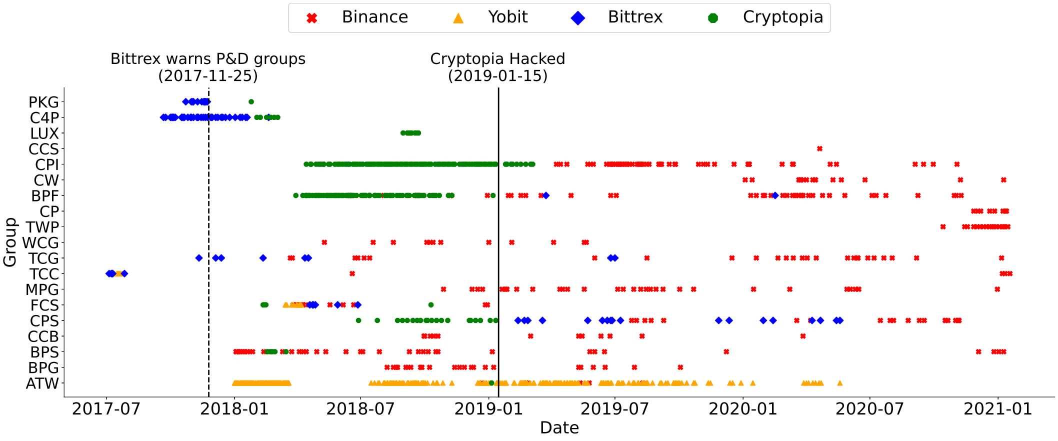


Figure 4.3. Pump and dump events by group and exchange during the period of the analysis.

Table 4.2. This table reports the acronym that we will use in the charts, the extended name, and the Telegram link of each monitored group.

Group code	Group name	Telegram link
TCC	Trading Crypto Coach	https://t.me/tradingcryptocoach
TCG	Trading Crypto Guide	https://t.me/TCGFORYOU
BPS	BigPumpSignal	https://t.me/bigpumpsignal
BPG	BigPumpGroup.com	https://t.me/bigpumpgroup_com
MPG	Trading Mega Pump Group	https://t.me/mega_pump_group
C4P	Crypto4Pumps	https://t.me/Crypto4Pumps
PKG	Pump King Community	https://t.me/pumpingking
ATW	AltTheWay	https://t.me/AltTheWay
LUX	Luxurious Crypto	https://t.me/LuxuriousCrypto
CPS	Cryptopia pump squad	https://t.me/cryptoflashsignals
CCB	Crypto coin B	https://t.me/CryptoCoinsCoach
FCS	Fast Crypto Signals	https://t.me/fastcrypt
WCG	Whales Crypto Guide	https://t.me/Whalesguide
TWP	Today We Push	https://t.me/TodayWePush
CP	CrypticPumps	https://t.me/CrypticPumps
BPF	Big Binance Pump Family	https://t.me/rocketpumptrader
CW	Crypto Waves	https://t.me/CryptoCoinsWaves
CCS	Coin Coach Signal	https://t.me/CoinCoachSignals
SE	Signal Express	https://t.me/signalexpresss
CPI	Crypto Pump Island	https://t.me/crypto_pump_island

4.2.1 The groups, the exchanges, and the target cryptocurrencies

We conduct an in-depth investigation of the cryptocurrencies and the exchanges used for the pump and dumps. We do so in a period that goes from July 2017 to January 2021. In this period, we found more than 100 groups, by keywords search (*e.g., : "Pump", "Dump", "Signal"*) on Telegram, Twitter, Reddit, BitcoinTalk [60], or manually extracting information from CoinDetect [99] or the PADL [286] Android app. From this set, we select 20 different groups since the others are not very active or have a small number of users. In Table 4.2 we report the extended name of the groups we monitored, their Telegram link, and the short version of the name that we will use in the charts for each group.

Table 4.1 shows a few metrics about different kind of groups with respect to hierarchy, number of users, and number of pump and dump operations[2]. Reading the Telegram channel history of these 20 groups, we found evidence of 1,108 pump and dump events carried out on 4 exchanges. We discovered that 206 of these operations were jointly arranged by more than one group. Hence we have in our dataset 902 unique pump and dump operations. Analyzing our data, we found that the scheme involved 378 different cryptocurrencies, only 340 of which CoinGecko still lists. CoinGecko is a service that exposes APIs [100] to retrieve historical trading data. Instead, leveraging the CryptoCompare API [111] we were also able to retrieve

[2] Data retrieved on October 2018

the market capitalization, at the time of the fraud, for 264 coins. Analyzing the volume of the 24 hours before the pump of the target cryptocurrencies, we can see that 284 (83.5%) of them moved less than $1 million in total in all the exchanges. 182 (53.5%) of them moved a negligible amount of money, less than $10,000. Also, analyzing the market capitalization of 264 coins, we find out that 140 (71%) coins are below the $20 million of market capitalization, with 44 (22%) below $1 million.

The market capitalization of targeted coins is low, considering that the first asset with less than $20 million is at the 616*th* position of the cryptocurrency ranking by market capitalization[3]. Typically, Binance is the market of choice for the pump and dump operations on currencies with higher market capitalization, Cryptopia for those with lower market capitalization. In particular, the median market capitalization of the cryptocurrencies for exchange is $25,574,192 for Binance, $2,619,703 for YoBit, $2,512,627 for BitTrex, and $144,373 for Cryptopia. Thus, the target cryptocurrencies of pump and dumps have a very low net worth value and a vast circulating supply. Lastly, we find that 264 (78.3%) assets are priced below $0.4. As such, with a relatively small investment, pump and dump groups can buy huge amounts of cryptocurrencies and easily increase their price in the pump phase of the fraud.

Figure 4.3 shows the number of the pump and dump operations by group and exchange. The figure shows that the groups typically work on one or a couple of exchanges. That is quite normal. Indeed, if the groups jump from one exchange to the other, the members would be forced to move their assets according to the selected exchange, pay the withdrawal and the network fees, and waste their time. Sometimes, the groups move from one exchange to another due to external circumstances. For instance, 2 out of 3 groups that operated mainly on Cryptopia suddenly changed the target exchange after Cryptopia was hacked in January 2019. In contrast, the third group waits almost a month before moving to Binance. Another example is when Bittrex warned the community about its intention to ban users involved in pump and dump operations [61]. We can note that before the warning (the dashed line on the figure), 42 out of 54 (77.8%) operations are arranged on BitTrex. After, only 48 out of 817 (5.9%). From a longitudinal perspective, it is possible to note that, until May 2019, pump and dumps are evenly distributed among the four exchanges. After this date, Binance has become the most popular exchange among the groups.

Looking at Figure 4.4, we can see that most of the pump and dump events are organized in the late afternoon of the European time-zones. Hours in which European web users are more active, according to [231, 165]. Moreover, the Binance exchanges do not allow US citizens to use their service. This information could indicate that the admins and the members of the groups under investigation are mostly Europeans.

4.2.2 YoBit

YoBit is a Russian exchange active since August 2014. In October 2018, it processed almost $1 billion, and it was the 43rd exchange by monthly traded volume. In October 10th, 2018, YoBit announced on Twitter that it is arranging a pump and

[3] According to CoinMarketCap data retrieved on February 18, 2021

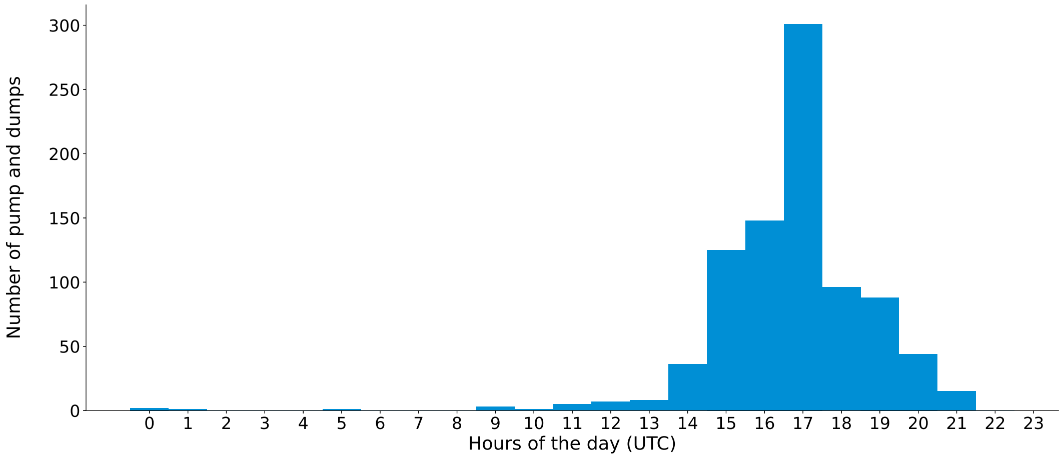


Figure 4.4. Pump and dumps during the hours of the day.

Table 4.3. YoBit pump and dumps.

Cryptocurrency	Date	Volume($)	Open price($)	Max price($)	Price increase
Putin Coin	2018-10-10	955,077	0.0075	0.1131	1408%
Lambo Coin	2018-10-15	980,645	158.08	320,000	2024%
Chat	2018-10-17	661,109	0.0320	0.3839	1099%

dump event. More in detail, they claimed that they would buy 10 Bitcoins of a random coin, in a range of 10 minutes. After the first pump, done on the Putin Coin, YoBit repeated the event twice—on October 15th on the Lambo Coin and on October 17th on Chat. All three cryptocurrencies had practically no transactions—in the 24 hours before the pump the three coins moved $36, $800, and $59 respectively. Table 4.3 shows the volumes and the prices during the events. The price of the Putin Coin, for example, has reached a peak of 14 times the opening price. The huge volumes and the high prices of the coins during the events went back to their original state a few hours later.

This unprecedented behavior of an exchange hit the news [187, 112, 361] and the community started to tweet about it. We collected all the tweets sent as a reply to the announcements by YoBit to see the impact that the event had on the community and their feelings about pump and dump schemes. We got 517 tweets, among which we removed 46 tweets containing images only and 157 tweets not related to the pump and dump events. We analyzed the remaining 314 tweets with the Google Cloud Natural Language API [173] to get the sentimental score on the reaction of the community. After the analysis, we got that 46.5% of the tweets had a negative sentiment on the events. 42% a neutral feeling, and only 11.5% of the tweets a positive feeling. Moreover, several crypto-influencers on Twitter strongly commented against YoBit, such as Rudy Bouwman, co-founder of DigiByte, the 37th cryptocurrency by market capitalization at the time.

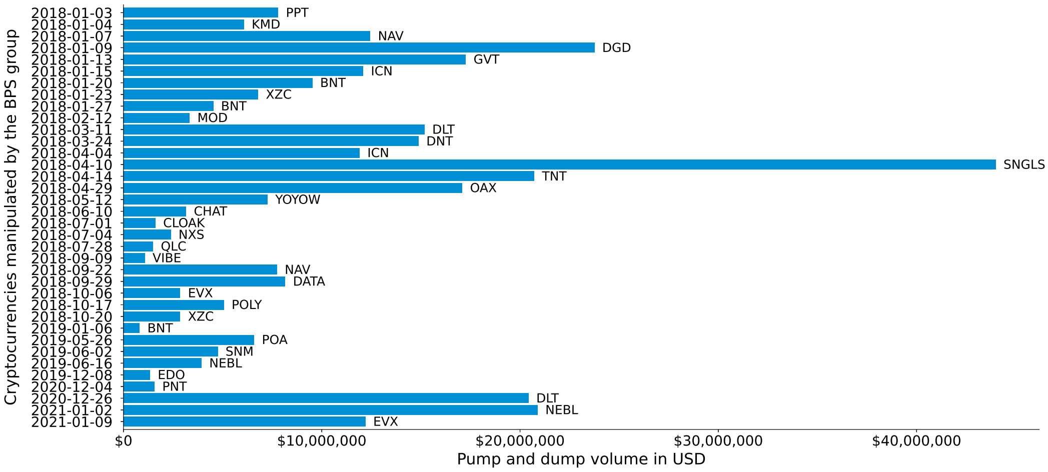


Figure 4.5. Big Pump Signal pump and dump operations.

4.2.3 The Big Pump Signal group

With a peak of more than 200,000 members on Telegram and 250,000 members on Discord in January 2018, Big Pump Signal (BPS) is arguably the largest pump and dump public community on the Internet. Reading the pump announcements on the Telegram channel of Big Pump Signal, we found 41 pump events organized by them, 36 of which carried out on the Binance exchange and 5 on Cryptopia. Figure 4.5 shows the operations carried out by the group and their volume. Throughout all their pump and dump operations, the group moved globally \$129,674,881 within an average of 5 minutes from their start; including the time interval of the whole operations, the group moved globally \$343,433,660. Their most successful pump and dump was arranged on May 10, 2018, when they targeted the SingularDTV (SNGLS) alt-coin. In this operation, the value of the SNGLS coin sharply oscillated for more than 9 hours and recorded a volume of around 5,176 Bitcoins.

BPS has an affiliation hierarchy. The highest level is achievable after inviting 250 new members. In ranked pump and dump operations, the affiliation guarantees the members to receive the signal between 1 and 10 seconds before the unranked members. Since the beginning, the Big Pump Signalers have promoted their group by advertising on social networks like Twitter and Quora. Thanks to their aggressive marketing campaigns and the hype on cryptocurrencies in late 2017, the Big Pump Signal group has grown extremely fast.

As the group grew larger, the admins started also targeting cryptocurrencies with medium market capitalization. The admins claim that they base the choice of the coin on technical analysis. They also claim to re-pump the cryptocurrencies by collaborating with a small investment firm. The investment firm is believed to be frequently involved in organizing pump and dumps on their own. Examples are the pump and dumps of the Monetha coin (MTH) and the WePower (WPR) coin on the Binance platform on September 17, 2018. Our analysis shows that BPS typically chooses cryptocurrencies with a steady price and news coverage in the recent past. They leverage the news coverage to generate interest and attract external investors. An example is the retweets of news from the fake Twitter account of John McAfee

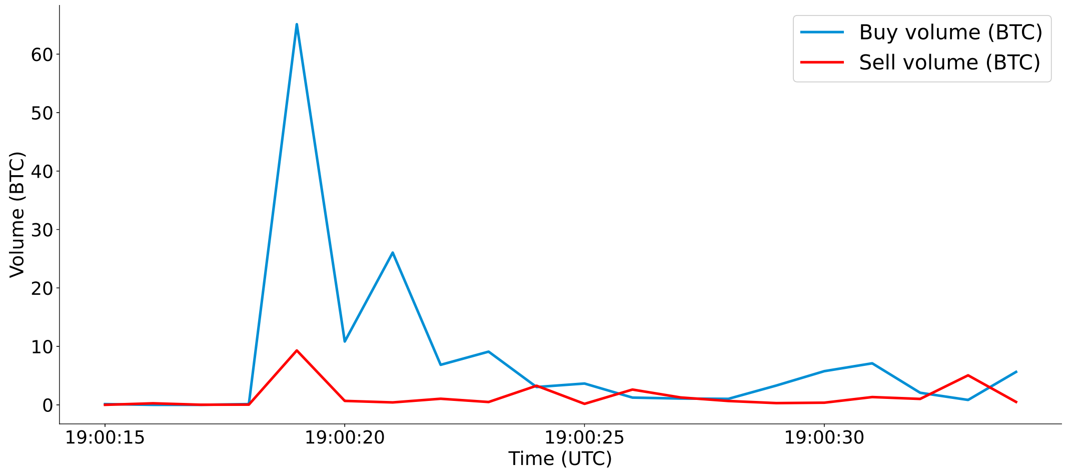


Figure 4.6. Pump and dump on the OAX coin.

(*e.g.,* @oficiallmcafee, and @TheJohnMcafee) belonging to the admins of the group.

Analysis of the BPS pump phase

The BPS group moves large volumes of Bitcoins in each operation. Figure. 4.6 represents a zoomed image of the very first 30 seconds of the pump on the OAX cryptocurrency. The blue line in the figure represents the volume of buys; the orange line the volume of sells. We observe that the buy and sell volume in the first seconds is very close to zero. Then, there are two buy peaks (blue line in Figure 4.6) of approximately 65 Bitcoins (sec. 19) and 26 Bitcoins (sec. 21) respectively. The two peaks correspond to the actions of VIPs and the common members—a normal behavior, considering that the group has a ranked policy. We also observe a peak in the sell volumes (orange line in Figure 4.6) of almost 10 Bitcoins at the moment of the first buy peak, the 19th second. Considering that group members are still buying and the reaction time for outsiders is too short, this sudden big sell volume is abnormal. There can be only two possible actors to sell their assets: the bots and the admins. To discern between the two, we need to investigate the single transactions. Our analysis shows that, as the price rises, there are many small sell operations at incremental values, probably by the arbitrage bots. Then, we observe a last single shot transaction for over 4 Bitcoins when the OAX coin reaches the trading value of 0.00012 BTC, probably done by the admins of the group. We believe they have operated through a *sell limit* trade order—a conditional order triggered when the price of a trading pair reaches/out-tops a given value. Of course, the same order could have also been placed by an outside investor. However, we believe that a sell limit of that amount, 41% more than the initial price, is most likely due to an insider.

4.3 Pump and dump detection

4.3.1 The idea

As we know, standard investors are the victims of pump and dump schemes. When they see that the price of a cryptocurrency rises, they can believe it can be a good investment opportunity. This is not the case when a pump and dump scheme is in action—the rise does not have economic grounds. It is just market manipulation. In order to protect investors, it is crucial to understand if we can detect a pump and dump in action and how quickly. This is the goal of this section.

To better understand how we can detect pump and dumps, it is essential to have some basic notions. The pending orders for a cryptocurrency, like securities, are listed in the *order book* for that cryptocurrency. The book is a double sorted list of sell (ask) and buy (bid) orders not yet filled. The asks are sorted from the lowest price to the highest, the bids from the highest to the lowest. The fastest way to buy on the market is through a *buy market order.* A buy market order looks up the order book and fills all the pending asks until the requested amount of currency is traded. Although a market order completes almost instantly, the price difference between the first and the last ask needed to fill the order can be very high, especially in markets with low liquidity. So, the total cost of the order can be unpredictably high. A more careful investor would use *limit buy orders*, orders to buy a security at no more than a specific price. Buy market orders are not frequent in everyday transactions, and investors use them when they need fast execution, just like the members of pump and dump groups in action. Our idea is to use this pattern and other information about volume and price to detect when a pump and dump scheme starts.

4.3.2 The data

As highlighted by Kamps et al. [212], it does not exist a dataset in the literature of the confirmed pump and dumps. Thus, we need to build one for this work. From the 20 groups we joined, we selected only the pump and dump schemes carried out on Binance. We made this choice for two main reasons. The first one is that Binance exposes APIs [55] that allow retrieving every single transaction in the whole history of a trading pair differently from other exchanges. The second is that pump and dumps on other markets are usually carried out by groups with few active members and economic resources. These groups can only target alt-coins that have almost no volume of transactions for days before the scheme. Thus, we believe that pump and dumps carried out on Binance are the most interesting and challenging to detect.

From the initial set of pump and dumps, we select all the events on Binance—317 pump and dump events. We retrieved the historical trading data for each pump and dump for 14 days, seven days before and seven after the event. Some pump and dump are a few days apart on the same alt-coin, so we discarded duplicate days. In the end, we have globally about 900 days of trading. The data are a list of trade records: Volume, price, operation type (buy or sell), and the UNIX timestamps. The records belonging to the same order at the same price have aggregated quantities, and a single order filled at different prices is split into more records.

Unfortunately, the Binance APIs do not tell the kind of order (*e.g.,* : *Market, Limit, Stop Loss*) placed by the buyer, so we need to infer this information. To do this, we can use the fact that market orders complete instantly, and we can aggregate the buy operations filled at the exact millisecond as a single market order. Since we do not know the original nature of these orders, we define them as *rush orders*. A problem with this inference method is that it misses the market orders that are filled by the first ask of the order book. Still, we believe we have a good witness of market orders' abrupt rise even with this approximation. As a contribution to the community, we will publicly release this dataset [343].

4.3.3 Features and classifiers

To detect the start of the fraudulent scheme, we analyze several kinds of features. Then, we use them to feed two different classifiers: Random Forest and AdaBoost. Random Forest [67] is an ensemble learning method consisting of a collection of decision tree classifiers such that each tree depends on the values of a random vector sampled independently, each tree casts a vote, and the prediction is the most popular class between all the votes. AdaBoost [146] is a meta-estimator that ensembles multiple weak classifiers—a classifier that performs slightly better than a random guess into a stronger one. It starts by training a weak classifier that assigns the same weight to all the dataset instances. It then fits additional copies of the classifier on the same data, tuning the weights in favor of the previously misclassified instances. In our case, the weak classifier is a Decision Tree [320] with a maximum depth of 5. We built our features upon the idea of [332] for the detection of Denial of Service attacks through an adaptive threshold. Since we do not want to find a threshold in our case, we rework their idea in this way: We split data in chunks of s seconds, and we define a moving window of size w hours.

We conduct several experiments with different sets of features and settings regarding the window and the chunk sizes. Since our goal was to build a classifier that detect a pump and dump scheme as soon as possible from the moment it starts, the chunk size must be reasonably short. At the end of our study, we achieved the best F1-score with a chunk size of 25 seconds and a window size of 7 hours; and the best speed with a chunk size of 5 seconds and a window size of 35 minutes. Here are the features we used:

- **StdRushOrders** and **AvgRushOrders**: Moving standard deviation and average of the volume of rush orders in each chunk of the moving window.

- **StdTrades**: Moving standard deviation of the number of trades.

- **StdVolumes** and **AvgVolumes**: Moving standard deviation and average of the volume of trades in each chunk of the moving window.

- **StdPrice** and **AvgPrice**: Moving standard deviation and average of the closing price.

- **AvgPriceMax**: Moving average of the maximum price in each chunk.

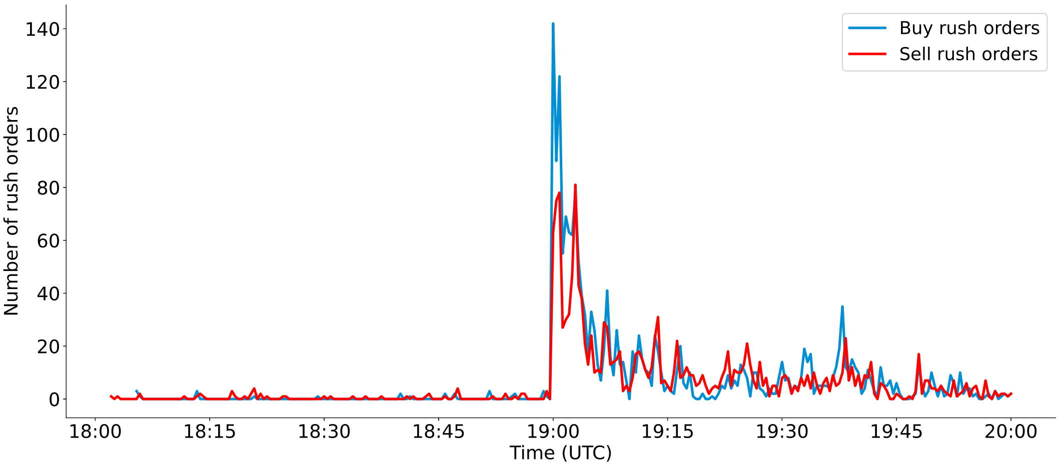


Figure 4.7. Number of rush orders during the pump and dump on VIBE cryptocurrency.

- **HourSin**, **HourCos**, **MinuteCos**, **MinuteSin**: The hour and minute of the first transaction in each chunk. We encoded this feature with the sine and cosine functions to express their cyclical nature.

Once a pump is detected, we pause our classifier for 30 minutes to avoid multiple alerts for the same event.

4.3.4 The importance of rush orders

In this section, we explore how rush orders are important to detect the start of a pump and dump operation.

Fig. 4.7 shows the number of buy and sell rush orders during a pump and dump scheme on the VIBE cryptocurrency on September 9th, 2018. As we can see, rush orders are rare during the hours before the pump and suddenly grow just at the start of the scheme. Comparing the number of buy and sell rush orders, we notice that buy rush orders are more prevalent than sell rush orders at the start of the pump operation. This is expected since the first part of the operation, the pump phase, consists of buying the asset as quickly as possible. For this reason, we consider only the number of buy rush orders as a feature for our machine learning models. Moreover, sell rush orders may indicate other phenomena (*e.g.,* , panic selling) and lead to false positives.

We perform an experiment to understand if the rush orders are a practical feature to detect the beginning of a pump and dump scheme and find a threshold beyond which the growth can be considered abnormal. To learn the threshold, we proceed as follows: we compute the *StdRushOrder* feature as described in Section 4.3.3. Then we label each chunk as True if the timestamp of the pump and dump signal falls into the chunk time range, False otherwise. We randomly split our dataset into the train (50%) and test (50%) sets, we compute the precision-recall curve for the train set, and we pick a threshold that is a trade-off between the precision and the recall. Then we evaluate the same metrics at the picked threshold for the test set. Fig. 4.8 shows the results. We choose 12.8 as the value for the threshold (the black dashed line in the figure). This value provides a precision of 81.2% and a recall of 91.1%

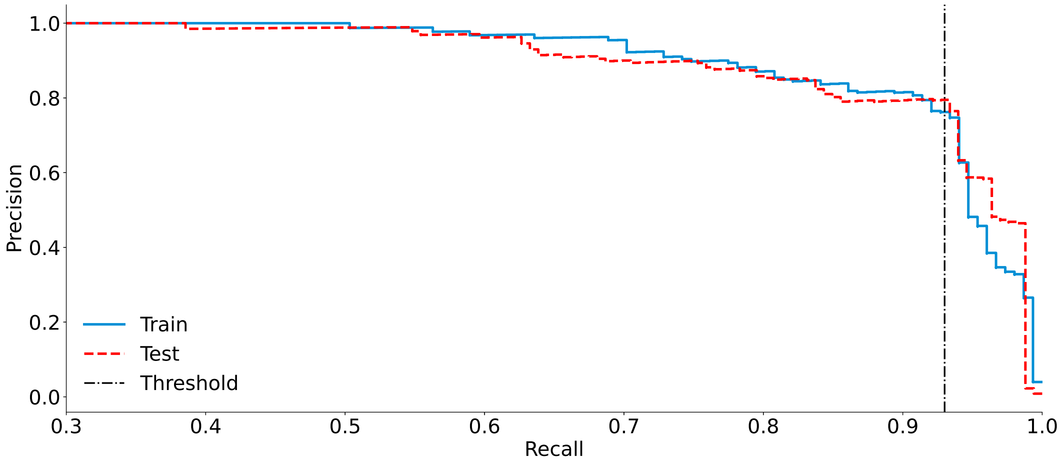


Figure 4.8. Precision recall curve for train and test sets.

on the train set (the blue line). As we can see, the same threshold also provides a very similar score on the test set (the red dashed line). Given these results, we can claim that the rush orders feature is an excellent parameter to evaluate the start of a pump and dump.

4.3.5 The results

Although we retrieved 2 weeks of data for each pump and dump scheme, initially, we use only 3 days—the day of the fraud, the day before, and the day after. We can reasonably assume that no other scams are present for the same coin in this time frame. Indeed, among the market manipulations we collected, different groups arranged schemes on the same alt-coin a few days apart. However, we are aware that some groups delete the pump and dump signal from the chat history and that there are groups that we cannot monitor, such as groups that communicate in Chinese or Russian. Since our dataset consists of 317 pump and dumps, we do not split the dataset into the standard train test sets. We performed a 5 folds cross-validation to get a more reliable performance evaluation.

For the Random Forest classifier, we use a forest of 200 trees and a maximum depth of 5 for each tree. Table 4.4 shows that the Random Forest classifier has outstanding results in terms of precision. However, the recall drops quickly, from 91.2% to 72.9%, when we reduce the chunk size from 25s to 5s. To address this issue, we introduce a new approach with respect to the one used in previous work [232] that leverages an AdaBoost classifier. This approach is more balanced in terms of precision and recall and has better results in terms of F1-score. Moreover, from the classifiers' results, it is possible to note the relationship between the chunk size and the performance of the classifiers. Indeed, while the precision is relatively stable in all the time frames, the recall increases as we increase the chunk size.

In Tab 4.5, we list the importance, computed with the Gini Impurity, of each feature used with the Random Forest classifier. As we can see, the best are the ones based on the rush orders and the number of trades. Once we defined our methodology, we trained a 25-second detector classifier with the 3 day dataset and

Table 4.4. Classifiers performance with K-Fold cross validation.

Classifier	Chunk size	Folds	Precision	Recall	F1
Kamps (Initial)	1 Hour	-	15.9%	95.3%	27.2%
Kamps (Balanced)	1 Hour	-	38.9%	93.2%	54.9%
Kamps (Strict)	1 Hour	-	52.1%	78.8%	62.7%
Random Forest	5 Sec	5	94.6%	72.9%	82.4%
Random Forest	15 Sec	5	96.4%	84.9%	90.0%
Random Forest	25 Sec	5	98.2%	91.2%	94.5%
AdaBoost	5 Sec	5	90.0%	79.2%	84.2%
AdaBoost	15 Sec	5	91.7%	87.7%	89.7%
AdaBoost	25 Sec	5	95.4%	90.9%	93.1%

Table 4.5. Features importance.

Feature	Importance
StdRushOrders	0.251
AvgRushOrders	0.123
AvgVolumes	0.081
StdTrades	0.073
StdVolumes	0.073
AvgPriceMax	0.055
AvgPrice	0.032
MinuteCos	0.031
MinuteSin	0.022
StdPrice	0.013
HourSin	0.011
HourCos	0.003

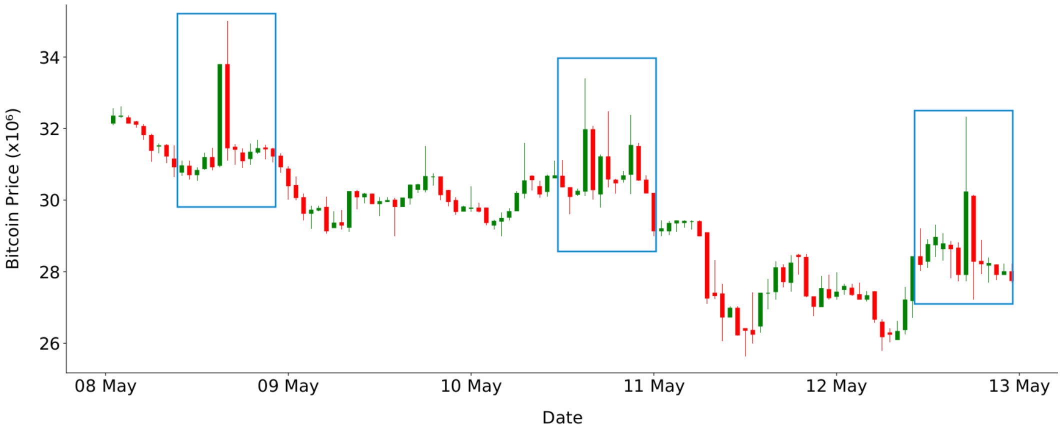


Figure 4.9. DLT candlestick chart.

used the remaining two weeks of data (more than 14 millions 25-second chunks) as a test looking for other suspect events. After the evaluation, we got 86 events that we are not able to link to evidence. Looking at the dynamics of the events, we believe that virtually all of them are pump and dumps whose evidence has been deleted or organized by groups that may not be public or that we cannot monitor.

Fig. 4.9, for example, shows the candlestick chart for the Agrello coin (DLT) from May 8 to 13. The event in the center is a pump and dump for which we have evidence. The other two are suspects detected by the algorithm. As you can see, the behavior is almost the same, including the fact that the currency quickly returns to its usual price (the dump). In any case, our classifier, based on the detection of the abnormal presence of rush orders and not just on the price, does a good job in detecting pump and dumps and suspect events that, anyways, the mindful investor wants to stay away from.

Long range experiment

In the previous section, we found 86 alleged pump and dumps events we are not able to link to evidence. These events can raise some concerns about the use of our model in a real scenario. Thus, we perform an experiment to assess the reliability of our detector over long time-frames. We test our detector over three very different cryptocurrencies: Ethereum [73], Algorand [84] and Bread [66]. Ethereum and Algorand are, respectively, high and medium market-cap cryptocurrencies. As mentioned in Section 4.6, these assets are unlikely to be the target of pump and dump events. Thus, we can assume that every alert of our detector on these cryptocurrencies is a false positive. Instead, Bread is a low market cap cryptocurrency with higher volatility. This means that this asset is more prone to quick market oscillations as well as market manipulations. Moreover, it is the most targeted by pump and dump according to our dataset. We consider all transactions performed on the three cryptocurrencies from their listing on Binance (2017-07-14 for Ethereum, 2019-06-22 for Algorand, and 2017-12-2 for Bread) to the end of the analysis (2021-01-31). For Ethereum, our classifier finds 24 suspicious events over a period of 1,276 days. We obtain similar results on Algorand, where our classifier raises only 19 alerts on 591

Table 4.6. Results for the long range experiment.

Cryptocurrency	Days analyzed	Events found
Ethereum (ETH)	1,276	24
Algorand (ALGO)	591	19
Bread (BRD)	1,156	17

trading days (on average, one false positive every month). Finally, for Bread, we find 41 pump and dump events on more than 3 years of data, 24 of which are present in our dataset and 17 are suspicious. Thus, one suspicious event every 2 months. Table 4.6 summarize the results of our experiments. At the light of this experiment, we believe that our detector can be handy in an real usage scenario, even if raises some false positive (less than 1 per month per monitored crypto).

4.3.6 Comparison with other pump and dump detectors

To the best of our knowledge, the best detector of pump and dumps in the literature is Kamps et al. [212]. We use their algorithm as the baseline. Their detector takes as input candlesticks of 1 hour. So, the best performance, in expectation, is of 30 minutes. Their methodology leverage two anomaly thresholds: Transaction volume and coin price. They compute the values of the thresholds using windows on the recent history of the candlestick under observation. If both the price and the volume are higher than the calculated thresholds, they mark the event as a pump and dump. Kamps et al. provide three different parameter configurations to compute the threshold: Initial, Balanced, and Strict. The Basic configuration maximizes the recall, the Strict the precision, while the Balanced is a trade-off of the two. In their work, they mention the number of alleged pump and dumps that their classifier detects. Unfortunately, they do not provide scores in recall and precision since their dataset lacks ground truth.

To use the Kamps et al. detector as the baseline for our task, we implemented their classifier and tested it on their dataset. We detected the same number of pump and dumps they report in their work. Then, we apply their methodology to our dataset—Table 4.4 shows the results. As we can see, all our classifiers outperform in terms of F1-score the Kamps et al. detectors. Our detector's performance is considerably better, we score 98.2% of precision and 91.2% recall against their 52.1% precision and 78.8% recall, and our detector is faster as well. These results also show that, due to the cryptocurrency market's high volatility, the detectors based only on the coin price and transaction volume are prone to many false positives.

Differently from our work, Xu et al. [391] build a classifier able to predict the next pump and dump's currency target to provide a tool for strategic trade. Since the goals are different, we can not make a comparison in terms of performance between our work and their solutions. Indeed, they prefer to maximize the probability of gain from the investment, maximizing the recall at the expense of low precision. Xu et al. assume that buying wrong currencies does not affect the trading strategy because, on average, this does generate an economic loss. Instead, in our case, we want to provide a reliable approach—with high precision and recall— to help investors stay out of the market when a pump and dump scheme is in action and to analyze anomalies in historical data.

4.4 The Crowd Pump

Now, we focus on a new kind of pump operation. We will call it *crowd pump*—a pump and dump event that results from the non-directly organized actions of a crowd of people. We analyze how these operations happen, and we illustrate the differences from standard pump and dumps. Lastly, we offer that it is possible to leverage our dataset to build a classifier that can also detect crowd pump events.

4.4.1 A description of the crowd pump phenomenon

In January 2021, the stock market was puzzled by an unprecedented rally of GameStop (GME). The GME stock had been gradually losing value for a couple of years, as sales of physical copies of video games plummeted due to the shift towards digital purchases [211]. During the COVID-2019 pandemic, the situation worsened to the point that GameStop announced it would close more than 1,000 stores by April 2021 [178]. GME quickly became easy prey for short-sellers, economic agents that bet on the fall of specific securities. Short-sellers borrow stocks, sell them, and buy them later, when the price is expected to be lower, to give them back to the lender.

This operation would have gone unnoticed, as this market practice, albeit somewhat controversial, is common. The turning point came when a group of users active on Reddit [201], one of the most popular social news aggregation and discussion websites [120], started to buy large quantities of GME stocks. These users communicated in a *subreddit*—a user-created board that covers a specific topic—called *r\wallstreetbets*. Initially, the users started to invest in the GME stocks because they believed they were undervalued. Only later they began to do it as a political statement against hedge funds [285]. The operation was a great success and the subreddit users managed to raise the stock price of GME by more than 1,900%, from $17.25 on January 4 to $347.51 on January 28 [243]. Due to the media interest, the subreddit gained more than 3 million followers in that period. GME became the most traded stock in the U.S. stock market on January 26 [242]. Due to the results of this operation, people started collaborating to buy other stocks such as AMC (AMC Entertainment Holdings), BB (BlackBerry Ltd.), and NIO (NIO Inc.). Their prices increase rapidly in a few days [292]. In response, several digital trading services like Robinhood began restricting trades on the stocks that were getting pumped [155].

Due to these limitations, the attention moved to cryptocurrencies—less regulated and still with a combined market capitalization that topped $1 trillion [113]. The first coin to get widespread attention was Dogecoin. Dogecoin was originally founded as a joke on December 6, 2013 [284]. The price of the coin skyrocketed on January 28, 2021, after a Reddit group, called *\SatoshiStreetBets*, proposed to make it the equivalent of GME for the cryptocurrency market. Dogecoin had an increase in the price of over 800% in 24 hours, from $0.0077 to $0.07 according to data from CoinGecko [101]. The price increased in several distinct phases, driven by the tweets of well-known personalities like Elon Musk, rock star Gene Simmons, and rapper Snoop Dogg, reaching its highest value ever of $0.079 [69].

The second target was the Ripple (XRP) crypto-coin. At the time of these events, the XRP suffered a challenging moment due to a lawsuit that started on December 22, 2020. The SEC accused Ripple of performing illegal security offerings of $1.3 billion in XRP for seven years beginning in 2013 [249]. This action caused a drop in the coin price from $0.42 on December 22 to $0.18 on January 4. Several exchanges delisted XRP. Including Coinbase, one of the largest [278]. The delisting reduced the liquidity of the coin significantly, creating the perfect breeding ground for market manipulations [64]. In this case, the operation was organized on a Telegram group called "Buy & Hold XRP FEB 1st, 2021" that was later renamed "BUY & HOLD XRP FEB 1st, 2021 @8:30AM" [169]. The group grew exponentially in the 24 hours following its creation, reaching the limit of 200,000 members of Telegram. The group aimed to buy massive quantities of XRP at a precise date and hour—February 1, 2021, at 13:30 UTC. However, many members started buying it massively the days before the pump, and the cryptocurrency jumped 56% up in price, reaching the biggest single-day percentage gain since December 21, 2017 [155]. So, the price was already high at the pump, and the group could not increase it any further.

4.4.2 Analysis of crowd pumps

Although it is well-known that the DogeCoin pump starts from some popular subreddits [268], it is unclear who started the pump and how they carried out the operation. We analyze all the Reddit users' posts on the subreddits mentioned above to answer these questions. A *submission* is the first post of a new discussion thread and may contain links, text, and images. To perform our analysis, we downloaded all the submissions from January 01, 2021 to February 02, 2021 of some popular crypto-related subreddits: *r\SatoshiStreetBets, r\WallStreetBets, r\Cryptocurrencies, r\DogeCoin* subreddits. We downloaded data from these subreddits since in the period the terms *"Doge"* and *"DogeCoin"* appear mainly in them, according to the Redditsearch tool [303]. To retrieve the submissions, we leveraged Pushshift [48], a service that provides access to Reddit data overcoming the limit of 1,000 posts of the official APIs.

We globally retrieved 656,146 submissions, of these 626,700 (95.5%) from *r\WallStreetBets*, 23,485 (3.6%) from *r\SatoshiStreetBets*, 5,443 (0.8%) from *r\Cryptocurrencies* and lastly 518 (0.1%) from *r\DogeCoin.* From the downloaded data we took into account only the submissions that contain the name of the coin ("DOGE", "DogeCoin") and some of their very popular variations used in the cryptocurrencies slang, such as: "DOGIE", and "DOGUE". In the end, we get 27,868 submissions with the following partition: 19,016 (68.2%) from *r\WallStreetBets*, 8,383 (30.1%) from *r\SatoshiStreetBets*, 194 (0.7%) from *r\Cryptocurrencies*, and 275 (1%) from *r\DogeCoin.* Finally, we study the message distribution over time and their relationship with the price of DogeCoin.

Figure 4.10 shows the number of submissions posted in the subreddits that mention DogeCoin (solid blue line) and the price of DogeCoin in Bitcoin (dashed gold line). As we can see in the upper left chart of the figure, subreddits rarely mention the coin in the weeks before the pump, and the price is stable. In the

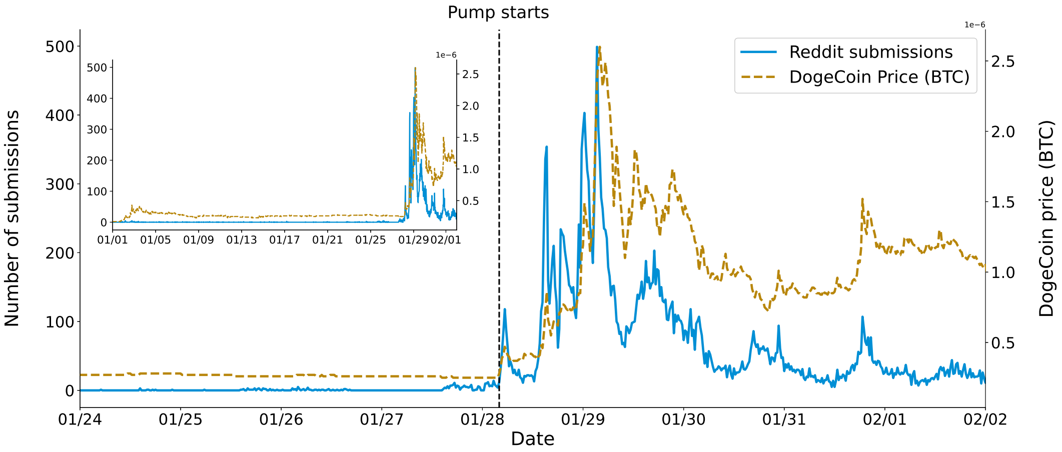


Figure 4.10. Number of submissions mentioning DogeCoin sent on the subreddits vs. DogeCoin price in BTC.

24 hours before the pump (vertical dashed line), it is possible to note that some submissions about the coin begin to pop up steadily. However, the price is still stable. After the vertical dashed line, the coin gets a massive spike in popularity, and the price abruptly rises. From this moment, the price of DogeCoin and the number of submissions on Reddit follow the same pattern.

In the light of this analysis, we dig into the posts before the pump. The goal is to understand how the users arranged the operation. We find out that most of these posts tried to drum up the attention on the DogeCoin proposing to pump the currency. Initially, the users did not welcome these posts. The administrators often removed the content because it violated the netiquette of the subreddit. Among these submissions, we found a particularly interesting one on the r\DogeCoin subreddit. Here, a few users were trying to arrange a pump on the DogeCoin on January 28 at 10 AM, 5 hours later than the actual start of the pump. Nonetheless, none of these submissions had any effect on the price of the DogeCoin, as shown in Fig. 4.10. In our opinion and news [214], the message that triggered the rally of the DogeCoin, for timing and users welcoming, was posted on January 28, 2021, at 4:05:50 UTC and states: *"Let's make DOGIECOIN a thing. That's it, that's the post"*. The submission had only the title, no message body, and no picture.

To better understand why this message triggered the pump, we investigated the creator of the submission, expecting her to be popular on the Reddit community. Surprisingly, we found out that, although the user is very active on Reddit with more than 854 submissions and 769 comments, only 4 submissions (0.4%) and 17 comments (1%) are related to crypto or finance. Thus it is doubtful that the author is a crypto-influencer, and it is hard to understand why so many users followed this message.

We performed a similar analysis also on the crowd pump carried out on the Ripple cryptocurrency. For this case study, we analyze the messages on Reddit in the same time frame we did for the DogeCoin, since the two events occurred within a few days of each other. We consider the same subreddits of the previous analysis, with the exception of $r\backslash DogeCoin$ subreddit and including the $r\backslash XRP$

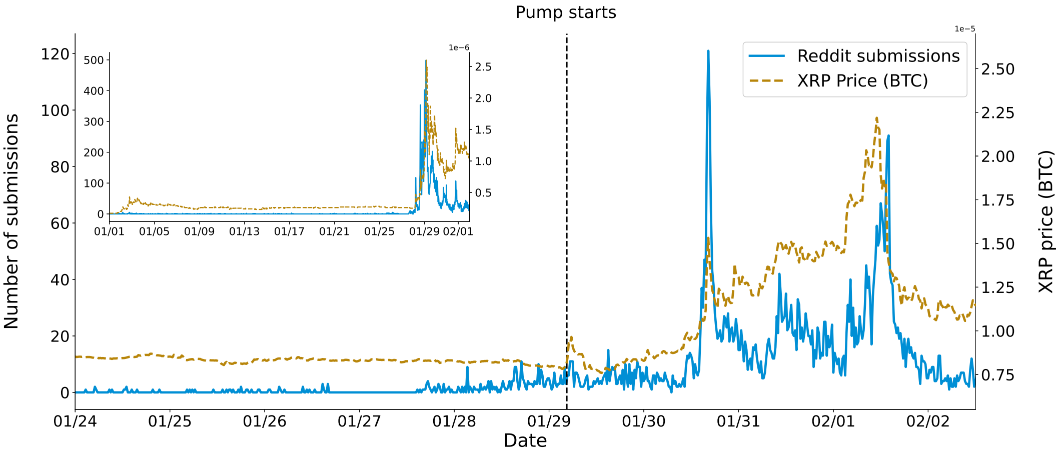


Figure 4.11. Number of posts mentioning XRP vs XRP price in BTC.

(5,444 submissions), obtaining globally 661,072 submissions.

In this case, we focus on the submissions that mention one of the cryptocurrencies. Figure 4.11 shows the number of posts in the subreddit that mention Ripple (solid blue line) and the price of Ripple (dashed gold line). As we can see, the coin is rarely mentioned in the weeks before the pump, while it starts to get attention in the days before the pump. Similar to what happened in the case of the DogeCoin pump. Reading these messages, we find out that the cause of this increase in the posts is due to Redditors driven by anti-SEC sentiment and inspired by the DodgeCoin and GME pump operations. The birth of the Telegram group *"OFFICIAL BUY & HOLD XRP"* gathered these users, and group members began to promote the group itself. Different from the DogeCoin crowd pump, where the number of posts on Reddit and the cryptocurrency price seem to follow the same trend, in this case, the two lines seem to be more independent, except for the price peaks. Analyzing the beginning of the pump (solid dashed line in Fig. 4.11), it is possible to note that the price quickly rises while the number of submissions on Reddit does not. Some hours later, the price returns to its real value (January 29 at 5:00 UTC), and then the price increases again (January 30 at 16:00 UTC).

This behavior makes us suspect that the pump does not start from Reddit. Thus, we investigate the messages sent on the Telegram group, for which we were able to export all the messages, files, videos, and images. Since, to the best of our knowledge, the group is no longer accessible, and the group chat is not publicly available, we publicly release it as a further contribution [344]. The Telegram group counted exactly 200,000 members and 45,548 messages. We do not know when the group was created, but the first message appeared on January 28 at 20:19:09 UTC. Unlike pump and dumps, the organizers did it on a Telegram Group instead of a Telegram Channel. Hence, all the group members could write in the chat, not only the admins. After the creation of the group, the chat was open, and the members could freely talk about the event and how to participate. However, the situation escalated around January 29 at 5:00 UTC. From this moment, maybe for a slight fluctuation of the Ripple's price or an extra-group coordinated action of a set of users, the members start to urge the chat to *BUY!* the coin, starting the pump way earlier than expected. This event occurs almost at the same time as the first spike in price that we see in Figure 4.11. The admins promptly reacted by turning off the chat and resumed it only twice before the day of the pump—the first time on January 30 at 20:05 UTC, the second one on January 31 at 6:03 UTC. In both cases, the chat opened only for 30 minutes, and the admins asked the member of the groups to indicate from which countries they were posting. Then, the chat was opened again 9 hours before the pump for a few seconds. As discussed before, the pump was a failure as the group could not further raise the price of the coin.

At the end of our analysis, we find the following main differences between crowd pump and pump and dump operations:

- **Different goal:** The aim of a crowd pump is not to inflate the price of an asset and sell it to scam unaware investors. In these kinds of operations, the organizer and part of the community often encourage the participants to hold their stock to keep the value high. We noticed this attitude in both the crowd pump events carried out on the crypto market. A clear example is the crowd pump organized on the XRP currency. In this case, the group creator clearly states in the Telegram group chat that the operation aims to hold the currency. The admin also publishes a disclaimer video on his Youtube channel explaining the purpose of the group. Quoting the description of the video: "This is not a "pump and dump" group. This is a community-led event to bring awareness to the XRP ledger" [71].

- **Lack of coordination and leadership:** Even if we saw on both the crowd pump events attempts to coordinate to buy at a specific hour, they always failed. Unlike standard pump and dump, the organizers reveal the coin to pump in advance. Thus, people start to buy the coin in advance or when they believe the operation has begun. A simple fluctuation of the market or a single post can trigger a ripple effect that leads to the start of the pump.

- **Different time frame and price increase rate:** As we saw, in standard pump and dump, the operation lasts for a few minutes or rarely for a few hours, and the price grows almost immediately. In a crowd pump, the price increases abnormally, but it takes hours or days before the coin reaches its maximum peak. This behavior is due to several factors. The goal is different, and some investors do not immediately sell the coin to take a profit. No one knows when the pump will start. Therefore it can take time before the crowd realizes that the operation has begun. Finally, the news and influencers work as an echo chamber, and more and more people join the process making the price of the coin increase in waves. Consequently, while in standard pump and dump the price of the coin returns to its natural level as the event ends, in crowd pump and dump, after more than a month[4], the price of the DogeCoin is still 500% higher than its pre-pump value, and the XRP is stll 100% higher.

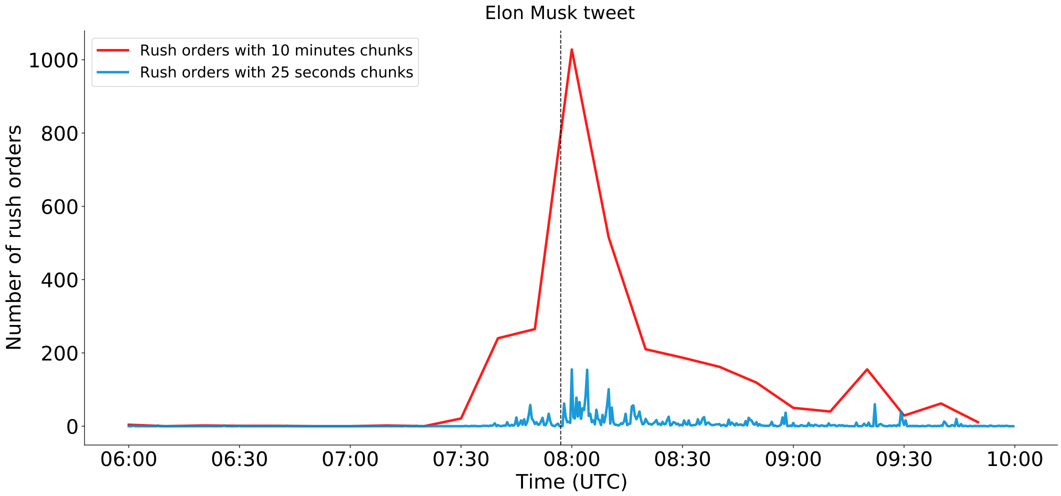


Figure 4.12. Number of rush orders before and after the tweet of Elon Musk about Dogecoin (February 4 at 7:57 UTC).

4.4.3 Crowd pump detection

In this section, we assess the potential of our machine learning model in detecting crowd pump operations. Although there are some key differences between the crowd pump and standard pump and dump, our intuition is that the rush orders are a very relevant feature also in this kind of operation.

In particular, we consider the number of rush orders in an interval of two hours around the publication of a tweet of Elon Musk that shill the DogeCoin [69]. We make this choice because, in this case, we have the timestamp of the tweet and we can be sure about the moment in which the operation starts. Figure 4.12 shows the number of rush orders in two hours around the publication of the tweet. The purple line represents the number of rush orders grouped in chunks of 25 seconds, while the red line in chunks of 10 minutes. In the figure, it is possible to note a considerable amount of rush orders after the tweet, precisely like in pump and dump events after the admin announces the target coin. However, looking at the purple line (25 seconds chunk), we find that the pattern of the rush orders is very different from the one we see for the standard pump and dumps (Figure 4.7). Indeed, there is no neat big spike in the number of rush orders, but a gradual increase with several small spikes. This behavior is not surprising. There is no synchronization of the investors—they jump into the market in waves depending on when the message hits the social platforms on the web and when they see it.

Due to this different behavior, our detector trained on the standard pump and dumps cannot capture the crowd pump analyzing short chunks of transactions. Moreover, we cannot efficiently train a new detector for the crowd pump operations because of the lack of a dataset. However, expanding the chunk's time frame size makes it possible to collapse the different waves of rush orders into a unique chunk and get a well-outlined spike. The red line in Figure 4.12 shows the number of rush orders grouped in chunks of 10 minutes. Here, we can see that the pattern is very

[4] March 2, 2021

similar to a pump and dump operation, like the one we reported in Figure 4.7, and thus it is now reasonable to think that our detector can find these kinds of events.

The new model

To detect the crowd pumps, we trained a new classifier based on the Random Forest algorithm like the one used to detect standard pump and dumps. This time we trained the model on the full dataset (317 pump and dump events) described in Sections 4.3.2. We used the same feature we leveraged to build the previous detector, except for the one related to the time. We removed these features because they are specifically tailored for the standard pump and dumps carried out by Telegram groups. The new detector achieves an F1-score of 89.4% in 5 fold cross-validation. In the case of crowd pumps, we test our approach only on two events: XRP and DOGE. For the training phase, we used 25 seconds chunks. Instead, we aggregate the trading data in chunks of 10 minutes for the test phase. After detecting an event, we pause our classifier for 6 hours to avoid multiple alerts. In this case, we pause the classifier longer than we did for the standard pump and dumps because the operations last more time.

The Dogecoin pump

To find out if our detector can catch the start of the Dogecoin crowd pump, we downloaded all the transactions from Binance from January 1 to February 10, 2021. Even though we know that the pump happened on January 28, 2021, we run the detector for some weeks before the pump to check if any suspicious activity is detected and to validate the classifier's robustness on false positives. At the end of the execution, our classifier detects the following 5 events:

1. **January 2, 2021, at 3:00 UTC:** At first sight, the event seemed a false positive. However, after a search on the web, we found that the news [156] reported a price surge of the DogeCoin driven by a tweet from the adult film star Angela White. The actress stated that she is a DogeCoin investor since 2014. The tweet features a photo of the actress wearing a T-shirt with a Shiba Inu image, the DogeCoin mascot, and received more than 10,000 likes.

2. **January 28, 2021, at 4:10 UTC:** This alert falls exactly in the same chunk of the Reddit post that sparkled the DogeCoin popularity in the *r\SatoshiStreetBets* subreddit, discussed in Section 4.4.2.

3. **January 28, 2021 at 14:20 UTC:** It is not easy to link this warning to an individual event. However, investigating on Reddit and Twitter, we find on Reddit an abrupt increase of messages that mention the coin (see Figure 4.10). Moreover, between 14:00 UTC and 15:00 UTC in the U.S.A., the hashtag *"#dogecoin"* became a trending topic on Twitter, with more than 91,000 tweets. 2 hours later *"#dogecoin"* became a worldwide trending topic, accordingly with the data provided by ExportData.io [135] and TT-History [197]. In our opinion, it is safe to assume that this alert detected many investors that have flooded the market.

4. **January 28, 2021, at 23:40 UTC:** This is very likely due to Elon Musk's tweet (January 28, 2021 at 22:47 UTC) on Dogecoin. In particular, the tweet contains a picture that mimics the Vogue magazine with a dog picture on the cover, and the title of the magazine changed to "Dogue." More than 450,000 users liked this tweet. The detector raised the alert about one hour later. However, looking at the price evolution of the DogeCoin in the hour following the tweet, it is possible to note that investors enter into the market slowly. Indeed, at the time of the tweet, the price of the DogeCoin was at $0.024; at the time of the alert, the price was $0.03 (+25%). The coin reached its first peak in price one hour later, touching $0.05 (+108%). Then, around the 4:00 UTC of January 29, the coin achieved $0.08 (+233%) its maximum price of the month.

5. **February 4 at 8:00 UTC:** This alert is also related to a Tweet of Elon Musk. This time he posted a tweet that contains a meme portraying him as Rafiki from the Lion King—the animated movie, standing on Pride Rock and raising a Doge-headed Simba [179]. In this case, our detector captured the abnormal market movements 13 minutes after the tweet has been posted (*i.e.* the very first chunk computed after the tweet). Unlike the previous tweet, this one gets much more attention, with more than 1 million likes on the social network, and the market reacts faster. In this case, our classifier detects the event when the price of the DogeCoin was at $0.04, while the coin reaches its price peak at $0.06, almost one hour later.

The Ripple pump

Again, for the Ripple crowd pump, we run our classifier on all the transactions closed on the Binance exchange from January 1 to February 10, 2021. In the considered time frame, the detector raises the following 4 alerts:

1. **6 January, 2021, at 14:40 UTC:** To the best of our knowledge, this alert is not related to the Reddit community. Instead, the news that a petition to the White House to stop the SEC lawsuit against Ripple hits 35,000 signatures [89] has probably caused new trust in the XRP coin. The price went from $0.23, at the moment of the alert, to $0.37 (+38%) of the following day.

2. **19 January, 2021, at 5:50 UTC:** This is the exact moment when several exchanges, including Coinbase, the 3^{th} exchange by volume of transactions, delisted XRP from the trading pairs [278]. The delisting follows the SEC lawsuit. Two hours after the alert, we record an abrupt rise in transaction volume on Binance and the price from $0.29 to $0.33 (+9%). The alert is probably due to trading bots or investors that moved their assets from one exchange to another.

3. **29 January, 2021 at 5:00 UTC:** This is when we noticed the excitement in the "OFFICIAL BUY & HOLD XRP" group, with the members of the group that start to urge to buy the coin. As we discussed in Section 4.4.2, the users' excitement comes together with the beginning of the XRP rally.

4. **30 January, 2021 at 16:00 UTC:** For this alert, we do not have clear evidence of what triggered the event. However, in the hour before this alert, the Ripple cryptocurrency starts to hit the news, becoming one of the most searched words worldwide on Google [175, 74]. At the same time, the number of posts on Reddit about the Ripple cryptocurrency increased dramatically. Driven by the news, an odd number of investors may have joined the market and started to buy in a rush the currency to avoid missing a good profit opportunity, triggering our detector.

It is important to note that when the pump was scheduled (February the 1st, 2021, at 13:30 UTC), the detector did not raise any alert. This is not surprising since, as discussed before, the pump failed [168]. Thus, the classifier detected the start of the pump two days before, catching the users that bought the coin in advance.

Looking at the results we achieved, we believe that our first attempt to build a classifier to detect crowd pump events shows excellent results. Nonetheless, we could further improve our detector by combining features from social media and related to the market exchanges' financial transactions.

4.5 Related work

The pump and dump phenomenon is older than the cryptocurrency revolution. Therefore, a vast portion of the literature is about pump and dumps done in the traditional stock market. Allen et al. [26] identify three categories of market manipulation schemes: Information-based, action-based, and trade-based. The pump and dump schemes are usually a combination of information-based and trade-based manipulation. In 2004, Mei et al. [258] show that it is possible to carry out pump and dump schemes just leveraging the investors' behavioral biases. They test their theory on the pump and dump cases prosecuted by the SEC from 1980 to 2002, which confirms their hypothesis.

Several case studies highlighted that emerging markets were prone to pump and dump schemes. Khwaja et al. in [215] show that the limited Pakistani regulation on the national stock exchange allowed brokers of the Karachi Stock Exchange to arrange pump and dump schemes. Jiang et al. [206] investigate the stock pools scheme of the '20s using daily trading volume from the New York Stock Exchange between 1927 and 1929. The stock pools are groups of traders that delegate to a single manager to trade stocks on their behalf. Since a pool can move a large amount of money, it can increase the volume of trades and attract outsiders to the market. When the stock pool exits the market, the price quickly drops. As reported by the University of Innsbruck in [147], the internet boom in the early years of 2000 led to the birth of a new email-based pump and dump scheme. In this new kind of fraud, the manipulators secure their position on the market and then send millions of e-mails claiming to have private information about substantial increases in the prices of specific stocks. After luring new investors, and the price higher, the manipulators sell the security and stop the spam campaign. A subsequent analysis in 2013 by Siering in [330] shows that despite the authorities have taken several countermeasures against fraudulent stock recommendations, email-based pump and dump campaigns are still flourishing.

The work of Gandal et al. [152] show evidence that the first price spike to $1000 of the Bitcoin may be market manipulation. Using the well-known dataset of the Mt.Gox exchange, they found suspicious trading activities carried out by two actors, named 'Willy bot' and 'Markus bot.' The purpose of these actors was to buy Bitcoin to increase the price and the daily volume artificially. Krafft et al. [221] investigate the behavioral patterns of the users on the Cryptsy exchange market. In their work, they show that even tiny volumes of buy trades can influence the market. They use bots to buy a small number of random currencies and conclude that traders tend to buy coins with recent activities. Li et al. [239] conduct an empirical investigation on trading data obtained from the pump and dumps from Binance, Bittrex, and Yobit, focusing on the economic point of view. They show that pump and dumps lead to a short-term increase in prices, volume, and volatility followed by a reversal of the trend after some minutes. Moreover, they show that the investors' gain depends critically on the time they obtain the signal. For this reason, outside investors are systematically disadvantaged. Victor et al. [373] perform quantification and detection of pump and dump schemes coordinated through Telegram and executed on Binance. They test their machine learning model considering 125 pump and dump collected from Telegram as confirmed pumps and the 20 most retweeted tweets of the official Twitter accounts belonging to each coin as negative samples. They do not aim to catch a pump and dump in real-time as their feature considers a 30 minutes interval and tries to capture both the pump and the dump phase. Hamrick et al. [185] conduct an analysis on Discord and Telegram, identifying more than 5,000 pump and dumps from January 2018 to early July 2018. However, they use a different definition of pump and dump, including events that we define 'signals.' With our definition, they found 704 pump and dumps. They measure the factors that lead to the success of a pump, defined as the increase in the price of the coin. Some of the most important are the volatility of the coin and the number of people in the groups. Dhawan et al. [121] study 355 cases of pump and dumps in the cryptocurrency markets. They show that pumps generate an average price distortion of 65%, abnormal trading volumes in the millions of dollars, and enormous wealth transfers between participants. They highlight that this kind of manipulation is likely to persist as long as regulators and exchanges turn a blind eye. Nizzoli et al. [279] conduct a study on 50M messages collected on Twitter, Telegram, and Discord. They highlight the existence of two different manipulations: Pump and dump and Ponzi schemes. They found that 56% of crypto-related Telegram channels are involved in manipulations and that bots massively broadcast these deceptive activities. Chen et al. [87] develop an apriori algorithm to detect pump and dump on Bitcoin using the leaked transaction history of Mt. Gox Bitcoin exchange. They do not have a ground truth of confirmed pumps. Thus, they try to find groups of users who usually buy or sell the asset simultaneously. This is possible thanks to each user's complete transaction history–information typically unavailable and protected by privacy. The work of Kamps et al. [212] shows a first attempt to detect pump and dumps using an adaptive threshold. They bring up the issue that a reliable dataset of the confirmed pump and dumps scheme does not exist, so they can not fully validate their results. A contribution of our work is to release such a dataset. Xu et al. [391] focuses on the difficult task of predicting pump and dumps, using one-hour intervals data from Cryptopia and Yobit, also showing an approach to leverage the prediction to invest in alt-coins. Since both works have some goals in common with ours, we conducted a thorough analysis of their results on subsection 4.3.6.

4.6 Discussion

Is it possible for pump and dump groups to avoid detection? We based our features on the abnormal change of some market parameters and, at the same time, to be robust against the natural oscillations of the volatile cryptocurrency market. If the admins of groups or other members buy the currency gradually, and the users are few, our classifier may not detect the pump and dump. Indeed, our classifier cannot detect four of the pump and dumps in our dataset. These four events were all carried out by one group, and all of them record a consistent pre-pump phase in the hours before the pump. Fortunately, admins cannot use this technique regularly to avoid detection. Indeed, outsiders could detect this pattern to increase the probability of predicting the target coin. Moreover, these events often fail, and most users could lose trust in the admins and leave the group.

Can pump and dump groups manipulate Bitcoin or major cryptocurrencies? To answer this question, we make a short simulation. Let us take the buy volume on the first 10 minutes of the largest pump and dump we monitored. It is 31 BTC on the SingularDTV (SNGLS). Now, we take a snapshot[5] of the exchange order book for the trading pair BTC/USD. We assume that the market is frozen and only the pump and dump group members can take action. This is the best case for raising the price. We find that the amount of money at their disposal can increase the BTC value by less than $5, which is way smaller than the natural oscillations of the coin in 10 minutes. So, the answer is no. Though these groups are very large, they cannot attack coins with large volumes like Bitcoin.

Is it possible for the exchange markets to stop pump and dump schemes? In this work, we show that it is possible to detect a pump and dump scheme as soon as it starts. We also believe that exchanges can catch better than us when a fraudulent scheme is in action. In fact, the data owned by the exchange is more fine-grained: It has complete knowledge of the kind of operations performed, their amount, and precisely who performed them during the scheme. Moreover, we notice that little policy enforcement can reduce the number of these market manipulations. As discussed before, on November 25, 2017, the BitTrex exchange announced that it actively discourages any market manipulation and will begin to punish the participants [61]. Since then, the amount of pump and dumps in the exchange drastically decreased. We counted, before the statement, more than 50 pump and dumps in the five months from July to the end of November 2017, and only 48 events in more than three years after the statement. Another countermeasure could be stopping transactions on a cryptocurrency when it gains or loses more than some threshold or giving special protection to cryptocurrencies with extremely low market capitalization and trading volumes. Moreover, some exchanges list cryptocurrencies with shallow trading volumes. De-listing these cryptocurrencies, as some exchanges do [96], could make smaller groups desist.

[5] Data retrieved on April 12, 2019

4.7 Conclusion

In this work, we conducted an in-depth analysis of the pump and dump ecosystem. We studied the relationship between the groups, the exchange, and the target cryptocurrencies in a longitudinal analysis that spans over three years. We thoroughly investigated the Big Pump Signal group and the pump and dump operations carried out by the Yobit exchange. Moreover, we introduced our classifier that leverages the *rush orders*, a peculiar kind of order that is particularly effective for the detection. The proposed classifier outperforms the state of the art both on performance (98.2% of precision and 91.2% recall against 52.1% precision and 78.8% recall) and on speed, moving the expected detection time from 1 hour to 25 seconds. Given the lack of a pump and dump dataset, as a further contribution, we release to the community our resource [343] of more than 900 confirmed pump and dumps to enable further studies on the topic. precision and 78.8% recall) and on speed, moving the expected time of detection from 1 hour to 25 seconds. Finally, we moved on the crowd pump. Here, we conducted the first analysis based on data on this kind of operation and show the potential of a purely market-based approach to detect these kinds of events. We achieved promising results, catching the start of the two operations a few minutes after their opening and detecting the abnormal market conditions driven by tweets of celebrities. We think that this work helps to understand a complex phenomenon, improves the awareness of the investors interested in the cryptocurrency market, and can help the authorities regulate this particular market in the future.

A possible future direction for this work is to integrate inside our system new features based on information extracted from social networks like Reddit or Twitter. As seen before, Figure 4.10 and Figure 4.11 show some correlation between Reddit submissions and the increase in the price of the coin. This suggests that information extracted from social networks may be used as a feature to identify crowd pumps. In particular, the advantage of integrating social media in the model may be crucial to help disambiguate cases where the rise in the price of the coin is not due to market manipulation but to solid market fundamentals. During our study, we found a large number of signal groups. These groups are more significant than the pump and dump groups and arrange operations more frequently. As future work, it would be interesting to study the impact of these groups and their activity on the market. We tested our classifier on two crowd pumps on the XRP and DOGE cryptocurrencies. It would be interesting to collect a larger dataset to further assess the performance of our detector. Finally, it would be also interesting to verify if the methodology we developed to detect pump and dumps on the cryptocurrency market can also be used to detect these market manipulation in the stock market.

Chapter 5

Token Spammers, Rug Pulls, and Sniper Bots: An Analysis of the Ecosystem of Tokens in Ethereum and in the Binance Smart Chain (BNB)

The cryptocurrency market is loosely regulated [232, 47]. Even if policymakers are moving towards building a safer environment for cryptocurrency investors [339], it is a complex task, and needs time. Meanwhile, blockchain-related technologies evolve fast, and with the birth of the DeFi [396] investors begin to move from centralized exchanges (CEX) like Binance to decentralized exchanges (DEX). DEXes are distributed Applications (dApp) for trading that run on-chain powered by smart contracts. While regulating the standard cryptocurrency market is difficult, ruling the on-chain trading platform is even more challenging. Indeed, even if the web interface of a DEX can be shut down [43], its smart contracts are still reachable and working on the blockchain.

DEX and DeFi dApp were born in the Ethereum blockchain, but DeFi services rapidly pop up on all the blockchains that support smart contracts. Although Ethereum plays a leading role in the DeFi world, with over $68 billion locked in its smart contracts, the BNB Smart Chain or BSC (former Binance Smart Chain) proposes itself as a faster and cheaper alternative.

Uniswap and PancakeSwap are the two most popular DEXes on Ethereum and BSC. They rely on the Automated Market Maker (AMM) model to handle the trading system. At the basis of the AMM model, there is the concept of liquidity pools, a smart contract that handles two tokens (trading pair) that the user can swap. Unlike CEX, where the platform defines the trading pairs, users can create their pair on DEXes and let the other users use it. However, as we will see in the following, some users abuse this freedom to carry out a series of malicious operations.

In this work, we conduct a longitudinal investigation of tokens and liquidity pools in the Ethereum and BSC blockchains. We start by parsing over 3 billion transactions of both blockchains, finding more than 1.3 million tokens and 1 million liquidity pools (Sec. 5.1). Then, we reconstruct their lifetime—the time from their creation to their last transfer, discovering that approximately 60% of the tokens have a lifetime shorter than one day (Sec. 5.2). Hence, we define them as *1-day tokens*. A tiny fraction of addresses, just 1%, is responsible for creating more than 20% of the tokens (Sec. 5.3). Surprisingly, we also find that the tokens with a very short lifetime are actively traded on liquidity pools. Albeit this phenomenon is present on both blockchains, it is more widespread on BSC. Diving into this subset of tokens, we observe that a large fraction of liquidity pools used to trade the 1-day tokens show a malicious pattern that we call *1-day rug pull* (Sec. 5.4). We analyze all the liquidity pools looking for this pattern, and we find 272,349 potential rug pulls on BSC and 21,742 on Ethereum. We estimate the cost of the operation and the gains of the organizers, finding that they earned approximately $240 million with such activity (Sec. 5.4.2). Here, we see that the success rate of the 1-day rug pull is not very high (between 40% and 60%). However, given the simplicity and the very low cost of the operation, attackers can serially arrange the rug pulls and cover a series of unsuccessful operations with a single successful one. Finally, we study how this kind of operation evolved over time, discovering that the BSC has gradually surpassed Ethereum in terms of the number of operations and gains. Moreover, we find that the operations are more prevalent during two specific events: the 2020 Defi Summer and the 2021 Altcoin season (Sec. 5.4.2).

Our key contributions are:

- **Analysis of BNB smart chain**: To the best of our knowledge, we are the first to study this young but well-established blockchain, performing a longitudinal analysis from its inception to March 2022. We study the tokens and the liquidity pools ecosystem, highlighting analogies and differences with Ethereum.

- **Short lifetime tokens and Token spammers**: We estimate the lifetime of the tokens on both blockchains. Discovering that about 60% of tokens last less than one day. Analyzing who creates the tokens, we observe that just 1% of addresses create an abnormal number of tokens (about 20-25% of tokens of the blockchains).

- **1-day rug pulls**: We investigate the presence of the rug pull pattern in 1-day tokens. We discover that on BSC, 81.2% of 1-day tokens listed on PancakeSwap have this pattern. We estimate the gains of the attackers, observing that even if the operation is very simple to arrange, given its cheap cost, it is profitable when performed serially.

- **The sniper bot 2.0**: We find the presence of sniper bots (Sec. 5.5), a particular kind of trader bot that observes the blockchain's mempool to buy newly listed tokens. To the best of our knowledge, we are the first to illustrate how this kind of trading bot works, detect their presence, and quantify their activity in the rug pull operations.

The work presented in this chapter was accepted at the USENIX Security Symposium in 2023. In this project, I worked with my supervisor Alessandro Mei, professor Massimo La Morgia and the PhD student Federico Cernera.

5.1 The Datasets

For our investigation, we build two different datasets: The *Token Dataset*, which contains all the ERC-20 (resp. BEP-20) tokens created, and the *Liquidity Pool Dataset*, which contains data about liquidity pools. Each dataset has two versions, one with data from the Ethereum blockchain and the other from the BNB Smart Chain.

We consider the whole history of both blockchains from their inception to March 2022. For the Ethereum blockchain, we process all the blocks from block 0 (2015-07-30) to block 14340000 (2022-03-07). For the BSC blockchain from block 0 (2020-04-20) to block 15854000 (2022-03-07). Given the large amount of data and the need to parse the entire blockchains multiple times, for performance reasons and to avoid overloading public nodes (*e.g.,* nodes provided by Binance [56] and Infura [202]) or services (*e.g.,* BscScan or Etherscan), we host and run an Ethereum and a BNB Smart Chain node. Finally, to query the blockchains and process the data, we use the Web3 [296] and the Ethereum-etl [257] Python libraries. Web3 is a collection of libraries that allow the interaction with a local or remote EVM-compliant node. Ethereum-etl allows extracting information from EVM-compliant blockchains and exporting it into formats like CSV or JSON. The data collection phase was performed on an Ubuntu 20.04 machine with AMD EPYC 7301 (16-Core Processor, 2.80 GHz), 1 TB of RAM, and 4 TB SATA SSD with 560/530 MB/s read and write speed. Data processing took between 24 and 72 hours each time we parsed the entire blockchain, depending on the kind of data retrieved.

5.1.1 The Token dataset

Gathering smart contracts

As a first step to building the Token dataset, we collect all the contract creation transactions issued by EOAs. As mentioned in Sec. 2.2.1, EOAs can deploy a smart contract by sending a *contract creation transaction* to the zero address. We process all the transactions in the considered time frame in BNB Smart Chain (2.6 billion transactions) and in Ethereum (1.4 billion transactions). We collect 2,195,399 and 4,420,389 contract creation transactions respectively.

However, tokens can also be created by a smart contract itself. Indeed, it could be the case that an EOA calls a smart contract method, and its execution generates a new ERC-20 (or BEP-20) compliant smart contract. In this case, the token is created with a so-called *internal transaction.* Despite the name, internal transactions are not real transactions, but rather calls performed by smart contracts. These kinds of transactions are stored off-chain—they are not visible simply parsing the blockchain.

To track the tokens created by internal transactions, we can operate in two ways: The first way is to re-execute all the transactions in the blockchain in the EVM and trace all the calls. This process is extremely expensive [342] from a computational point of view. The alternative is to scan the Event log looking for events that emit a *Transfer event*. The second way is much faster and we estimate that it loses only 12% of the total number of tokens created by internal transactions. Moreover, the missing tokens are never been used, traded or transferred, and are thus of little importance for our study (we discuss in detail the impact of this choice in Sec. 5.8). So, we parse all the logs of both blockchains, searching for smart contracts that emit a Transfer event compliant with the ERC-20 (resp. BEP-20) interface. Then, we use Etherscan [219] and BscScan [220] to retrieve the transactions that created these smart contracts and all the information.

At the end of these two steps, we have a collection of 3,087,274 and 4,534,599 smart contracts extracted from BSC and Ethereum, respectively. For each of them, we store the following information: The *address of the contract*, the *block number* in which the smart contract has been generated, the block in which the smart contract emits its last event, the EOA that deployed the smart contract or in the case of internal transactions the EOA address that triggers the first smart contract, the amount of *gas used*, the cost of the gas unit (*gas price*), the *bytecode* of the smart contract, and if the smart contract has been deployed by an EOA or through an internal transaction.

Token identification

Smart contracts are not only used to create tokens, as well as not all smart contracts that emit a Transfer event are tokens (*e.g.,* NFT contracts). Thus, we need to identify which of the retrieved smart contracts are ERC-20 (resp. BEP-20) compliant. Unfortunately, this is not a trivial task, and in the last years several works [122, 374, 88, 85, 148], attempted to face this problem with several approaches that we describe in Sec. 5.7. For our analysis, we follow the approach proposed by [374, 88] that leverage the bytecode of smart contracts.

According to the Solidity specification [236], in the bytecode, smart contract's methods are identified by signatures that consist of the first 4 bytes of the *Kekkack-256* hash of the method name and parameters' type. Thus, to verify if a bytecode of a retrieved smart contract represents an ERC-20 (resp. BEP-20) compliant token, we verify if it contains at least all the signatures of the ERC-20 (resp. BEP-20) mandatory methods. Tab. 5.1 shows the signature of the mandatory and optional methods of the ERC-20 and BEP-20 interfaces.

Of the 4,534,599 smart contracts' bytecodes retrieved on the Ethereum blockchain, we find that 389,348 (8.5%) are ERC-20 tokens compliant, and 381,551 (98%) of them also implement the optional functions of the ERC-20 interface. Instead, on the BNB Smart Chain, we find that 1,887,484 out of 3,087,274 (61%) are BEP-20 compliant, and, as for Ethereum, almost all of them also implement the optional methods of the BEP-20 interface. Although we found more smart contracts on Ethereum than in BSC (4,534,599 vs. 3,087,274), there are many more compliant tokens in BSC (1,887,484) than in Ethereum (389,348). This discrepancy suggests that BSC may be a more interesting environment to study tokens and, possibly, their misuse.

Lastly, we retrieve all the information about the identified tokens such as the name, the symbol, the number of decimals, and the total supply. To do so, we use the Ethereum-etl library and the Contract Application Binary Interface (ABI) [133].

Table 5.1. Functions and events of the ERC-20 (Ethereum) and BEP-20 (Binance Smart Chain) standard interface. We report in yellow the methods that are optional in the ERC-20 interface and in red the only method that is optional in both interfaces.

Function	Signature
name()	06fdde03
symbol()	95d89b41
decimals()	313ce567
totalSupply()	18160ddd
balanceOf(address)	70a08231
transfer(address,uint256)	a9059cbb
transferFrom(address,address,uint256)	23b872dd
approve(address,uint256)	095ea7b3
allowance(address,address)	dd62ed3e
Event	Signature
Transfer(address,address,uint256)	ddf252ad
Approval(address,address,uint256)	095ea7b3

Table 5.2. An overview of the Token dataset.

	Ethereum		BNB Smart Chain	
Contracts	Total	ERC-20	Total	BEP-20
External	4,420,389	293,688	2,195,399	1,021,427
Internal	114,210	95,660	891,875	866,057
Total	4,534,599	389,348	3,087,274	1,887,484
Total (w/o LP)	-	**323,863**	-	**1,078,016**

The ABI is an interface between two program modules. It contains the specification for encoding/decoding methods and structures to interact with the machine code and interpret the results. Through the library, it is possible to instantiate smart contracts in an object-oriented manner and call its methods using an appropriate ABI. We instantiate the token contracts using an ABI that contains the specifications of ERC-20 (resp. BEP-20) methods and call the *name()*, *symbol()*, *decimals()*, *totalSupply()* methods.

At the end of the process, we have a dataset of ERC-20 (resp. BEP-20) tokens containing all the information about the smart contracts described in Sec. 5.1.1 and the related tokens. Table 5.2 shows the number of smart contracts on both blockchains.

5.1.2 Liquidity Pools dataset

To create the Liquidity Pool dataset, we consider Uniswap, its forks, and the other protocols that leverage its smart contracts.

Uniswap has three main smart contracts: *Factory*, *Pair*, and the *Router*. The Factory contract is responsible for creating the smart contract that handles the liquidity pool and the LP-tokens. The Pair contract keeps track of the balances of the tokens in the pool and implements the AMM logic explained in Sec. 2.2.4. The

Table 5.3. An overview of the Liquidity pools dataset.

	Ethereum		BNB Smart Chain	
Events	Uniswap	Others	PancakeSwap	Others
PairC.	65,098	5,483	941,220	30,907
Mint	1,399,599	512,319	21,944,474	5,027,980
Burn	824,359	243,482	7,339,286	2,481,023
Swap	54M	27M	571M	179M

Router contract offers the entry point to interact with the liquidity pools. Thus, it is possible to swap tokens and add or remove cryptocurrencies from a liquidity pool by interacting with the Router. Each of these contracts implements a set of Events that notify their status changes.

To build our datasets, we parse the Event log of the Ethereum and BSC blockchains. Following, we report the events we look for and a brief description:

- **PairCreated:** This event is fired by the Factory contract each time a new liquidity pool is created. We find 972,127 and 70,581 PairCreated events emitted on BSC and Ethereum, respectively. From the event, we can obtain the transaction hash, the block of the creation of the liquidity pool, the address that created the liquidity pool, the address of the liquidity pool, and the addresses of the two tokens (the pair of the liquidity pool), the gas used and the price paid per gas. Analyzing the address that fired the event, we find that almost all the liquidity pools of BSC are created in PancakeSwap (96.8%), and almost all the liquidity pools of Ethereum are created in Uniswap (92.2%). Analyzing the address that fired the event and looking online for notable smart contract addresses, it is possible to have a rough idea of the diffusion of the Uniswap forks in the blockchains. In BSC, we find that PancakeSwap created most liquidity pools, with 941,220 emitted events (96.8%), followed by ApeSwap [34] (3,265 events), BakerySwap [44] (2,418 events) and Mdex [255] (1,602 events). In Ethereum, Uniswap emitted 65,098 events (92.2%), while the SushiSwap [73] Factory contract, a popular alternative to Uniswap on Ethereum, 2,637 (3%).

- **Mint & Burn**: The Pair contract emits a Mint (or Burn) Event each time an LP-token is minted (or burned). This occurs whenever a liquidity provider adds (or removes) tokens into a liquidity pool. Analyzing these events, we obtain the transaction hash and the block of the Mint (Burn) Event, the address of the liquidity pool, the address that added (removed) the liquidity, the number of LP-tokens minted (burned), the gas used, and the price paid for the gas. We find 26,972,454 Mint events and 9,820,309 Burn events on BSC, and 1,911,918 Mint events and 1,067,841 Burn events on Ethereum.

- **Swap:** This event is fired by the Pair contract each time a user swaps tokens in a liquidity pool. From the event, we obtain all the information related to the swap: The transaction hash, the block in which the swap occurs, the address of the liquidity pool used, the address that performs the swap, the number of

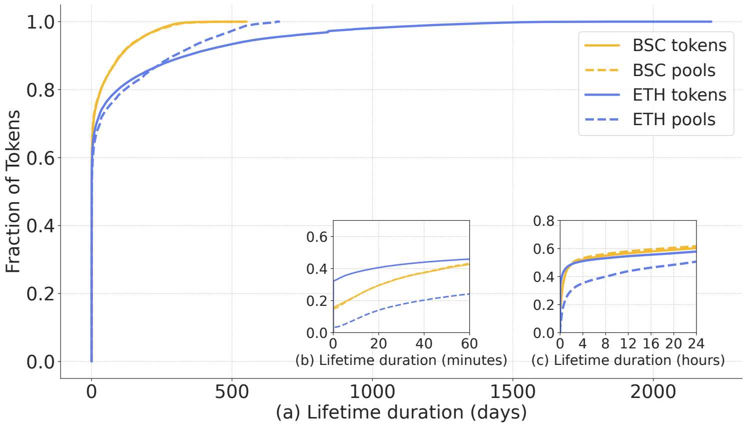


Figure 5.1. Lifetime of tokens and liquidity pools on BSC and Ethereum.

tokens swapped, the gas used and the gas price. We find 750,508,160 events on BSC and 82,447,051 events on Ethereum.

Moreover, we complete our dataset collecting for each smart contract the block number in which it emits the last event. Tab. 5.3 describes the final dataset.

Given that LP-tokens are ERC-20 (resp. BEP-20) compliant tokens, they are already present in our Tokens Dataset. However, our goal is to study standard tokens and liquidity pools separately. Thus, as the final step, we get rid of the information related to the LP-tokens from the Tokens Dataset. The last line on Tab. 5.2 reports the number of tokens after we get rid of the LP-tokens.

5.2 The Lifetime of tokens

Our data collection revealed a surprisingly high number of tokens and liquidity pools on Ethereum and BSC. Services like CoinGecko [100] or CoinmarketCap [104] list about 13,000 cryptocurrencies on 602 centralized and decentralized exchanges. Therefore, it is unclear what is the role of the large majority of tokens in the blockchain ecosystem.

To obtain a first insight into the characteristics of tokens and liquidity pools, we introduce the concept of *lifetime*. We define the lifetime of a token in the following way: A token begins its lifetime at the block where its smart contract has been deployed, while it ends its lifetime in the last block where it emits any Event. Similarly, a liquidity pool begins its lifetime at the block where the PairCreated event is emitted, and it ends in the last block where the liquidity pool' smart contract emits any Event.

Fig. 5.1 shows the CDF of tokens' lifetime and liquidity pools' lifetime on Ethereum (blue lines) and BSC (yellow lines). Tokens and liquidity pools are shown with solid and dashed lines, respectively. The slope of the curves tells that the lifetime of the tokens in BSC is generally shorter than the lifetime of the tokens in Ethereum. Consider that BSC is a young blockchain, with slightly more than two years of activity (released on 2020-04-20), while Ethereum is more than seven years old (released on 2015-07-30). The longevity of Ethereum is also visible by the

Table 5.4. Summary of 1-day and 1-block tokens for BSC and Ethereum.

Lifetime	BSC	Ethereum
1-day	638,703 (59.2%)	187,378 (57.8%)
1-block	167,318 (15.5%)	104,836 (32.4%)
Total tokens	1,078,016	323,863

long tail of its tokens in the CDF. Nonetheless, it seems that Ethereum's tokens that tend to be more solid and long-lasting. This difference is smaller when we look at liquidity pools. Indeed, PancakeSwap, which handles about 97% of the liquidity pools in BSC, was born only four months after the release of Uniswap V2. From the CDF, we can also note a few additional interesting facts, particularly when we look at the first 24 hours of the life of tokens and liquidity pools.

A significant fraction of tokens is never active. Looking at the zoomed image in the center of Fig. 5.1 (b), it is possible to see that a significant fraction of tokens have a lifetime of length zero, meaning that the token is active only in one block, when it was created. This phenomenon is more common in Ethereum, with 104,836 out of 323,863 (32.4%) tokens that belong to this category, against 167,318 out of 1,078,016 (15.5%) in BSC. In the following, we refer to the tokens that last only one block as *1-block tokens*, while to the other tokens as *active tokens*. We find 910,698 and 219,027 active tokens on BSC and Ethereum, respectively. Table 5.4 succinctly reports on these statistics.

A large part of active tokens has an extremely short lifetime. Fig. 5.1 (c) shows that about 60% of the tokens in BSC and Ethereum have a lifetime shorter than one day. We refer to these tokens as *1-day tokens*. Considering only active tokens, we find that 471,385 (51.7%) of all the active BSC tokens and 82,542 (37.7%) of all the Ethereum active tokens are 1-day tokens. Looking at the data at a higher granularity (Fig. 5.1 (b)), we can note that the death ratio of BSC tokens is surprisingly high. Proportionally, BSC has approximately half of the 1-block tokens of Ethereum, about the same proportion of dead tokens after 60 minutes, and a significantly larger proportion of dead tokens after the first 4 hours. As we can see in Fig. 5.1 (c), the first four hours of token life are also crucial in Ethereum.

Almost all the BSC tokens with short lifetimes have a liquidity pool. Here, we find one of the main differences between BSC and Ethereum. 468,556 out of 471,385 (94.8%) active tokens with a lifetime shorter than one day in BSC have a liquidity pool. In Ethereum, only 33.1% (27,346). It seems that on BSC the liquidity pool is the main reason for creating a token.

5.3 Token spammers

In this section, we change perspective and explore who creates tokens. Retrieving the list of creator addresses from our token dataset, we find 144,795 and 464,095 different addresses that create at least one token, respectively, in Ethereum and BSC. Comparing these numbers with the total number of cumulative unique addresses

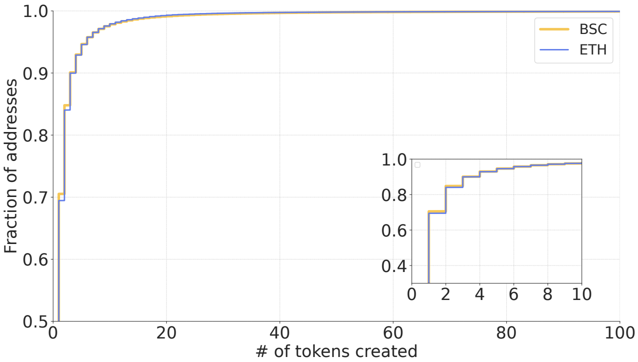


Figure 5.2. Distribution of the number of tokens created by the addresses that create at least one token in BSC and Ethereum. For the sake of visualization, the CDF is cut at 100 tokens.

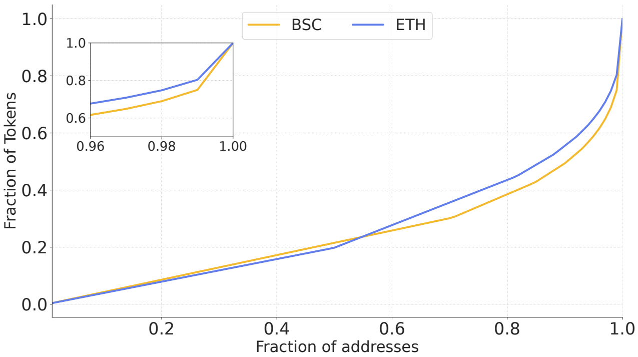


Figure 5.3. Fraction of addresses that create at least one token with respect to the fraction of tokens that they create.

on Ethereum (189,858,744) and BSC (140,522,222)[1], we see that they represent only a very small fraction of the addresses, the 0.07% in Ethereum and 0.33% in BSC. Fig. 5.2 shows the distribution of the number of tokens created by addresses in Ethereum and BSC. The first thing to notice is that the two distributions are extremely similar. The large majority of these addresses (70%) create only one token, as we can see in the zoomed image on the bottom right corner of Fig. 5.2. 95% of addresses create five tokens or less, and just 1% of addresses create more than 18 tokens.

A small fraction of addresses creates a disproportionate amount of tokens. Fig. 5.3 shows the CDF of tokens created by fraction of addresses. From the figure, we can see that although 70% of addresses create just one token, the total amount of tokens created by these addresses account for only 30% of the tokens on the two blockchains. And more interestingly, we find that just 1% of the addresses creates 24.3% (262,023) of the tokens in BSC, and similarly, 1% of the addresses in Ethereum create 20.1% (67,869) of the tokens. These addresses create an average of 51 and 61 tokens in Ethereum and BSC, respectively. We will refer to these addresses as *token spammers*.

[1] Data retrieved from Etherscan and BSCscan respectively

Token spammers are more prevalent in BSC. Although the distribution of the number of tokens created by addresses in Ethereum and BSC is almost identical (Fig. 5.3), the absolute numbers are different. Indeed, in terms of raw numbers, we find that BSC has almost four times more token spammers than Ethereum (4,231 vs. 1,329), and the spammers of BSC create almost four times more tokens in BSC than in Ethereum (262,023 vs. 67,838).

Token spammers create tokens mainly with contract creation transactions. As mentioned in Section 2.2.1, tokens can be created in two ways: By sending a contract creation transaction or by sending a transaction to a smart contract that generates the token. We find that 94.8% of the tokens on BSC and 82.3% of the tokens on Ethereum are created directly by sending a contract creation transaction.

Token spammers create short lifetime tokens. As we have seen, a significant amount of tokens have a lifetime shorter than one day. Investigating the relationship between token spammers and 1-day tokens, we discover that most of the tokens created by the spammers have a lifetime shorter than one day. The spammers created 170,768 1-day tokens out of 262,023 (65.1%) and 40,552 1-day tokens out of 67,869 (59.8%), respectively, in BSC and Ethereum.

5.4 The Anatomy of a Rug Pull

The top token spammer creates 17,936 tokens in the timeframe of our analysis. If we look at the name of these tokens, we find that almost all of them have the same name (the tokens have only 76 unique names), with the most used being 'Pornhub' with 605 occurrences. The median lifetime of these tokens is extremely small: 45 mins. Lastly, almost all of the tokens (99.7%) created by this address have a liquidity pool. We study the liquidity pools of these tokens and find out that they are used to perform an operation commonly known as rug pull [301, 253]. In the following, we report a detailed example of a rug pull operation carried out by this address.

We focus on *OnlyFans*[2], a token created by the top token spammer on block 8090747 (2021-06-07 01:40:34 PM UTC) by issuing a contract creation transaction. On block 8090751 (2021-06-07 01:40:46 PM UTC), after 4 blocks from its creation, the token spammer creates a liquidity pool that contains the pair (OnlyFans, WrappedBNB) and adds a liquidity of 20 Wrapped BNB (almost $7,180 at the moment of the operation) and 44 trillion of OnlyFans tokens.

After just 6 seconds, on block 8090753, an address swaps 4 million OnlyFans for 0.002 Wrapped BNB ($0.718). This operation is followed by 11 other swaps—performed by 11 different addresses—for a total buy of $5.1740396 * 10^{12}$ OnlyFans for 2.67 Wrapped BNB ($958). After 2 hours from the creation of the token, at block 8093101 (2021-06-07 03:38:55 PM UTC), the token spammer removes all the liquidity from the liquidity pool, leaving it drained. Since the 12 addresses added Wrapped BNB into the pool by buying OnlyFans, the token spammer collects 22.67 Wrapped BNB and has a profit of 2.67 Wrapped BNB ($958).

We can formalize these operations in the following way:

1. Eve creates a new ERC-20 token τ.

[2] 0xe8b6f08841d668605343A63144D76ff2dE9A1199

2. Eve creates a new liquidity pool with pair (τ, B), where B is a valuable token, *e.g.,* Wrapped BNB.

3. Eve adds liquidity to the liquidity pool. The reserves of the pool are now $(reserve_{\tau}, reserve_B)$.

4. At this point, Eve is the only one that owns token τ. Investors can buy token τ by swapping their tokens with token τ in the liquidity pool.

5. Suppose that Bob buys a few τ swapping it with B. The new reserves of the liquidity pool are $(reserve_{tau} - \delta_{tau}, reserve_B + \delta_B)$

6. Lastly, Eve removes all the liquidity from the liquidity pool. The net gain of the operations is δ_B minus the gas fees to execute the transactions.

An improved version of the operation. The rug pull described above is the simple version of the operation. However, to attract more investors, the attacker can manipulate some statistics of the liquidity pool. A well-known market manipulation that the attacker can use is *wash-trading* [75]. In this case, the creator of the pool tries to create the impression that the liquidity pool is active, faking the trading volume by repeatedly buying and selling tokens. Similarly, another way that attackers have to drum up the attention of investors is to inflate the price by buying the 1-day token gradually.

Finally, the attacker can also hedge his gains—eliminating the risk of an unrealized profit while the liquidity pool is still active. The attacker can maintain a reserve of tokens and, when investors start to buy the 1-day token, gradually sell the owned token, starting to take profit from the operation.

Clearly, rug pull operations can harm investors. However, we cannot consider it a "fraud" because the phenomenon is currently not regulated. E e discuss this subject in depth in Section 5.8.

5.4.1 Looking for 1-day Rug Pulls

We leverage our datasets to identify rug pulls systematically. Since we saw a considerable number of 1-day tokens and most of them are created serially, we narrow our investigation to the 332,265 in BSC and 25,180 in Ethereum 1-day tokens with a liquidity pool. Given the duration of these operations, we will refer to them as *1-day rug pulls.* We analyze all the Events emitted by the liquidity pools, looking for all the pools that emitted only one Mint and one Burn event in which the address that performs the transaction burns at least 99% of the minted LP-tokens (we don't use 100% since a small fraction of tokens might be stuck in the wallet due to rounding).

Estimating the gains of the operations

The simple operation, where the attacker does not swap in his liquidity pool, can be carried out by performing just four transactions: A transaction that creates the token, one that creates the liquidity pool, one to add the liquidity, and finally, the last transaction to remove the liquidity. These transactions can be performed individually, or they can be aggregated by leveraging a smart contract. Of course, we consider both cases when computing the fees. If the attacker performs swaps on the liquidity pool, we also consider the transaction fees paid for each swap.

To perform our estimation we use the following formula:

$$base\_gain = \delta_B - fees \tag{5.1}$$

$$net\_gain = base\_gain - T_{in} + T_{out} - fees_{swap} \tag{5.2}$$

The formula can be split into two components. The first part computes the gain in the case of the simple operation.The second formula takes into account the improved version of the operation, where the creator of the liquidity pool manipulates it by performing swaps operations. In this case, we remove from the gain T_{in}, that is the amount of tokens that the manipulator artificially adds to the liquidity. We also add to the gain T_{out}, the quantity of tokens that the manipulator removes from the liquidity pool before the final removal of the liquidity (T_{out}). Finally, we remove from the gain the fees used to perform the swap operations (fee_{swap}).

5.4.2 Results

After processing our data, we discover that an incredibly high number of liquidity pools are actually rug pulls. In BSC, 272,349 out of 332,265 (81.2%) of the considered liquidity pools have a rug pull pattern, while 21,742 out of 25,180 (86.3%) in Ethereum. This result shows that attackers use most of the 1-day tokens as disposable to carry out rug pulls.

These operations are arranged by 116,516 different addresses in BSC and 16,539 different addresses in Ethereum. As we can expect from the previous analyses, most of the token spammers that operate in BSC are linked to this kind of activity. Indeed, in BSC, 2,112 out of 4,231 (50%) token spammers performed at least one rug pull. Instead, in Ethereum, there are only 45 token spammers (0.3%) that have been involved in this activity. We find 115 addresses that perform more than 100 rug pulls in BSC, accounting for 19.1% of the operations, with the most active performing 16,102 operations. Instead, in Ethereum, we find only one address performing more than 100. Interestingly, combining the information in the BSC and Ethereum dataset, we find a token spammer that operated on both blockchains with the same address [3]. He performs five rug pulls on Ethereum and three on BSC.

Looking at the liquidity pools, we find that BNB (97.8% of the cases) is the token paired the most with the 1-day token. It is followed by USDT (0.67%) and BUSD (0.15%), two stablecoins pegged to the USD. Instead, Wrapped Ether is paired with all the 1-day tokens in almost all the liquidity pools with a rug pull in Ethereum. As the next step, we want to estimate the number of users that fall prey to such activities. To do so, we exclude the addresses that swap into liquidity pools they have created themselves from this analysis. We collect 251,250 different addresses in BSC and 57,552 in Ethereum that interact with at least one liquidity pool with a rug pull pattern. These addresses performed 2,903,022 swaps on the considered liquidity pools in BSC and 317,257 in Ethereum.

[3] 0x87605612492c74bA0037fFaef676c0f3f6958918

We divide the swaps into buy (1-day token) and sell operations. As we can expect, given the anatomy of the 1-day rug pull, we find that most of the operations are buy operations. More in detail, in BSC 2,286,056 (78.7%) are buy operations and 616,966 (21.3%) sell operations. In Ethereum, we find a very similar pattern, with 254,061 (80.1%) buy operations and 63,196 (19.9%) sell operations.

As final metric, we compute the average value of the swaps performed by the users. The average amount of swaps is almost identical for buy and sell operations on both the blockchains, with 0.01 BNB for BSC and 0.19 ETH for Ethereum's liquidity pools. Interestingly, we notice a considerable difference in the average swap amount between the two blockchains. Indeed, the average swap is approximately $3 on BSC and $360 on Ethereum.

The gains

Before computing the gains of the attackers, we calculate the average price an attacker has to invest to arrange the operation. If the attacker does not perform any swap into the liquidity pool, the cost of the operation is on average 0.03 BNB in the case of BSC and 0.2 ETH for the Ethereum blockchain. Thus, the investment needed to perform such operations is low, even if it could vary substantially when the blockchains are overloaded. For instance, we found some rug pulls that reached the cost of 1.1 BNB or even 3.3 ETH. The base cost to arrange the operation is interesting because it represents a bound to the loss the attackers have to afford for each operation.

We leverage our datasets to compute the gain of the operation using the formula 5.1 described in Sec. 5.4.1. We describe the 266,340 operations on BSC and the 21,594 on Ethereum in terms of successful and unsuccessful operations based on the operation's net gain. In particular, we consider an operation successful if the net gain is strictly positive.

Successful operations. Among the liquidity pools with a rug pull pattern, there are 104,404 (39.1%) operations in BSC and 13,368 (61.9%) in Ethereum closed with a profit for the attacker. A possible reason for the higher success rate of the rug pull on Ethereum could be that, as we saw, on average, users tend to invest more money. Indeed, on average, attracting only one investor is enough to cover the operation's cost. To investigate what can affect the gains, we combine information on gains with those of the manipulations. When the creator of the liquidity pool does not perform any kind of manipulation, the net gain is, on average 0.11 BNB in BSC and 1.34 ETH in Ethereum. Operations carried out on liquidity pools that suffer wash-trading activity have an average gain of 0.25 BNB in BSC and 12 ETH in Ethereum, which is considerably higher than the previous case. Instead, we notice a negligible increase in gains in the case of pump operations with respect to the gains obtained by the liquidity pools without manipulation. Moreover, we find that both kinds of manipulation have no impact on the success rate. This show that operations that have wash trading are generally more profitable. However, the attacker has to perform several swaps, increasing its cost and loss in case of an unsuccessful operation.

Unsuccessful operations. There are 161,936 (60.9%) liquidity pools in BSC and 8,226 (38.1%) in Ethereum, for which the attacker does not cover the transaction

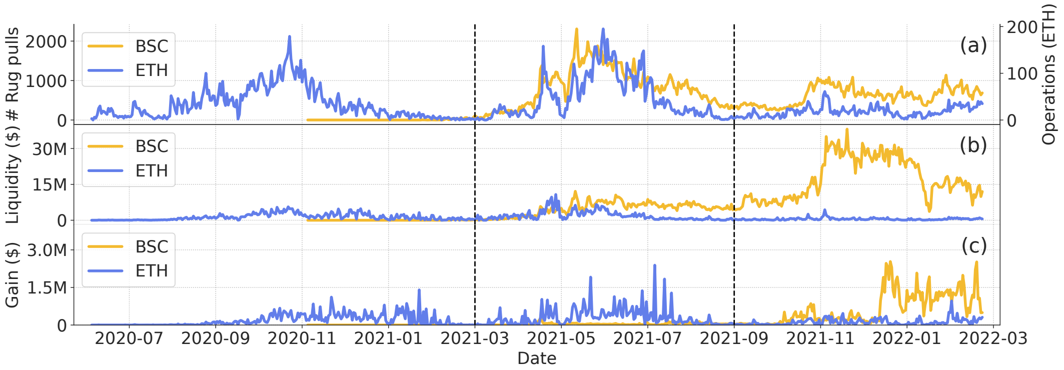


Figure 5.4. The figure shows the number of rug pull operations (a), the initial liquidity added to each pool (b), and the gain for each operation over time. All the metrics are aggregated daily. The dashed vertical lines divide the three phases we identify.

fees with the operations. For the 14% (21,122) of these liquidity pools of BSC and the 20% (1,506) of Ethereum, we notice that the operations were unsuccessful because nobody swapped into the liquidity pools. Considering the results we obtained, we can conjecture that the aim of the attackers is not to be successful every time but to arrange rug pulls serially and take profit in the long run. Indeed, the loss of an unsuccessful operation is minimal, and a streak of operations closed in loss can be covered with a single profitable operation.

Financial cost of 1-day rug pulls and comparison with other blockchain phenomena. In our study, we find that the number of 1-day rug pulls (21,594) and attackers (16,439) in Ethereum is significantly lower than in BSC (266,340 operations carried out by 117,110 rug pullers). Nonetheless, the total gain of Ethereum operations, around $150 million, is remarkably higher than the gains of BSC operations, that amount to $91 million. Moreover, the same trend holds when considering the volume of rug pull operations, which we define as the total value of BNB and ETH swapped. Here we find that Ethereum has a volume of $772.5 million against the $243.5 million of BSC. To gain insight into the magnitude of 1-day rug pull operations, we compare our metrics with popular blockchain shenanigans, like MEV and front-running. Tab. 5.5 reports more relevant metrics collected from related works about operations carried out in Ethereum. As we can see, 1-day rug pull is the second type of operation by profit, generating slightly lower gains than Sandwich Attacks ($174.34 million in accordance with Qin et al. [304]). Particularly interesting is the number of addresses that perform the operations. Indeed, in Ethereum, the number of attackers that performed 1-day rug pulls is almost five times the number of the Sandwich Attackers (the fraud with the higher number of attackers in our comparison). We believe the operations are performed by a large number of addresses due to their ease of execution. The reported numbers highlight that 1-day rug pulls is a significant phenomenon in the DeFi ecosystems that involve hundreds of thousands of malicious actors and move more than 1 billion USD.

Table 5.5. Comparison with other blockchain operations. We report only on frauds performed on Ethereum since our work is the first to analyze the Binance Smart Chain.

Operation	Tot Gain ($)	# Addresses	# Operations	Blockchain	From	To
[153] Counterfeit Tokens	17.35M	364	573	Ethereum	2015-07-30	2020-03-18
[356] Displacement	4.1M	74	2,983	Ethereum	2015-07-30	2020-11-21
[304] Fixed Spread Liquidations	89.18M	2,724	31,057	Ethereum	2018-12-01	2021-08-05
[357] Honeypots	90K	53	690	Ethereum	2015-08-07	2018-10-12
[356] Insertion	13.9M	1,975	196,691	Ethereum	2015-07-30	2020-11-21
[304] Sandwich Attacks	174.34M	3,488	750,529	Ethereum	2018-12-01	2021-08-05
[86] Smart Ponzi	17.70M	444	835	Ethereum	2015-08-01	2020-05-20
[356] Suppression	1.03M	128	50	Ethereum	2015-07-30	2020-11-21
1-day rug pulls	148.93M	16,439	21,594	Ethereum	2015-07-30	2022-03-07
1-day rug pulls	90.78M	117,110	266,340	BSC	2020-04-20	2022-03-07

A longitudinal view

Fig. 5.4 provides a longitudinal view of the daily number of rug pull operations (Fig. 5.4 a), the liquidity added (Fig. 5.4 b) and the gains (Fig. 5.4 c). Analyzing the trends of the chart, we identify three different phases, divided by the black dashed lines in the figure. In the first phase, we find the first spike of 1-day rug pulls in Ethereum. In the second phase, rug pulls start to increase in the BSC. However, the Ethereum gains are generally higher for the same invested liquidity. Finally, in the third phase, we see that BSC surpasses Ethereum in terms of liquidity added, number of operations, and gains. In the following, we describe in detail the three phases:

Phase 1: DeFi Summer. The first phase took place approximately from June 2020 to March 2021. At the beginning of this phase, we see an increase in the daily number of rug pull operations in Ethereum, with a peak of 179 daily operations in October 2020. Then, the number of operations steadily decrease until March 2021. We believe that the increase in the number of operations was bootstrapped by a phenomenon known in the crypto-community as DeFi Summer 2020 [252]. During this period, DeFi became extremely popular, and, as a result, the market capitalization and prices of several tokens soared [317]. This interest in DeFi attracted new users looking for investment opportunities, which may have triggered the increase in rug pull operations. Fig. 5.4 (b) shows that there is a significant amount of liquidity invested in these operations, on average $37,941 (44 ETH), with an average gain of $5,969 (5.65 ETH) (Fig. 5.4). Note that this phase involves only the Ethereum blockchain because the BSC was released in September 2020 and was not very popular yet.

Phase 2: Altcoin season. Fig. 5.4 (a) shows a second spike in the number of rug pulls from March 2021 to September 2021. In this case, the spike involves both Ethereum and the BSC, which reach a maximum peak of 195 and 2,309 daily operations. It is interesting to notice that the number of operations over time follows the same trend for Ethereum and BSC. For this reason, we believe an exogenous event caused this spike. Analyzing the events of that period, we believe this rise in the number of operations may be a so-called *Altcoin Season.* An Altcoin Season is a period in which Altcoins[4] perform better than Bitcoin, significantly increasing their value. Previous study [223] shows that an Alt Season is marked by a drop of an indicator called *Bitcoin dominance.* This indicator measures the ratio between the market capitalization of Bitcoin to the total market capitalization of the entire cryptocurrency market. According to Coinmarketcap [104], in this period, the Bitcoin dominance decreased from 69% of January 2021 to 39% in May 2021. This market phase is frequently characterized by "Fear of missing out" (FOMO) [49], which makes investors more inclined to buy riskier tokens. For this reason, we believe investors have flocked to AMM markets to buy tokens, and rug pull operations skyrocketed. Fig. 5.4 (b) shows that the liquidity invested in these operations is higher for Ethereum, with an average of $39,625 (50 ETH) against the $5,624 (15 BNB) of BSC. Operations in Ethereum are also way more profitable, with an average gain of $5,836 (6.3 ETH) against the $48.4 (0.12 BNB) of BSC operations.

[4] Altcoins [184] is a combination of the two words "alternative" and "coin". The term is used to indicate all cryptocurrencies except Bitcoin.

Table 5.6. Token names most frequently used in 1-day rug pull operations.

BNB Smart Chain		Ethereum	
Name	# of tokens	Name	# of tokens
Pornhub	1,023	Hyve.works	50
Galaxy	588	Deriswap	32
Seedswap	502	Shibaswap	28
Lionswap	429	Apple core finance	17
Eco.finance	421	X20.finance	16
Spacex	419	Yield farm rice	15
Onlyfans	419	The sandbox	14

Phase 3: The overtaking of the BSC. The last phase goes from October 2021 to March 2022. In this phase, we find an interesting twist, as BSC surpasses Ethereum in terms of liquidity added and gains of rug pull operations. Indeed, in this phase, rug pulls in BSC have significantly more liquidity invested than in the past (56.9 BNB on average vs. 15.3 BNB of the previous phase) and higher gains (2.26 BNB on average vs. 0.12 BNB of the previous phase). For this reason, we can see in Fig. 5.4 that the total daily invested liquidity and gains in BSC are significantly higher than Ethereum and reached more than one million USD. In Sec. 5.8, we explore some possible reasons for this increase.

Tokens' names

To further deepen our analysis of rug pulls, we focus on the names used in the operations. Analyzing the rug pulls, we notice several tokens with the same name in BSC and Ethereum. We find that of the 272,349 tokens involved in the operations in BSC and 21,742 in Ethereum there are only 157,864 (57.9%) and 18,801 (86.4%) unique names. Thus, we attempt to cluster the 1-day tokens into categories and enumerate them. Table 5.6 shows the most used names and the number of occurrences for each of them.

As a first category, we explore clones—tokens with the same name as an existing (and more popular) cryptocurrency. To systematically search for these cases, we use as an authoritative source the CoinGecko APIs [100]. Leveraging them, we retrieve the names and the addresses of all tokens created and verified with the indexer service on the BSC and Ethereum. At the end of the process, we build a list of 5,325 tokens for BSC, and 5,172 tokens for Ethereum. We complement this list by adding popular variations for some tokens' names (*e.g.,* we also considered ADA as a possible name for the Cardano token). Using our list, we discover 22,002 cloned tokens in BSC and 1,781 in Ethereum. The most cloned tokens in BSC are Berryswap (370), Shiba Inu (191), and SafeMoon (158).

The second category we explore is the one of tokens that attempt to impersonate companies or websites. In this case, to obtain a list of possible target companies, we retrieve the name of the companies of the Standard and Poor's 500 (S&P 500) stock market index. Instead, for the websites, we extract from the Alexa ranking [5]

[5] Data retrieved 2022-04-26

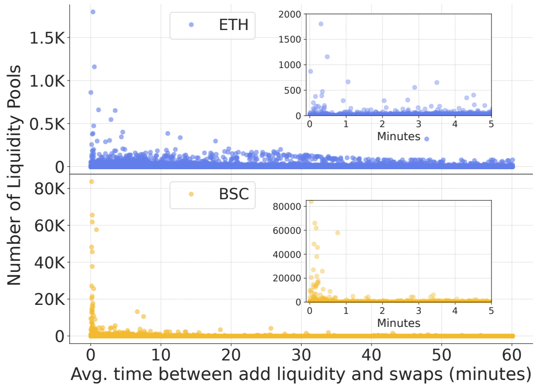


Figure 5.5. Each data point represents an address that swaps inside liquidity pools with a rug pull pattern. On the *y axis*, we represent the number of different liquidity pools where the address swaps. On the *x axis* we show the average time interval between the first time the liquidity is added to the liquidity pool and the swaps operations of the address.

the name of the top-ranked 200 websites. Using in conjunction these two lists, we find 4,638 tokens of this category in BSC and only 95 in Ethereum. The companies and websites that are present the most are Pornhub (1,023), Spacex(419), Onlyfans (398), Oracle (319), and Amazon (270).

We find several names that contain popular meme-related words like "Doge", "Inu" or "Shiba". This is not surprising, since meme tokens are very popular after the events that involved the "meme stocks" of GameStop (GME) and AMC Entertainment (AMC) in late 2020 [233]. Luckily, CoinMarketCap and CoinGecko offer a categorization of the tokens that also contain the "meme" category. We leverage these lists to extract the most frequent words and search for them into the tokens involved in rug pulls. We find a huge amount of tokens of this category: 54,229 in BSC and 4,835 in Ethereum.

As the last category of our investigation, we look for DeFi services (*e.g.,* Deriswap, Shibaswap, and Eco Finance). In this case, we simply search for tokens containing the "swap", "defi" and "finance" keywords. With this approach, we find for this category 25,524 tokens in BSC and 3,751 in Ethereum.

With our simple categorization, we covered the names of 39% of the 1-day tokens on the BSC and 48% on Ethereum. Even if we were not able to categorize all the tokens, we get some insights on how attackers pick the name to arrange their operations. In particular, we note a strong trend in choosing tokens' names related to the meme category and leveraging the name of popular cryptocurrencies, services, and companies.

5.5 Sniper Bots 2.0

We find that a large fraction of rug pulls are successful, even if they are zero-effort operations, without fake tokens or wash trading. Since these kinds of operations are very quick and simple, it is still unclear how they can be profitable. We analyze the operations carried out inside rug pulls more in-depth and discover that their success may be due to the activity of a particular class of trading bots, called Sniper Bots.

Sniper bots are automated bots that monitor time-bound activities and perform an action before or after anyone else. An example of sniper bot are “Scalping Bots”, bots that monitor the availability of target products from a website and buy them as soon as they are available (*e.g.,* Nvidia GPUs) [68].

With the birth of and the widespread adoption of AMMs, a new kind of sniper bot has been developed, which we define *Sniper Bots 2.0.* These kinds of sniper bots are programs that buy tokens on liquidity pools as soon as they are listed. To do so in the fastest way, sniper bots can leverage the mempool— the list of transactions not yet inserted in blockchain blocks. We find examples of these bots distributed for free on Github [359, 319, 116] and for a price at several other websites [13, 302]. Analyzing the code, we can infer how they work. As a first step, the sniper bot must search for newly listed tokens. The fastest implementation scans the mempool looking for transactions whose byte-code indicates that they are adding liquidity to a brand new liquidity pool. Another possibility is that the sniper bot waits for the token to be listed on services like BscScan or Etherscan. Then, the bot sends a swap transaction to buy the token, and if the gas price is properly adjusted, it is executed in the same block (but immediately after) of the transaction that adds the liquidity. Sniper bots typically execute only the buy operation. The user then can freely decide when to sell the token and make a profit. However, we also found some variants that automatically sell the token when the price reaches a pre-defined goal.

5.5.1 Identifying Sniper Bots

We conjecture that one of the reasons for the profitability of rug pulls operations are sniper bots that buy tokens from every liquidity pool indiscriminately. Thus, we can consider the liquidity pools involved in rug pulls as “honey pots” to detect sniper bots. To verify our intuition, we focus on addresses that swapped inside liquidity pools with a rug pull Fig. 5.5 shows the phenomenon: Every dot is an address, and its position indicates the number of different liquidity pools where the address swapped and the average delay from the pool creation. The figure shows a few addresses that swap in thousand of liquidity pools almost immediately after their creation. Since these addresses perform these operations serially and incredibly fast, we believe they must be sniper bots. We set up two conservative thresholds to identify evidence of addresses used by sniper bots.

For BSC, we consider all the addresses that swap on average with a delay smaller than five blocks (15 seconds) and that swap in at least 100 different liquidity pools. We flag 130 addresses as possible sniper bots. These addresses represent only 0.03% of all the addresses that swap inside liquidity pools involved in rug pulls. What is impressive is that they swap in 235,777 liquidity pools, representing 68.7% of all the liquidity pools with a rug pull. Moreover, these addresses also perform an impressive number of swaps: 2,691,173, that account for 24% of all the swaps performed in liquidity pools with a rug pull. We find that 31% of these swaps are performed in the same block where the liquidity is added for the first time in the liquidity pool. In these cases, we can confirm that the sniper bots scanned the mempool to swap in the same block where the liquidity is added. However, we also find sniper bots that perform the swap operations a few blocks after the liquidity is created.

We find sniper bots to be less present in Ethereum. Also, in this case, we pick two thresholds and consider all the addresses that swap on average with a distance lower than three blocks (45 seconds) and that swap in at least 10 liquidity pools. We find 64 possible sniper bots that swap in 30% of all the liquidity pools and perform a much smaller fraction of swaps with respect to BSC sniper bots (3.5% of the total). However, interestingly, a higher percentage of swaps are performed in the same block where the liquidity is added in the liquidity pools (60%).

5.6 1-day Rug Pull Mitigation

Our study highlights that the 1-day rug pulls have some distinctive features. In the following, we propose some metrics that stem from the lessons learned from our analysis that may be useful to build a detection system.

- **Token lifetime:** This metric measures the time that elapses since the creation of the token. Indeed, we find that 1-day rug pull operations are performed in a very short timeframe (§5.4.1).

- **Distribution of the liquidity:** This metric tracks the distribution of the LP-tokens. In 1-day rug pulls, the liquidity pool creator owns all the liquidity (§5.4). Thus, it should be considered extremely risky when a single address owns most of the liquidity.

- **Address rug pull records:** This metric tracks addresses that performed a rug pull operation to add them to a list of potential malicious addresses. Indeed, we find that some addresses perform rug pulls multiple times. (§5.3).

- **Deceptive token name:** This metric measures the similarity between the name of tokens contained in the liquidity pools and popular existing tokens or companies. We find that attackers often deceive investors by exploiting the name of the token. (§5.4.2).

An attacker aware of these metrics can try to evade the detection by putting more effort into carrying out the operations (*e.g.,* using different addresses or creating the token in advance). Nonetheless, a distinctive characteristic of 1-day rug pulls is that they are easy to execute and require low effort by the attacker. Thus, we believe the proposed metrics could be sufficient to discourage this operation. Moreover, new metrics and more sophisticated techniques can be developed to identify attackers trying to circumvent the detection. For example, it is possible to follow the money flow between addresses associating different addresses to the same attacker.

We believe that AMMs are interested in leveraging the proposed metrics to build a detection system. Indeed, some have already put effort into this direction. For instance, PancakeSwap recently included in its interface a service called HashDit [188], which provides a risk level in investing in a liquidity pool. HashDit is a Token Contract Scanning service, that estimates the risk of a token by analyzing the code of its smart contract [58]. We believe the proposed metrics can enhance this and other existing services by adding insightful information.

5.7 Related Work

Tokens identification. In previous work, there are mainly two token identification approaches: behavior-based and interface-based. The behavior-based method assumes that a token contract maps addresses to the number of tokens owned and contains a function to transfer tokens. Chen et al. [85] follow this approach, analyzing the EVM execution path to find smart contracts data structures that indicate the bookkeeping of a token. The interface-based approach, the technique we take in this work, aims to find tokens that conform to specific interfaces (*e.g.,* the ERC20 interface). This method involves discovering the implemented functions within the smart contract bytecode. Several works use this approach [122, 374, 88]. Frowis et al. [148] proved that the interface-based technique could detect 99% of the tokens in their ground truth dataset.

Liquidity pool scams. Xia et al. [390] characterize scam tokens on Ethereum. First, they leverage CoinMarketCap [104] to obtain a ground truth of official and scam tokens. They used The Graph [177] to obtain 21,778 tokens and 25,131 liquidity pools from May 2020 to December 2020. A guilt-by-association heuristic is adopted to enlarge the dataset, subsequently used to train a machine learning model. More than 11,182 fraudulent tokens were discovered after they ran their classifier on the expanded dataset. Mazorra et al. [253] extended Xia et al. dataset by including Uniswap data until 3 September 2021, discovering an additional 18 thousand scam tokens. They provide three categories for rug pulls: simple, sale, and trap-door. Then, they found that more than 97.7% of the tokens labeled as scams are involved in rug pulls.

Rug pull mitigation. Rug pulls are a very recent issue, and to the best of our knowledge there is no actual solution to prevent them. However, there is a new proposed standard and some protocols that can help to mitigate the problem. To counter the theft of tokens, Wang et al. [381] proposed a new token standard called ERC-20R. With this standard, a transaction is reversible for a short time (dispute period) after it has been performed. During this period, the sender can request to freeze the disputed asset to a set of decentralized judges. If judges agree to lock the disputed asset, it starts another period of time in which the sender can convince judges to revert the transaction. Instead, liquidity locker protocol (*e.g.,* Unicrypt [363]) allows locking LP-tokens inside smart contracts for a given amount of time. This solution assures that the liquidity cannot be removed from the pool until the timer expires, making rug pull impossible. Of course, this solution does not prevent rug pulls after the time expires or dumping one of the tokens in the liquidity pool.

5.8 Discussion

What is the impact of not collecting all the internal transactions? Unlike other works [374, 88], we do not collect all smart contracts generated by internal transactions. We collect smart contracts created directly by EOAs, and expand our dataset by adding contracts that emitted at least one Transfer Event. This approach could lead to the loss of a small percentage of tokens. We can perform a rough estimation of the ERC-20 token we miss by comparing the number of tokens we retrieved with the number of tokens retrieved by Chen et al. [88] at the same block height.

Our approach retrieves 146,928 tokens instead of 165,955, approximately 12% less. However, it is important to note that, by design, our approach misses only tokens that are never used, traded, or transferred. So, the missing tokens do not represent interesting cases for our study.

Why does it appear that rug pulls and token spammers are more frequent in BSC than in Ethereum? From a technical point of view, rug pulls work the same way in the two blockchains. Indeed, since BSC is EVM compliant and PancakeSwap is a fork of Uniswap, the same smart contract can be used on both blockchains. However, the cost of the operation is significantly different. As we saw in Sec. 5.4.1, performing a rug pull in BSC is cheaper (on average $10.5 with peaks of $600) than in Ethereum (on average $400 with peaks of over $2,000). These costs represent a fixed cost for the attacker, and going even or gaining money may be more difficult in Ethereum versus BSC.

Are cost-efficient blockchains vulnerable to 1-day rug pulls? As discussed, one of the possible reasons for the prevalence of rug pulls on BSC is the low transaction cost. This could suggest that cost-efficient blockchains are more vulnerable to 1-day rug pulls. However, to confirm this hypothesis, it is necessary to examine whether the phenomenon is common in blockchains with costs similar to the BSC.

Considering our case study of BSC, we believe the low cost of transactions is not the only reason for the high number of rug pulls. In particular, BSC provides one of the first DeFi ecosystems that is cheaper and faster than Ethereum. It quickly became very popular. Moreover, thanks to EVM compatibility, many no-code tools, libraries, and smart contracts already developed for Ethereum can also be used on BSC. This allows the deployment of smart contracts and the creation of new tokens with limited technical capabilities. Thus, the high number of potential victims, the little technical challenge, and the cost-efficiency made the BSC fertile ground for malicious actors to carry out 1-day rug pulls. Even though the low cost can facilitate rug pulls, increasing the costs of blockchains is not a real solution. Instead, a possibility is to shift the focus to DEXes's protocol and smart contracts for token creation. In particular, it could be possible to design more secure smart contracts to handle tokens (*e.g.,* ERC-20R) or AMM protocols with policies that disincentive rug pull operations.

Can different users coordinate to carry out the same operation, or can a user use multiple addresses? In this work, we considered each address belonging to a single different user, and we assumed there is no coordination among addresses. Nonetheless, a user may change the address he uses to perform each rug pull. It is also possible that a group of users coordinate to carry out the operation. For example, a user can create a liquidity pool while others perform wash trading. A possible approach to detect this malicious behavior is to gather all the transactions among the allegedly involved addresses and look for malicious patterns or communities (*e.g.,* using graph analysis). In this work, we do not perform this analysis, but we plan to explore more sophisticated rug pulls as an extension of this work.

Are 1-day rug pulls frauds? 1-day rug pulls are very different from more notorious rug pulls like Squid Game [297] or Luna Yield [103]. Indeed, these operations lasted weeks or months, and their perpetrator exploited extensive marketing campaigns and misleading advertising to deceive users into investing in their tokens. In the case of Squid Game, the scammer created a token in the BSC following the success of the homonym Netflix television series [326].

Due to the extensive marketing campaign promoting the token as official on social media platforms such as Twitter and Telegram, its value skyrocketed from a few cents to over $2,856 in less than a week [106]. Then, the scammer removed nearly all of the liquidity from the pool ($3.3 million), causing the token's value to plummet to near zero [297]. In our chapter, we study 1-day operations that aim to make a profit with the least possible effort in a short time frame. For this reason, it is unlikely that they leverage sophisticated marketing campaigns to lure investors, like in the case of Squid Game. However, some 1-day rug pull operations use other kinds of deceptive tactics. The first uses token names identical or slightly different from well-known companies or popular tokens. As we saw in Sec. 5.4.2, this case involves 8.7% of Ethereum rug pulls, and 10% of BSC rug pulls.

Another deceptive technique consists in attempting to legitimate the project by verifying the smart contract code on BSCscan and Etherscan. The verification consists in uploading the source code so that the platform can compile it and verify that it matches the bytecode of the token stored in the blockchain. The verification provides users transparency and gives more guarantee that the token is not fraudulent. We find the smart contract is available and verified for 55% (147,069) of BSC and 67% (14,722) of Ethereum tokens involved in the 1-day rug pull operations. Finally, another technique to legitimate the token consists in creating the "official" Telegram group of the token. We find evidence of this technique in the smart contract's code and then inspect the groups on Telegram. Indeed, analyzing the source codes, we notice that 19,096 token smart contracts in BSC and 1,334 in Ethereum report a link to the Telegram group of the token. Although the organizers of 1-day rug pulls use deceptive techniques to dupe investors, we cannot consider these operations frauds because the phenomenon is still not regulated. In any case, people lose money: Investors bought the token in 92.7% of the Ethereum rug pulls and in 91.2% of the BSC ones, and in all these cases the investment is lost. For this reason, we believe that these operations, even if not illegal, are exploitative of the DeFi ecosystem and should be contrasted to safeguard investors. Indeed, regulators are starting to take action to contrast them. For example, New York State Senator Kevin Thomas proposes criminalizing rug pulls and other crypto frauds by introducing a new bill amendment request (Senate Bill S8839) [327]. The idea of the bill is to introduce the crime of *illegal rug pull* that occurs if the creator of the token sells more than 10% of his tokens within five years of their last sale.

5.9 Ethical considerations

In this chapter, we examined 3 billion transactions from Ethereum and the BSC. We focused our research on the addresses that create tokens and how they use them. All data we retrieved is publicly available, and EOA addresses are pseudo-anonymous. We never attempted to deanonymize the addresses or violate their privacy during this work. Consequently, and in accordance with our IRB's policies, we did not require express approval to conduct our analysis.

5.10 Conclusion

In this work, we conduct a thorough investigation of the tokens and the liquidity pools of the BNB Smart Chain and Ethereum. We studied the lifetime of the tokens and their creators. We discovered two very interesting metrics: 60% of the total tokens of both blockchains do not survive their first day (1-day token), and a tiny fraction of addresses (1% of addresses), which we called token spammers, created more than 20% of the tokens. We explore the correlation between token spammers and 1-day tokens, and we found that token spammers strongly impact the existence of 1-day tokens. More interestingly, we find that token spammers use 1-day tokens as disposable tokens to arrange rug pulls, exploiting the mechanism of liquidity pools. We selected from our dataset all the liquidity pools that show evidence of a rug pull and dissect the operations, analyzing them from several perspectives. Finally, we introduce the sniper bot, trading bot that aims to buy tokens at their listing price. However, they unwillingly became victims of the rug pulls because of their mechanism. As future work, we believe it is interesting to further refine our results by including addresses that cooperate to perpetrate rug pulls in the analysis. It could be possible to uncover other malicious and more sophisticated patterns. As discussed in Sec. 5.8, cost-efficient blockchains could be more exposed to the 1-day rug pulls. Thus, it is interesting to extend our analysis to blockchains with transaction costs comparable to BSC (*e.g.,* Algorand [164]).

Chapter 6

Ready, Aim, Snipe! Analysis of Sniper Bots and their Impact on the DeFi Ecosystem

The cryptocurrency market is renowned for its high volatility [232], with cycles where the value of tokens can skyrocket and plummet [171]. This behavior is prevalent among tokens with small market capitalization, especially those that are newly listed. The rapid increase in value often triggers FOMO [49] (fear of missing out) among investors, who often purchase tokens based on hype rather than their intrinsic value. With the emergence of DeFi (Decentralized Finance) and specifically Automated Market Makers (AMMs) [392]—platforms where trading is powered by smart contracts—every blockchain user can list and make their tokens tradable. As a result, hundreds of tokens are listed on AMMs daily [390], and finding the next token worth investing in can be a challenging task. This has created the ideal environment for the emergence of sniper bots—automated systems designed to buy tokens quickly as soon as they are listed on an AMM platform.

In this work, we leverage open-source implementations of sniper bots to gain insight into their features. We find that sniper bots implementations are more sophisticated than we might expect. Indeed, some of them offer features such as protection against fraudulent liquidity pools (*e.g.,* honeypots and rug pulls), as well as anti-bots mechanisms that are commonly implemented in token smart contracts. Then, we build the liquidity pools dataset, consisting of Ethereum and BSC liquidity pools and their activities. Inspired by what we learned analyzing the implementation of the sniper bots, we devised a straightforward approach to detect addresses that take advantage of sniper bots. We discover that the sniper bots phenomenon is more widespread on BSC than in Ethereum. However, after analysis of the operations conducted by these addresses, we surprisingly find that Ethereum operations have a higher likelihood of being closed with a profit, despite requiring a larger investment. Finally, we leverage Etherscan and BSCScan, two popular explorers for Ethereum and BSC, respectively, to download the source code of the smart contracts of the tokens contained in the liquidity pools dataset. We search among the retrieved smart contracts implementation of anti-bot mechanisms that can limit the action of the sniper bots. In line with our previous findings, we discover that developers of BSC token smart contracts are more active in countering bots' activities, implementing more mechanisms to hinder their actions.

Our main contributions are:

- **Analysis of sniper bots**: To the best of our knowledge, we are the first to conduct an in-depth analysis of sniper bots and their implementation. We explore each phase of a sniping operation, from the choice of the target liquidity pool to the sale of the token. For each phase, we report in detail the different techniques implemented by the most popular open-source sniper bots on GitHub.
- **The impact of sniper bots**: We propose an identification methodology for addresses that take advantage of sniper bots serially. We find 161 addresses on Ethereum 819 addresses in BSC. Analyzing the operations of the identified sniper bots, we note that sniper bot users behave differently accordingly to the platform they operate. We analyze their operations to estimate their success rate and their profit. We quantify their impact on the ecosystem of liquidity pools, finding that they move a volume of 11360.7 ETH on Ethereum and 45606.3 BNB in BSC.
- **Smart contract analysis**: We describe the most popular mechanisms to counter bots used by smart contract developers. Then, we quantify the adoption of anti-bot mechanisms by tokens, leveraging a dataset of almost 600,000 smart contracts. We observe that 17.9% token smart contracts on Ethereum and 37.36% on BSC implement at least one mechanism to hinder the action of bots.

This work was accepted at the Companion Proceedings of the ACM Web Conference 2023 (CAAW 2023). In this project, I worked with my supervisor, Professor Alessandro Mei, Professor Massimo La Morgia, and the Ph.D. students Alberto Mongardini and Federico Cernera from the Sapienza University of Rome.

6.1 Background

6.2 Sniper Bots

Sniper bots are software applications that monitor a specific activity to automatically perform an action before anyone else [80]. Examples of these bots are "Scalping bots," [68] programs designed to purchase limited-availability goods quickly. These kinds of bots have been used to buy limited-edition sneakers [262] and Nvidia GPUs during the 2021 graphic card shortage [68]. The goal of bots' users is usually to resell the purchased items at a higher price. In the blockchain world, sniper bots are typically used to buy tokens as soon as they are listed on an AMM platform.

6.2.1 Sniper bots dataset

To understand how sniper bots are implemented and the features they offer, we leverage Github [167], one of the most popular Internet hosting services for software.

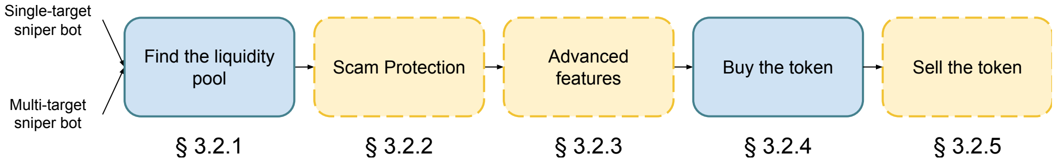


Figure 6.1. The phases of a sniper bot's execution. We report in blue the phases we always find implemented by sniper bots. Instead, we report in yellow the optional phases that a sniper bot can implement to improve its usability.

We systematically search sniper bots on Github using keywords such as: "Sniper bots," "Sniping bots," and other similar variations. This research yielded hundreds of open-source repositories that are impractical to analyze manually. Therefore, we decide to focus on sniper bots that have some popularity. To do so, we leverage GitHub's star ranking system [63] as a metric to infer popular repositories. We decided to analyze only sniper bots with at least 15 stars. Using this criterion, we select 70 sniper bots. Then, we discard from our analysis 25 repositories containing the code of sniper bots unrelated to AMMs. Most of these sniper bots are used in online video games (8 repositories) and to buy NFTs as quickly as possible in NFT marketplaces (5 repositories). Analyzing the remaining repositories, we notice that 17 of them do not contain open-source code. In these cases, the repositories are used to promote and sell closed-source sniper bots. Some others contain only executable files and instructions to use the bots. In the end, considering only the open-source implementations, we focus on analyzing 28 repositories.

6.2.2 The anatomy of sniper bots

Analyzing the sniper bots' source codes, we first notice that almost all the considered sniper bots target Ethereum or the BNB Smart Chain (BSC). The only exception is a sniper bot that operates on the Avalanche [313] blockchain. In particular, 17 sniper bots exclusively support the BSC, three support Ethereum, and seven offer multi-chain support, being able to target both BSC and Ethereum. Looking more in detail at the implementations, we find these bots target the PancakeSwap and Uniswap AMMs. Only a few of them also offer the possibility to snipe tokens released on other AMMs operating on the Ethereum and BSC blockchains.

Analyzing the code repositories, we find that there are two categories of sniper bots:

- **Single-target sniper bots.** These sniper bots target a specific token, requiring the user to input the smart contract address of the token. A user can use this kind of sniper bot to buy the token of a highly hyped project at its listing price, expecting its value to skyrocket right after. We find 25 implementations of this kind of sniper bot.

- **Multi-targets sniper bots.** The second category of sniper bots is designed to buy every token as soon as it is listed. In this case, the goal of their users is more speculative. Indeed, the strategy behind using these bots is to buy as many different tokens as possible, hoping that at least a few of them will gain value in the future. In our dataset, we find 3 implementations of multi-targets sniper bots.

Despite having different goals, these two categories operate similarly and follow the same execution phases, which we illustrate in Fig. 6.1 and report in the following.

1. **Find the liquidity pool.** As a first step, the sniper bot must identify the liquidity pool from where to buy the target token. Since these kinds of bots aim to buy the token as soon as it is listed, the sniper bot looks for newly created liquidity pools that are available for trading (*i.e.,* actually containing liquidity). In § 6.2.2, we will explore the strategies the sniper bots implement to handle this phase.

2. **Scam protection.** Then, the sniper bot can perform some checks to ensure that the token to buy is not a scam. If these checks fail, the sniper bot will not buy the token, and in the case of a multi-target sniper bot, it will search for a new liquidity pool. This kind of security measure is implemented only by 13 sniper bots. We will explore the different implemented scam check solutions in § 6.2.2.

3. **Advanced features.** As we will discuss in §6.4, some token smart contracts implement techniques to avoid bot interactions. Thus, the sniper bot can implement workarounds to evade detection and still buy the token. This is an advanced feature that we find implemented in 10 cases. We will describe the anti-bot mechanisms in § 6.2.2.

4. **Buy the token.** Finally, the sniper bot buys the desired amount of the token. This phase can be performed by interacting with the AMM router, performing a simple swap operation, or interacting with a custom smart contract. We will explain these two techniques in § 6.2.2.

5. **Sell the token.** The sniper bot can also offer the possibility to sell the token automatically. In § 6.2.2, we describe how this phase is implemented by the 10 sniper bots that offer this feature.

Find the liquidity pool.

Exploring the source code of the sniper bots in our dataset, we find that they use different methodologies to discover new liquidity pools. In particular, we find that single-target sniper bots use the following techniques:
Mempool scan. The fastest way to find the liquidity pool containing the target token as soon as it is available is to leverage the blockchain mempool – the list of pending transactions waiting to be included in the next blockchain blocks. In this case, the sniper bot monitors the mempool, searching for the first transaction that adds liquidity to the target token's liquidity pool. Technically, this is done by checking if the bytecode of the transaction contains the signature of the *addLiquidity* function of the Uniswap router (*i.e.,* the function that is used to add liquidity to a liquidity pool).

Leverage Uniswap smart contracts. These kind of sniper bots directly interacts with the smart contracts of the AMM. In particular, it calls the *getPair* function of Uniswap's Factory contract at regular time intervals. This function takes a pair of token addresses as input and returns as output the address of the liquidity pool that contains the pair, if it exists, or the zero address if it does not. Thus, the sniper bot can use *getPair* providing as input the token to snipe and the valuable token they want to buy the token with (*e.g.,* ETH). This method is slower than the previous one. Indeed, in order for the Uniswap smart contract to be updated, the transaction that updates its status must be confirmed in the blockchain.

However, even if the liquidity pool exists, there is no guarantee that it contains liquidity. Indeed, a user can create a liquidity pool but not add tokens to it, making any kind of swap impossible. Thus, once identified the liquidity pool, to understand if there is liquidity, the sniper bot performs polling requests to the *getReserves* function of the contract. This function returns the quantity of the two tokens in the pool. When this quantity becomes different than zero, the liquidity has been added, and the sniper bot can perform the swap. Instead, we find that multi-target sniper bots usually follow one of these approaches:

Event Log monitoring. Sniper bots monitor the blockchain Event Log looking for new *PairCreated* events. As mentioned in the Background Section 2.2.4, this Event is emitted by the Factory contract of Uniswap each time a new liquidity pool is created. From the data in this Event, the sniper bot can retrieve the addresses of the two tokens in the liquidity pool and the address of the liquidity pool itself. As for the previous case, the sniper bot must verify that the liquidity pool actually contains the tokens. Thus, before sniping the liquidity pool, it ensures that there is liquidity through the *getReserves* function.

Telegram channels. Some sniper bots use Telegram [352], a very popular messaging app with more than 700 million active users, as a source to discover new liquidity pools. Indeed, on Telegram, there are many channels—public groups where only the admin can write [230]—dedicated to token release announcements. These sniper bots use Telegram APIs to monitor a list of channels. The sniper bot parses newly-published messages of these channels, looking for the address of a liquidity pool created on the target AMM. For instance, we find a sniper bot that monitors Telegram channels [350, 5] reporting newly-listed tokens by the CoinMarketCap [104] and CoinGecko [102], two of the most popular cryptocurrency aggregator websites. Usually, the list of monitored channels is customizable by the user, which can add or remove specific channels. Additionally, users can specify a list of token addresses or words blocklisted to avoid buying specific tokens or tokens including in their name specific words.

Scam protection

We find that sniper bots often perform checks to avoid buying scams or suspicious tokens. This is not surprising, as anecdotal evidence (*e.g.,* SquidGame [297]) and previous works [80, 253, 390], have shown that investing in liquidity pools can be risky as thousand of tokens are purposely created to perform scams. The sniper bots' countermeasures are mainly designed to prevent two threats: rug pulls [390, 80] and honeypot tokens [357]. We find that sniper bots employ the following solutions to avoid these threats:

Trial trade. A possible countermeasure to avoid falling prey to honeypots is to perform a trial trade. With this practice, the sniper bot buys a small number of tokens and right after sells them. The goal of this practice is to check that the token smart contract does not prevent the sale of the token. Thus, if the trial trade is successful, the sniper bot purchases the desired token amount.

RugDoc. A second possibility is leveraging the API of RugDoc [318], a tool designed to help DeFi investors to make informed decisions about the tokens they choose to invest in. RugDoc performs some tests on the token to check if it is a honeypot and provides results through APIs. So, the sniper bot queries the RugDoc's APIs to retrieve the tests' results and infer the level of risk of the target token. If the estimated level of risk is acceptable, the sniper bot will proceed with buying the token.

Source code check. Before buying the token, some sniper bots check the source code of the smart contract. In particular, they only buy tokens whose smart contract is public and verified on popular blockchain explorers like Etherscan (for Ethereum) or BSCScan (for BSC). These websites offer contract verification where developers can publish their smart contract source code on the site. The site will then compile the code and check if the generated bytecode matches the stored bytecode on the blockchain. If it matches, the contract is considered verified. Other than the verified status, we find sniper bots that avoid buying the token if the smart contract contains specific keywords.

Liquidity check. Lastly, some sniper bots offer the feature to buy only in liquidity pools with more than a certain amount of liquidity. To perform this check, snipers bots call the *getReserves* function of the liquidity pool's smart contract.

Advanced features

Some sniper bots offer advanced features to circumvent smart contract functionalities designed to directly or indirectly limit the action of sniper bots. Indeed, as we will see in Sec. 6.4, several token smart contracts implement techniques to hinder sniper bots or bots in general (*e.g.,* trading bots).

Wait n-blocks. This feature enables the user to specify the number of blocks the sniper bot waits to purchase after the liquidity is added. This precaution is to avoid penalties imposed by some token smart contracts that want to penalize automatic trading actions at the early stages of the liquidity pool. For instance, a smart contract may blocklist addresses that buy the token too quickly, prohibiting subsequent token transfers from the blocklisted addresses. Others may impose a very high fee on purchase transactions (*e.g.,* 99% of the acquired token returns to the liquidity pool) executed on the first blocks the liquidity is added.

Check trading enabled. Some token smart contracts implement the possibility to enable and disable the transfer of the token at will. The token creator can use this functionality for different technical or marketing reasons. To handle this case, some sniper bots implement a procedure to infer when a token enables the transfer functionality as soon as possible. The sniper bot sends a small transaction. If the transaction succeeds, the bot performs a second transaction and buys the intended amount of tokens. Otherwise, we find two different approaches implemented by the

sniper bots in our dataset: In the first, the sniper bot starts to poll the liquidity pool's smart contract monitoring the token's price. If the price oscillates, the sniper bot infers that the transfer is enabled and attempts to buy the token. Instead, with the second approach, the sniper bot monitors the mempool looking for a transaction that contains the bytecode of commonly known functions used to enable the transfer of the token, such as: *openTrade*, *enableTrading*, *tradingStatus*.
Multiple buys. There are smart contracts that restrict the number of tokens an address can buy in the same transaction. This feature prevents big players—also known as *whales*— from buying a large token supply in a short amount of time. Even if not intended to contrast sniper bots directly, this mechanism can cause the sniper bots' buy transactions to fail if the desired quantity of tokens overcomes the restriction of the smart contract. Some sniper bots offer the possibility to buy the desired amount of tokens using multiple buy transactions, working around the smart contract limitation.

Buy the token

Finally, the sniper bot buys the token. In particular, we find two ways the sniper bots perform the purchase:
Interacting with the Router contract. The sniper bot can buy the token by sending a transaction to the Router contract of the target AMM. To finalize the purchase, the user of the sniper bot has to specify the number of tokens to buy and the maximum slippage (*i.e.,* the difference between the expected and the actual price) tolerated.
Using a custom smart contract. The sniper bot buys the token by sending a transaction to a custom smart contract rather than directly to the AMM router. This approach incurs higher costs, including smart contract deployment fees, but provides advantages. Indeed, the smart contract enables atomic execution of multiple operations, such as checking if the token is a honeypot.

Sell the token

While all the sniper bots provide an automatic way to buy tokens, not all of them offer the feature to sell them automatically. Indeed, we find that the selling functionalities are present only in 10 out of 28 sniper bots.
Sell percent gain. The sniper bots that automatically sell tokens allow the user to set a target profit percentage. Once the token's value increases by the designated percentage, the sniper bot automatically sends a swap transaction to sell the token.
Stop loss. Most sniper bots also provide a mechanism to protect investors from excessive loss, namely a *stop loss*. The stop loss is a simple threshold and allows the bot to sell the tokens if the token price drops below a specified percentage relative to the buy price.
Trailing stop. Some sniper bots implement a more sophisticated trading strategy called the Trailing Stop. With the Trailing Stop, the sniper bot continuously tracks the token's price. If the maximum value of the token falls below a given percentage, the sniper bot automatically sells the token.

6.3 Sniper bots detection

In the previous section, we focused on understanding how sniper bots work by analyzing their source code. In this section, we change perspective, investigating how they are operatively used by analyzing blockchain data.

6.3.1 Liquidity pools dataset

To study the sniper bots, we create the liquidity pools dataset, a collection of liquidity pools and their operations in Ethereum and BSC. To retrieve the data, we run an Ethereum and a BNB Smart Chain node on our machine and synchronize the two blockchains. Then, we use Web3 [296], a Python library that allows interaction with EVM-compliant nodes to query the blockchains and obtain the data from their inception to March 2022. To collect the data, we use the same approach of previous works [390, 253, 80]. In particular, we parse the Event Logs of both blockchains, collecting Events compliant with the Uniswap smart contract implementation. Note that all Uniswap forks, including those deployed in the BSC, also implement these Events. In detail, we retrieve the data of the following events: PairCreated, Mint, and Swap.

- **PairCreated:** With this Event, we collect the addresses of liquidity pools and other relevant data: the addresses of the two tokens they contain, their block of creation, the transaction hash, and the address that created the pool. We find 70,656 liquidity pools on Ethereum and 972,467 on BSC, which contain in their pairs 61,507 unique tokens in Ethereum and 840,862 unique tokens in BSC.

- **Mint:** By collecting Mint events, we infer when liquidity providers added liquidity to the pool. From the Event, we collect the address that added the liquidity, the amount of liquidity added, the address of the pool, the transaction hash, and the block where the operation occurred. We collect 2,359,333 Mint events in Ethereum and 26,972,440 Mint Event in BSC.

- **Swap:** Gathering Swap events, we obtain information such as the transaction hash, the block in which the operation occurs, the address that performs the swap, the address of the liquidity pool, the number of tokens swapped, the gas used, and the gas price. We collect 82,430,138 Swap events in ETH and 749,188,792 Swap events in BSC.

6.3.2 Sniper bots identification

As a first step towards understanding how sniper bots are operatively used, we have to identify them. Although sniper bots can target any liquidity pool pair, we focus on sniper bots that target liquidity pools containing the native coin of the blockchain (BNB or ETH), which are 86.5% and 91.3% of the liquidity pools on Ethereum and BSC, respectively. Narrowing our research on these liquidity pools allows us to easily define two operations: the buy and the sell. In particular, we define as a buy operation any swap that takes as input ETH (BNB) and provides as output any

Table 6.1. Summary of sniper bots operations and their profits.

Metric	Ethereum	BSC
# Liquidity pools	55,678	710,515
# Sniper bots	161	819
# Operations	14,029	1,395,042
Avg. buy	0.75 ETH	0.03 BNB
Avg. gain	0.84 ETH	0.08 BNB
Success rate	25.6%	7.0%

other ERC-20 (BEP-20) token. Conversely, we define as a sell operation any swap that takes an ERC-20 (BEP-20) token as input and provides as output ETH (BNB). Furthermore, considering the speculative nature of sniper bots, it is reasonable to assume that a user would never snipe a liquidity pool he created. Thus, we remove from our dataset all the buy and sell operations performed in the liquidity pool created by the same address performing the swap (3,201,920 swaps).

As we saw in the previous subsections, sniper bots are developed to perform buy operations immediately after the liquidity is added to the liquidity pool. However, in some cases, they can not always buy the token in the same block the liquidity is added, but they have to wait for some blocks to be sure they do not fall prey to scams or high taxes (see Sec.6.2.2). Even if it is difficult, a standard user could swap into a new liquidity pool a few blocks after it has been created. Thus, to avoid this case, we focus only on addresses that serially take advantage of sniper bots. Moreover, in our identification process, we have to consider that the user can operate with the same address for sniping tokens but also for his regular trading activities. Thus, some sniper bots' addresses could have operations carried out far from the creation of the liquidity pool. With these considerations, we outline two conservative thresholds to identify sniper bots' addresses by looking at their activities:

- At least 90% of the address buy operations have to be performed into 5 blocks from the block in which the liquidity was added for the first time to the liquidity pool.
- The address has to perform a buy operation in at least 5 different liquidity pools.

In the following subsection, we analyze the addresses selected by applying these two thresholds.

6.3.3 Results

Fig. 6.2 shows a scatterplot where each address is represented by a dot (blue for Ethereum addresses and yellow for BSC addresses). The *y-axis* displays the number of liquidity pools the address has traded in, and the *x-axis* shows the 90th percentile of the time intervals in blocks between the first addition of liquidity to the pool and the address's buy operations. Both figures contain a zoom of the first 30 blocks. We leverage the Mint events in our liquidity pools dataset to calculate the time

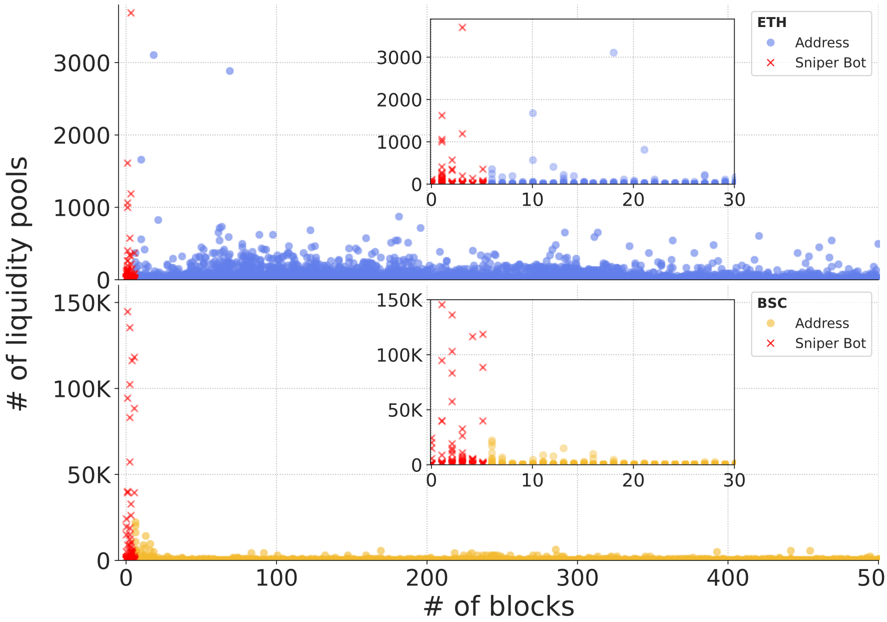


Figure 6.2. The scatter plot shows on the y-axis the number of liquidity pools where each address performed buy operations. On the x-axis, we plot the blocks elapsed between the buy operations and the first time liquidity is added to the pool.

elapsed from the buy operation and when the liquidity is added for the first time. We indicate with red crosses the addresses selected using our thresholds. As we can see, these addresses perform buy operations extremely close to the first liquidity addition and in hundreds of liquidity pools, exhibiting a pattern highly compatible with sniper bots' operations. For the remained sections, we will refer to these addresses as "sniper bots". Analyzing them, we discover that:

Sniper bots are more prevalent in the BSC. Using our thresholds, we select 161 addresses on Ethereum, and 819 addresses on BSC, performing 15,052 buy operations and 1,440,945 operations, respectively. The total Ethereum and BSC liquidity pools targeted are, respectively, 7,879 and 198,786. To confirm that these addresses are sniper bots, we quantify how many performed a buy operation in the same block where the liquidity is added to the pool for the first time. This operation is virtually impossible to perform by a human, as it requires monitoring the mempool. We find that 144 (89.4%) addresses on Ethereum and 512 (62.5%) on BSC perform at least a buy operation at the same block of the first liquidity addition.

Sniper bots use different strategies in Ethereum and BSC. Ethereum sniper bots perform, on average, fewer operations than BSC sniper bots (93 vs. 1,759). However, they tend to invest higher sums than BSC sniper bots, with an average of 0.75 ETH ($673) against 0.03 BNB ($13). These different behaviors are arguably dictated by the different costs of fees on the two blockchains. Indeed, computing the fee spent to buy tokens by snipers bots, we find that, on average, they spent 0.019 ETH ($23.1) on Ethereum while 0.001 BNB ($0.46) on BSC.

Sniper bots have a relevant economic impact. Summing up the buy operations, we observe that sniper bots have a significant economic impact. These bots invest

11,360.7 ETH ($10,144,808) in Ethereum and 45,606.3 BNB ($18,720,447) in the BSC.

6.3.4 Gains

In this section, we analyze in detail the operations performed by sniper bots to estimate their profitability. For each sniper bot, we aggregate all the buy and sell operations performed on a token in a single *sniping operations*. Indeed, as mentioned in Sec. 6.2.2, sniper bots can buy or sell a token using multiple transactions. After this aggregation, we find 14,029 sniping operations in Ethereum and 1,395,042 sniping operations in BSC. For each sniping operation, we estimate its profit using the following formula:

$$balance = T_{out} - T_{in} - fees \tag{6.1}$$

Where T_{out} is the profit obtained by the sell operations, T_{in} is the amount spent to buy the token, and $fees$ is the transaction fees paid for buy and sell operations. In the following, we divide the operations into successful and unsuccessful, considering an operation successful if the *balance* is strictly positive.

Successful operations. Interestingly, we find that in BSC only 96,809 (7.0%) of the sniping operations are successful. The success rate is better on Ethereum, with 3,571 operations (25.6%). Moreover, we find that the average gain of Ethereum (0.84 ETH) is higher than the average BSC gains (0.08 BNB). Even if sniping operations are unsuccessful on average, we find some extreme cases of profit indicating that sniping tokens can be a high-risk, high-reward strategy. In particular, we find an address[1] that performs a sniping operation with a profit of 299.8 ETH. The address buys 1.86M TrustSwap [105] tokens paying 90 ETH (0.00004 ETH for each token), exactly in the same block when the liquidity is added to its liquidity pool (block 10426750). The sniper bot sells 1M of TrustSwap tokens 23 blocks after the buy, with a price increase of 600% (0.00024 ETH). Then it sells the remaining tokens for a similar price in 3 subsequent transactions for a total of 390 ETH. If we subtract the initial investment of 90 ETH and the transaction fees, the address profits 299.8 ETH from the operation.

Unsuccessful operations. Most of the sniping operations are unsuccessful. Indeed, 10,458 (74.5%) Ethereum operations and 1,298,233 (93.0%) BSC operations are unsuccessful. We notice that almost all BSC operations (85.5%) and a large fraction (48.8%) of Ethereum sniping operations are unsuccessful because the sniper bots did not sell the token. Possibly, these addresses did not sell the token because they could not do so. Indeed, Cernera et al. [80] show that almost 60% of BSC and Ethereum liquidity pools have a rug pull in the first day of their life. Thus, it is possible that the sniper bots did not sell the tokens before all the liquidity was removed from the pool. In the cases where the sniper bots sell the tokens, the loss is generally not too high, with 0.11 ETH ($108) in Ethereum and 0.01 BNB ($4.1) in BSC. Tab. 6.1 resumes our findings about sniper bots and their profits.

[1] 0xc0c5c6ea185b331ffc97499fb6bf7c1f1a0fc48c

6.4 Anti-bot mechanisms

In this section, we analyze the source code of smart contracts to understand how many tokens implement mechanisms that can directly or indirectly limit the action of sniper bots. As mentioned in Sec. 6.2.2, Etherscan and BSCScan offer the possibility to upload on their website the source code of a smart contract to verify it. Thus, to build the smart contracts dataset, we query the APIs [219, 220] of the two explorers to retrieve the smart contracts source code of the tokens contained in the liquidity pools dataset. At the end of the process, we are able to retrieve 47,619 out of 61,507 (77.42%) verified smart contracts source codes for Ethereum and 545,048 out of 840,862 (64.82%) for the BSC tokens.

6.4.1 Smart contract analysis

Since it is not feasible to manually analyze the code of all the retrieved smart contracts, we search on the Internet for reference implementations of anti-bot measures. In particular, we search for these implementations in sector forums (*e.g.,* OpenZeppelin [282], Ethereum StackExchanges [337]), tools for automated token creation (*e.g.,* Tokensbygen [354], Cointool [107]), or querying Google with keywords such as: *smart contract anti-bot measures*, *anti-bot protection*, *sniper bot countermeasures*, *token sniper bot protection*. Following our research, we find six different mechanisms that can hinder the action of sniper bots and 34 reference implementations. Next, we create a regular expression for each implementation that we can use to automatically identify similar snippets of code in our smart contracts dataset.

In Tab. 6.2, we describe the implementations for each mechanism and how we identify the token smart contracts adopting it. Moreover, we publicly release the regular expressions we used in [345].

In the following, we briefly describe the six different mechanisms and report the number of smart contracts adopting them.

Disabled trading. This mechanism allows to enable or disable the transfer of the tokens, and hence the trading, at will. As we discuss in Sec. 6.2.2, when a liquidity pool has the trading disabled at its first blocks of life, sniper bots must implement advanced features to be successful in their operations. In our dataset, we find that the smart contracts implementing this mechanism are 4,584 (9.62%) on Ethereum, and 15,170 (2.78%) on the BNB Smart Chain.

Tax during the launch window. With this mechanism, the smart contract imposes a high tax on each token transaction (*e.g.,* 99%) during the launch window of the liquidity pool. Sniper bots can avoid falling prey to this mechanism using the advanced feature *Wait n-blocks* (see in Sec. 6.2.2). We identify 9 (0.018%) and 15,540 (2.85%) token smart contracts on the Ethereum and BNB Smart Chain, respectively, implementing this technique. In particular, more than 88% of these smart contracts impose the tax only for the first two blocks from the token launch, while the remaining smart contracts define a different number of blocks, either with a fixed number or through a variable.

Token amount limit. This mechanism consists in limiting the number of tokens per transaction and/or per address that can be purchased during the early stage of the

Table 6.2. Implementation of anti-sniper bot mechanisms.

Mechanism	Description of the implementation
Disabled trading	This strategy involves managing the trading status for a token using a boolean variable, commonly called *tradingOpen*, that is initially set to false. Only the smart contract owner can change its status to true to enable trading. We search for token smart contracts having a method (such as *tradingStatus*, *openTrading*) to set a variable that is checked in the Transfer method and that, if set to false, does not allow the token trading.
Tax during the launch window	This solution aims to penalize addresses trading too fast for a human by temporarily increasing the fee to 99% for blocks close to the token launch. We search for token smart contracts defining a function (typically called *getTotalFee*) that checks whether the block of the transaction is greater than the block of the token launch plus a certain threshold and, if not, raises the fees.
Token amount limit	This solution restricts the number of tokens that can be purchased during the launch phase. We search for token smart contracts that, in the Transfer function, check the amount of tokens to transfer and if this is greater than a certain variable (*e.g., _maxTxAmount*), revert the transaction. Some smart contracts perform this check with a specific function like *checkTxLimit*.
Transactions number limit	Some smart contracts check the number of transactions sent by an address in a given time window, setting a cooldown that blocks further transactions for that address until it expires. We look for token smart contracts implementing in the Transfer function a check that reverts the transaction if its block timestamp is lower or equal to the cooldown timer associated with the transaction recipient (*e.g., cooldownTimer[recipient]*). Some smart contracts define a function (*buyCooldown*) to set the variable managing the cooldown and its duration.
Gas price limit	Here the goal is to slow down bots setting a gas price limit and block transactions using a gas price higher than a certain threshold. We look for token smart contract defining functions, commonly called *setPriceLimit*, *setLimitsInEffetc*, or *setProtectionSettings*, to set a gas price limit.
Sniper bot blocklist	This strategy consists in blocking all the transactions sent by addresses already known for being sniper bots. We look for token smart contracts blocking the transaction if its sender belongs to the blocklist (*isSniper*). The list is updated with sniper bots' addresses buying the token at the same block of its launch.

Table 6.3. Smart contracts implementing anti-bot mechanisms.

	BSC	Ethereum
Disabled trading	15,170 (2.78%)	4,584 (9.62%)
Tax during the launch window	15,540 (2.85%)	9 (0.018%)
Token amount limit	189,465 (34.76%)	7,749 (16.27%)
Transactions number limit	13,018 (2.38%)	10 (0.02%)
Gas price limit	1,157 (0.21%)	143 (0.3%)
Sniper bots blocklist	464 (0.08%)	75 (0.15%)

liquidity pool. Although we find sniper bots successfully bypassing the transaction limit (Sec. 6.2.2), we have no evidence of sniper bots being able to evade the limit per address. We find 7,749 (16.27%) on Ethereum and 189,465 (34.76%) on the BSC smart contracts implementing the limit per transaction mechanism. In contrast, only 18 on Ethereum and 24,714 on the BSC implement the limit per transaction.

Transactions number limit over time. To solve the problem of multiple transactions used to circumvent the previous mechanism, some smart contracts do not permit multiple transfer operations requested by the same address in a given time window. In particular, we identify 10 (0.02%) and 13,018 (2.38%) token smart contracts adopt this mechanism on Ethereum and BSC, respectively.

Gas price limit. As shown in Sec. 6.2.2, a common practice used by sniper bots to ensure their transactions are executed as fast as possible is to use a gas price higher than those of other transactions at that moment. Thus, a strategy to slow them down is to set a gas price limit and block transactions using a gas price higher than a certain threshold. Using this approach, we find the token smart contracts implementing this strategy are 143 (0.3%) on Ethereum and 1,157 (0.21%) on the BSC.

Sniper bots blocklist. The last mechanism consists in blocking all the transactions sent by addresses already known for being sniper bots or that perform transactions in the first blocks of life of the liquidity pool. Overall, we find 464 (0.08%) token smart contracts on the BSC and 75 (0.15%) on Ethereum.

Tab. 6.3 summarizes the number of token smart contracts implementing the different mechanisms analyzed. As we can see, the strategy that limits the token amount that can be bought is the most popular one on both blockchains (16.27% on Ethereum and 34.76% on BSC). Interestingly, we find that the second most popular mechanism to limit the sniper bot actions on BSC (*Increased fees*) is implemented by only nine (less than 0.02%) smart contracts on Ethereum. Instead, the runner-up mechanism on Ethereum (*Disable trading*) is implemented by more than 9% of the smart contracts on Ethereum and only by 2.78% on BSC.

Looking at the number of mechanisms used by each token smart contract in our dataset, we find that usually, they do not implement any mechanism to limit the actions of the sniper bots. Indeed, there are 9% token smart contracts on Ethereum and 31.4% on BSC implementing only one mechanism and very few more than one. The maximum number of mechanisms adopted is four (*disabled trading*, *token amount limit*, *transactions number limit*, and *increased fees*), implemented by 1,024 token smart contracts, all running on the BSC. From our data, it appears that BSC token creators are more active in contrasting the action of the sniper bots with 37.36% of the smart contracts that implement at least a mechanism against the 17.9% on Ethereum. This is probably because, as we have seen in Sec. 6.3.3, the sniper bot phenomenon is more spread on the BSC ecosystem than on Ethereum.

6.5 Related Work

Several works study the presence of bots in the AMM market, with a particular focus on front-running bots that perform arbitrage or sandwich attacks. Daian et al. [114] investigated the behavior of front-running bots that exploit arbitrage opportunities by monitoring the mempool. The bots scan the mempool for large buy transactions that result in an overpriced token on a particular market. They then swiftly send a transaction to purchase underpriced assets on another market and sell them on the overpriced market, capitalizing on the big buy. Qin et al. [304] propose heuristics to identify arbitrage operations and quantify their impact on the market. They find that from 2018 to 2021 arbitrage bots obtained a profit of 277.02M USD. Zhou et al. [398] studied sandwich attacks. This kind of attack is performed using two transactions. The first is placed just before the target transaction (*i.e.,* front-run), and the second just after it (*i.e.,* back-run). This strategy allows making a profit when a significant buy is performed in the AMM. They find that on Uniswap, an attacker can obtain an average daily profit of $3,414. Instead, Qin et al. [304] study the sandwich attacks on a larger scale, taking into account several marketplaces on Ethereum, quantifying the profit obtained through sandwich attacks in 174.34M USD. Front-running bots have also been studied by Torres et al. [356], they analyze 11 million Ethereum blocks finding more than 200 thousand attacks with an accumulated profit of $18.41M.

Sniper bots have received little attention from the scientific community since they have been partially analyzed only by Cernera et al. [80]. The work analyzes blockchain data to identify rug pulls, finding 21,594 and 266,340 operations performed respectively in the AMM markets of Ethereum and the BSC. Then, they identify addresses that frequently fall prey to rug pull operations and classify them as sniper bots. With respect to their work, we perform a deep characterization of sniper bots and analysis of their implementation. Moreover, we quantify their presence outside rug pull operations and analyze their investment, gains, and success rate.

6.6 Limitations

In this work, we focus only on open-source implementations of sniper bots that we find on GitHub. However, during our investigation, we find also several closed-source implementations [97] and providers that offer "Sniper bot as a service" [334]. Thus, there may be sniper bots that offer more advanced features that we could not analyze. From the point of view of the sniper bots identification, we purposely focus on detecting addresses that perform sniping operations serially. However, it is also possible that some addresses use single-target sniper bots to perform only one operation or rotate the addresses they use. For these reasons, our work only shows a lower bound on the usage and impact of sniper bots on the DeFi ecosystem. Finally, in our investigation of the anti-bot mechanisms implemented by smart contracts, we rely on reference implementations, which we find disclosed on the web. Even if we added some flexibility using regexes, the same techniques could have been implemented in different ways that we did not cover. Thus, also in this case, our estimation of the diffusion of anti-bot mechanisms is only a lower bound.

6.7 Conclusion

This study provides a thorough analysis of the phenomenon of sniper bots operating on Ethereum and BSC. First, we analyzed how sniper bots work, defining the phases composing a sniping operation. Then, we identified sniper bots operating on AMMs compatible with Uniswap and its forks. We studied their behavior and quantified their economic impact on the DeFi ecosystems. Lastly, we described the anti-bot mechanisms implemented by smart contracts to limit sniper bots and estimated their adoption on Ethereum and BSC. As future work, it is interesting to investigate the reasons for the low success rate of sniper bots, especially on BSC. Another possible direction is to assess the impact of sniper bots on the listing price of the target token. Finally, extending our analysis to addresses that do not use sniper bots serially would be valuable for a more comprehensive understanding of the phenomenon.

Chapter 7

The Conspiracy Money Machine: Uncovering Telegram's Conspiracy Channels and their Profit Model

Conspiracy theories have been an integral part of human history, offering alternative interpretations for complex events [367]. The most common definition, that we adopt in our work, is that a conspiracy theory is a belief that an event or situation is the result of a secret plan made by powerful people [123]. A notorious example is the Flat Earth theory [98]. Despite centuries of scientific evidence proving the Earth's roundness, the theory continues to be discussed and promoted by several communities [264, 140]. Throughout history, several conspiracy theories have emerged on a wide range of topics, like the Moon Landing Hoax [137], JFK Assassination [387], Holocaust Denial [244], Elvis Presley's Faked Death [94], and 9/11 Conspiracy Theories [338].

Nowadays, with the advent of the Internet and social media, conspiracy theories have found new outlets to spread and gain traction [130, 290]. A notable example is the Pizzagate conspiracy theory, which originated and spread on online bulletin boards in 2016 [362]. In 2017, online forums acted as a catalyst for QAnon conspiracy theories [119], which alleged that a global cabal of malevolent elites was involved in heinous activities. The advent of the COVID-19 pandemic has sparked various online conspiracy theories, including claims that the virus is a bio-weapon for population control [198], and that 5G technology is somehow linked to the spread of the virus [21]. Finally, on January 6, 2021, a pro-Trump mob stormed the U.S. Capitol building, disrupting the certification of the 2020 presidential election results [266]. These incidents led the major social media platforms to implement content moderation to curb the dissemination of these theories [50]. In response, conspiracy theorists are flocking to less moderated platforms to freely share their views. Anecdotal evidence from various news sources [52, 376, 10] underscores Telegram, one of the most popular instant messaging applications, as one such platform. This is not surprising, as Telegram offers a permissive content policy and channels—virtual rooms where the admins can broadcast messages to large audiences.

In our work, we perform a large-scale study of Telegram to shed light on its ecosystem of conspiracy channels. We propose a novel approach to identify channels related to conspiracy theories by examining the URLs they share. In particular, we leverage previous scientific work on conspiracy theories to build the Conspiracy Resource Dataset, which contains a list of online resources (*e.g.,* YouTube videos, Reddit posts) linked to conspiracy theories. Then, we use the TGDataset, a public dataset of over 120,000 Telegram channels, to find channels sharing conspiracy-related URLs with their subscribers. Then, we utilize a community detection algorithm to analyze Telegram communities, finding that conspiracy-related channels are clustered in four specific communities. We characterize these communities by analyzing their language and most influential channels. We refer to the channels contained in these communities as the Conspiracy Channel Dataset. The analysis of the Conspiracy Channel Dataset highlighted the presence of channels actively seeking to profit from their subscriber. We characterize and quantify this phenomenon, focusing on three monetization strategies: affiliate programs, donations, and crowdfunding campaigns. Moreover, we find that conspiracy theorists exploit the lenient product policies of eBay, Teespring, and Etsy to promote questionable items to their subscribers, such as 5G shields, EMF stone protectors, and healing wands. Conversely, they exploit Amazon's tolerant book content policies to self-publish and profit from books claiming to "reveal the truth" about several topics. Then, we focus on analyzing donation and crowdfunding platforms. While we could not extract information about donation URLs, we find several insights about crowdfunding campaigns. Indeed, crowdfunding projects sponsored by conspiracy channels collected millions of dollars donated by over 900,000 backers. Moreover, analyzing the top-funded campaigns, we find they are linked to far-right support, COVID-19 restriction opposition, and truth-revealing documentaries against governments and powerful individuals. Finally, we also find fake charity campaigns that are outright scams. Our work makes the following contribution:

- **Conspiracy Datasets.** We release two datasets. The first one is the Conspiracy Resource Dataset, a collection of conspiracy-related web resources gathered through an extensive literature review. The second is the Conspiracy URLs Dataset, a list of 193,431 resolved unique URLs shared by conspiracy theory-related channels. We believe these two datasets can enable further studies on identifying and characterizing conspiracy communities on other platforms and determining their activities and ideologies.

- **Conspiracy Detection and Analysis.** We propose an approach to identify conspiracy communities on Telegram, finding four large communities comprising 17,806 channels. We characterize each community by analyzing their language and their most influential channels.

- **Conspiracy Monetization.** Finally, we identify the potential strategies that conspiracy theorists can employ to generate revenue from the subscribers of Telegram channels. We find that the most popular approaches involve affiliate programs, donations, and crowdfunding campaigns. We discover more than 132K URLs linked to donation platforms and 31K URLs related to crowdfunding campaigns. Quantifying the amount of money raised with projects sponsored by conspiracy channels, we discovered that they amassed a total of $90M donated by over 985K backers.

The work presented in this chapter is under revision at a top security conference. In this project, I worked with my supervisor Alessandro Mei, the professor Massimo La Morgia from Sapienza University of Rome; PhD students Alberto Mongardini and Vincenzo Imperati from Sapienza University of Rome.

7.1 Related work

Telegram has recently gained substantial attention, with several prior studies investigating questionable activities on the platform. Weerasinghe et al. [382] discovered organized Telegram groups known as "pods", where members artificially boost each other's Instagram account popularity. Other studies have examined the usage of Telegram channels to organize cryptocurrency market manipulations, such as pump and dump [233, 212] and Ponzi schemes [279]. Lastly, some researchers have focused on the misuse of Telegram by terrorist organizations, using the platform to disseminate propaganda and recruit new members [76, 393, 385]. To the best of our knowledge, the research on analyzing conspiracy theories on Telegram is very limited, as existing studies primarily focus on social media platforms. Here is a report on these studies.

Telegram. Hoseini et al. [192] examined 161 QAnon groups on Telegram, analyzing their toxicity and performing topic modeling to understand the QAnon narrative in multiple languages. Garry et al. [154] focuses on analyzing 35 QAnon Telegram channels, discovering that they spread disinformation messages to recruit new adepts. La Morgia et al. [230, 227] analyze over 120,000 Telegram channels, focusing on detecting fakes and clones. They discovered that these channels are used to lure users into conspiracy-related channels. In contrast to these works, we analyze Telegram's overall landscape of conspiracy-related channels without focusing on a specific one. Additionally, we propose a method to identify communities of conspiracy channels and analyze their profit model.

YouTube. Leidwich et al. [237] explore whether YouTube's recommendation algorithm promotes radicalization by guiding users to increasingly extreme content. They categorized 816 channels, including 79 conspiracy ones. Clark et al. [93] leverage the dataset of [237] to find YouTube communities. They create an embedding for the channels considering the channel's subscribers. Then, they leverage cluster algorithms to reveal the YouTube communities, finding QAnon and conspiracy-related ones. Ballard et al. [45] leverage the dataset of the previously mentioned works to investigate the monetization strategies of YouTube conspiracy channels. They find that these channels have a high prevalence of predatory or deceptive ads, are often demonetized, and use alternative income sources via third-party platforms.

Reddit. Phadke et al. [294] analyze conspiracy theories on Reddit [299], a popular news aggregation and discussion website structured into subreddits—dedicated forums that users can create to discuss specific topics [37]. They use a radicalization model (RECRO) to study the evolution of radicalization in users who join conspiracy theory subreddits. Analyzing 169 million contributions from 36,000 users, they identify and describe four types of engagement trends that show different behaviors.

Papasavva et al. [287] focus on the QAnon conspiracy, analyzing 4,949 "Q" messages (also known as "Q drops") collected from various aggregation sites. Then, they assess Q drop dissemination on Reddit, finding that these messages are still shared after the Reddit ban of the most popular QAnon subreddits. Engel et al. [131] identify 19 QAnon-related subreddits and study the submission of 13K users participating in them. They discover these users are active across several subreddits, often unrelated to QAnon. A further analysis of these users' submissions reveals that they post harmful content and links from low-quality sources. Phadke et al. [293] study 56 conspiracy communities on Reddit to determine the factors that drive users to join these communities. They build a ground truth of 60k users to develop a logistic regression model to predict if a Reddit user will eventually join conspiracy communities.

Voat. Voat is a Reddit clone that gained notoriety after Reddit banned the Pizzagate conspiracy theory subreddit in November 2016 and the QAnon-related subreddits in September 2018 [370]. Papasavva et al. [288] analyze over 150,000 posts from the largest QAnon forum on Voat. The researchers examine the entities most frequently referenced in these posts and the predominant topics of discussion, revealing an emphasis on Trump and US politics. Mekacher et al. [259] build an extensive dataset of over 2.3 million Voat submissions, covering the whole lifetime of the website. They analyze the users' activity, finding that some of the most active subverses focus on hate speech and conspiracy theories.

4chan/8kun. 4chan [53] is an image-based bulletin board featuring several boards to discuss various topics. It has been associated with conspiracy theories, with several mainstream news sources identifying the platform as the source of the PizzaGate conspiracy theory [346]. Papasavva et al. [289] analyze more than 3.5 million messages on 4chan from the */pol* (politically incorrect) board. They find slang terms that allude to antisemitic conspiracy theories portraying Jews as "malevolent puppet-masters" exerting control over media, financial institutions, and even governments [17, 18].

Strong evidence of conspiracy theories is also documented in the 8kun (aka 8-chan) website, a "free-speech-friendly" alternative to 4chan linked to white supremacism, racism, and hate crimes [186, 181]. Aliapoulios et al. [287] study QAnon on several platforms including 4chan and 8kun. They find that 8kun threads about QAnon are significantly larger than the ones discussed on 4chan.

7.2 Methodology

To uncover the structure of channels in Telegram that are related to conspiracy theories, we build a methodology consisting of three steps: First, data collection (including the introduction of a new dataset); second, detection of "conspiracy channels;" third, identification of communities of channels linked to conspiracy theories.

7.2.1 Data collection

We leverage two datasets. The first one is the TGDataset [229], the largest collection of public Telegram channels, with over 120,000 channels and 400 million messages. This dataset provides information about the channels (*e.g.,* their title, description, and creation date), all the messages sent with their timestamp and whether a message has been forwarded, and from which channel it originated. Then, we build a novel dataset, the Conspiracy Resources Dataset, that we describe in the following.

Conspiracy Resources Dataset

This dataset is a collection of conspiracy-related resources extracted from an extensive review of the previous works about conspiracy theories reported in Section 7.1. To construct this dataset, we focus on studies that provide explicit pointers to the sources they analyze, either within the manuscript or in dedicated repositories. In the following, we report for each platform the number of resources we find and the reference article:

- **YouTube.** We follow the approach of [45] and use two repositories of YouTube channels reported in [237, 93]. These repositories contain a list of 4,007 YouTube channels manually labeled as conspiracy-related. For each channel, we extract the complete list of their videos, resulting in a total of 1,973,439 video IDs.

- **Reddit.** We leverage the work of [287, 294, 131, 293] to collect a list of 92 subreddits identified as conspiracy-related.

- **Voat.** We consider 3 Voat subverses related to the QAnon conspiracy theory reported in [288, 259]. Indeed, these works highlight Voat as a popular Reddit alternative for conspiracy communities.

- **4chan/8kun.** We did not find any resource specifically related to conspiracy theories on 4chan from previous work. Indeed, the infamous */pol* discussion board is too general and discusses topics unrelated to conspiracy theories. Instead, we collect a list of eight boards related to QAnon on 8kun from the work in [287].

- **Websites.** From the work of Aliapoulios et al. [287], we collect six websites that are well-known aggregators of Q drops spread by the QAnon conspiracy. Then, we use OpenSources [6] (used also to study QAnon in [186]) to extract 122 website domains linked to conspiracy theories.

Although there are other works mentioning conspiracy theories in other platforms like Parler [24, 46] or Twitter [21], they do not publicly release their data-sets or provide URLs related to conspiracy theories that we can use in our study. Tab 7.1 reports all the resources we find and the related papers.

7.2.2 Conspiracy channels detection

We devise a methodology that combines the TGDataset and the Conspiracy Resources Dataset to find channels on Telegram related to conspiracy theories. The detection is performed in three steps: First, we extract and pre-process URLs from the TGDataset, then we perform the match with the resources found in the Conspiracy Resources Dataset, and finally, we use graph analysis to find clusters of conspiracy-related channels.

Table 7.1. Summary of the conspiracy-related resources we find in previous work and related URLs we extract from Telegram.

Paper	Type	# Resources	# URLs in Telegram
[237, 45, 93]	YouTube	4,007 channels	138,274
[287, 294, 131, 293]	Reddit	92 subreddits	17,889
[288, 259]	Voat	3 subverses	21
[287]	8kun	7 boards	3,787
[287, 186]	Web	128 websites	299,262

Data extraction and pre-processing

We parse all the messages (498,320,597) in the 120,979 TGDataset's channels and use regular expressions [83] to extract all the URLs. In this way, we obtain 205,046,775 URLs, 84,809,578 of which are unique. A first analysis of the URLs reveals that almost 20% (17,213,640 URLs) have been shortened using URL shortener services [33] such as *bit.ly* (2,378,987 occurrences) or *if.tt* (1,516,131 occurrences). Since our methodology for detecting conspiracy theories channels revolves around identifying URLs associated with conspiracies, we want to ensure we do not miss any of them because it has been shortened. Thus, we resolve the shorted URLs by sending a HEAD request to the shortening service using the Python Requests library [81] collecting the final URLs.

URLs matching

Then, we extract URLs associated with the resources collected in the Conspiracy Resources Dataset.

We start by using the conspiracy channels and videos' IDs in the Conspiracy Resources Dataset to search YouTube URLs related to conspiracy theories. We detect 2,446 URLs linking to conspiracy channels and an impressive number of 135,828 URLs (53,179 unique) linking to conspiracy videos. The most widely shared channel is *Fall Cabal*, which is now removed due to violations of YouTube's policy. We use AltCensored [27], to gather information about this channel, finding it had over 480,000 subscribers and an impressive 24 million views. The content posted by the channel reveals it is associated with the "Fall of the Cabal" [16], an antisemitic documentary used to recruit QAnon followers affiliated with Dutch conspiracy theorist Janet Ossebaard [256]. Instead, for Reddit, 8kun, and Voat, we extract all the URLs linking to conspiracy subreddits, boards, and subverses, respectively. We find 17,889 Reddit URLs (17,238 unique), 3,787 8kun URLs (2,542 unique), and 21 Voat URLs (11 unique). Most of the URLs we find are related to the *r/conspiracy* subreddit, the largest conspiracy theory discussion board on Reddit [294].

Finally, we extract all the URLs having the domain of the flagged websites, finding 299,262 URLs (120,463 unique) from 405 different domains. The website providing the most matches (103,210) is Zerohedge, a far-right news aggregator known for spreading conspiracy theories, particularly about COVID-19 [275, 276]. The second most popular is InfoWars, well-known for promoting conspiracy theories and fake news [91, 395]. We also detect over 10k URLs linking to the *qagg.news* website, a popular repository that stores the messages of the QAnon conspiracy theory [154, 287].

In total, we find 459,233 URLs (193,431 unique) posted by 11,618 Telegram channels. In the following we will refer to this dataset as **Conspiracy URLs Dataset**. Tab. 7.1 succinctly reports the number of URLs collected inside the Telegram channels divided by resource type.

Clustering conspiracy channels

The previous analysis shows 11,618 Telegram channels that posted at least one message with a link to a conspiracy-related resource. We study how and if these channels are connected to better understand the phenomenon. To do so, we follow the approach of [230] and build the Telegram forwarding graph for the whole dataset of channels. The forwarding graph is a graph $G = (V, E)$ in which nodes in G are channels and an edge $u \to v$ in E represents the presence in u of a message forwarded from channel v. Users of channel u can follow the forwarded message and reach channel v. Thus, edges represent the possible flows through channels of users following forwarded messages.

After the graph creation, we want to identify communities—subsets of nodes within the graph that are highly connected with respect to the rest of the graph [305]. In this specific case, a community is a subset of Telegram channels that consistently forward messages among themselves and rarely forward messages from channels of the other communities. To perform community detection, we use the Leiden algorithm [358] since it is widely adopted and has proven to be effective in identifying communities in social graphs (*e.g.,* Twitter [248, 62, 23]). We determine the number of communities that maximize the modularity [65], a metric that assesses the quality of a network's partitioning into communities. In particular, it compares the density of connections within communities to what would be expected by a random partition of the nodes. The metric ranges between 1 to -1, and we obtain a score of 0.78. With this approach, we identify 47 distinct communities of channels.

Then, we analyze how the conspiracy-related channels are distributed inside these communities. To illustrate this analysis, we report the scatterplot in Figure 7.1. The Figure shows a dot for each community, the number of channels in the community (x-axis), and the percentage of conspiracy channels (y-axis). It is evident that some communities (highlighted in red) stand out due to an unusually high concentration of conspiracy-related channels. In particular, one community has over 80% of potential conspiracy channels, while the other 3 have more than 40%. These four communities contain 17,806 channels, 15% of all the Telegram channels in the TGDataset. Furthermore, a vast fraction (77%) of the channels containing at least one link obtained from the Conspiracy Resource Dataset belong to one of these four communities. In the following, we will refer to all the channels in the four communities as Conspiracy Channel Dataset, while we will refer to all the links contained in these channels as **Extended Conspiracy URLs Dataset**. Since these communities seem the most relevant to thoroughly understand the diffusion of conspiracy theories on Telegram, in the following, we will focus on analyzing them in detail.

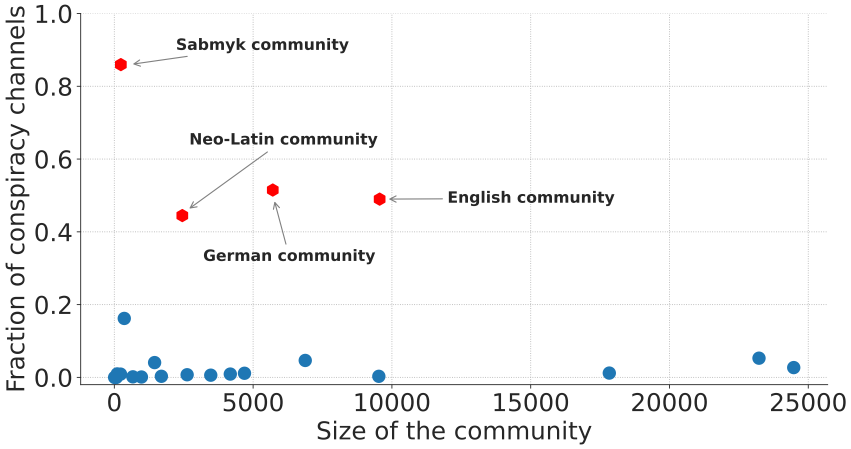


Figure 7.1. Each dot represents a community. The y-axis shows the percentage of conspiracy-related channels, and the x-axis represents the community size (number of channels). We highlight in red the communities that show an outstanding amount of conspiracy-related channels. We named the red communities considering their main language or, in the case of Sabmyk, the topic of discussion.

Table 7.2. Top channels by authority ranking in each conspiracy community.

HITS	English	German	Neo-Latin	Sabmyk
1	Disclose.tv	Fakten Frieden #FreeJanich	LA QUINTA COLUMNA TV	sabmyk
2	Tommy Robinson News	Eva Herman Offiziell	Noticias Rafapal	ChicagoReporter
3	RT News	Uncut-News.ch "Das Original"	El Investigador.org	GreatAwakening-Channel
4	Police frequency	Freie Medien	COVID-1984	CapitolNews
5	Covid Red Pills	#freejanich Oliver Janich öffentlich	DESPERTADOR DE LA MATRIX	NicolaTeslaNews

7.2.3 A look into the Conspiracy Channel Dataset

This section provides a bird-eye view of the discovered communities, focusing on their most influential channels. To find such channels, we adapt the well-established Hub and Authorities algorithm (HITS) [126] to our context. The algorithm was originally developed to identify relevant web pages in the World Wide Web graph [217]. The underlying idea is that a web page is a good hub if it links to many pages with a high authority score, and conversely, a web page is a good authority if it is linked by many good hubs. We adapt this idea to the Telegram graph, where we consider a channel authoritative if many channels with a high hub score forward its messages. Conversely, a channel is a good hub if it forwards messages from many highly authoritative channels. In our analysis, we are particularly interested in highly authoritative channels. Indeed, according to the definition, these channels are very influential in the community as their messages are widely forwarded. Therefore, we use the HITS algorithm to determine the channels with the highest authority score in each community and analyze the top five. Moreover, to better analyze each community, we also detect the language used within their channels using LangDetect [117], a port of Google's language-detection library that supports 55 languages. Finally, we provide a longitudinal analysis of these communities, analyzing the creation time of their channels over time. In the following, we will analyze separately each community.

English community. This is the largest community, with 9,480 channels and more than 20 million messages sent. More than 89% of these channels communicate using the English language. The most influential channel of this community is *Disclose.tv*, a website that discusses alternative viewpoints on the news and is notorious for propagating conspiracy theories [322]. Following closely in terms of authority ranking is *Tommy Robinson News*, a channel allegedly managed by Tommy Robinson, a British activist known to promote conspiracy theories, particularly those related to the threat of Islam for Western societies [95]. In particular, he created the English Defence League (EDL), and organization that promotes the theory that white European populations are being deliberately replaced with non-white Muslims through mass migration. *RT News* is a state-funded international media company headquartered in Russia, known for its alleged bias and for disseminating information that supports the Russian government's positions [109, 378]. About the *Police frequency* channel, by searching online, we did not find evidence that this channel is associated with well-known entities or individuals linked to conspiracies. However, looking at the content of its messages, we find that it is a far-right channel that focuses on American news about law enforcement, anti-gun control, and anti-immigration. Finally, *Covid Red Pills* claims to unveil the truth behind the COVID-19 pandemic.

German community. The German community comprises 5,663 channels that share more than 13.6 million messages. Over 94% of these channels communicate in German. Among the most influential channels, three of them, *Fakten Frieden #FreeJanich* , *Uncut-News.ch "Das Original"*, and *Freie Medien*, propose themselves as alternative media that share unmanipulated and free news, emphasizing their independence from government or political parties. Instead, the other two channels feature well-known German personalities. *Eva Herman Offiziell* claims to be the

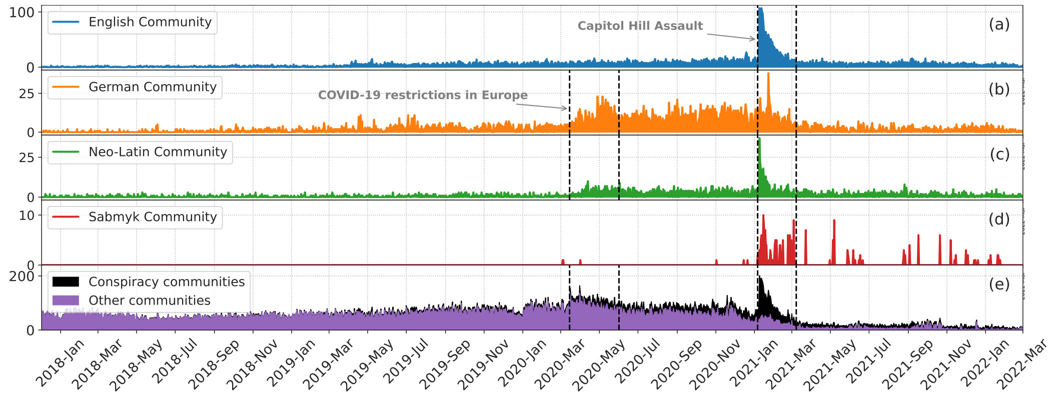


Figure 7.2. Channels created on Telegram over time.

official channel of Eva Herman, a former German news presenter recognized for promoting various conspiracy theories [149]. Finally, the last channel is about Oliver Janich, a German far-right conspiracy theorist and book author who gained notoriety for his writings on 9/11 conspiracy theories. He is also one of the most famous supporters of the QAnon conspiracy theory in Germany [372].

Neo-Latin community. In this case, the community is not predominantly monolingual with 58% of the channels primarily communicating in Spanish, 21% in Portuguese, and 16% in Italian. This community consists of 2,431 channels and has shared over 7.3 million messages. Similarly to the German community, the most influential channels promote themselves as alternative media, emphasizing their independence from government influence and advocating freedom of speech. Notably, three channels (*El Investigador.org* , *COVID-1984* and *DESPERTADOR DE LA MATRIX*) mostly focus on COVID-19 conspiracies, claiming that the virus is created in a laboratory, that the vaccine is used to reduce the population, and that the World Health Organization (WHO) is a genocidal organization.

Sabmyk community. The last community stands out, with over 85% of its 232 channels sharing URLs related to conspiracy theories. Almost the entire community (95% of channels) communicates in English. The authority scores of this community highlight a unique pattern with respect to the other communities. We discover that the only true authority is the *sabmyk* channel, while all the other channels of the community primarily forward its messages. Searching for information on the web, it emerges that Sabmyk is a complex conspiracy theory proposed as a successor to QAnon [335]. This theory celebrates the cult of a new messianic figure called Sabmyk, who actively promotes various conspiracy theories, *i.e.,* against COVID-19 vaccines and concerning the 2020 US elections [180].

Longitudinal analysis

An interesting aspect to explore is how conspiracy communities have evolved over time. We analyze this dimension in Figure 7.2, which shows the number of channels created daily on Telegram. The Figure is divided into five parts: the first four charts (a,b,c,d) analyze each community, while the last one (e) compares the four conspiracy communities aggregated against the other Telegram communities. The chart shows

Table 7.3. Summary of metrics about e-commerce platforms.

Platform	**URLs**	**Products**	**Affiliation**
Amazon	61,170	19,908	34,980
Teespring	2,415	256	-
eBay	1,969	879	161
Etsy	1,285	367	-
Total	66,839	21,410	35,141

that the creation of conspiracy channels is not evenly distributed over time. Instead, we find two spikes in channel creation. The first one begins around mid-March 2020, reaches a peak in May, and starts declining until June 2020. Instead, the second spike is much steeper and goes from 2021-01-06 to the middle of March 2021.

Figure 7.2 (e) shows a first insight into this phenomenon. The increase in channel creation is more evident in the conspiracy communities (black line) than in the rest of Telegram (purple line). This is particularly evident in the second spike. This behavior suggests that the spike in channel creation is not merely a result of overall Telegram platform growth. Instead, we hypothesize that some specific event may have contributed to the abnormal growth of the conspiracy channel. Thus, we examine the messages and descriptions of the conspiracy channels created during these two periods to gain insights into the reasons for their creation. The first peak can be directly linked to the stringent COVID-19 restrictions imposed in Europe during that period and is more prevalent in the German and Neo-Latin communities. We find the surge in Telegram channels offering alternative viewpoints on the pandemic. Instead, we find that the second peak is linked with the unprecedented Capitol Hill events. During this period, we observed the emergence of several pro-Trump channels, especially in the English community, that became focal points to promote alternative discussions surrounding these events.

7.3 Monetization

The manual review of hundreds of messages from the Conspiracy Channel Dataset during the previous phase highlighted an interesting phenomenon. Indeed, while most of the effort of conspiracy channels is in promoting their theories to recruit followers, we have identified some suspicious messages posted to sell products or promote crowdfunding campaigns. This discovery raises the question of whether some conspiracy channels try to exploit their followers for financial gain. Intrigued by this aspect, we leverage the Extended Conspiracy URLs Dataset to systematically analyze conspiracy channels' possible monetization strategies and quantify the magnitude of this phenomenon. After an explorative analysis of the links shared by the channels, we identify three main strategies to monetize: affiliation, donation, and crowdfunding.

7.3.1 Affiliation Programs

Conspiracy channels adopting this strategy use affiliation programs from e-commerce platforms to earn a commission. Thus, as a first step, we identify the most popular e-commerce platforms they can utilize for this purpose. To this end, we leverage SimilarWeb [9], a popular web analytics service that categorizes websites and ranks them by traffic. We retrieve all the 42 services reported in the global rank for the *eCommerce & Shopping* category. Then, we extract URLs containing the domain of these platforms from the Extended Conspiracy URLs Dataset, finding 68,819 URLs from 28 platforms. Among the top five platforms, covering 97.83% of e-commerce links, only Amazon and eBay offer an affiliate program. In this program, a participant, known as partner, can generate unique links pointing to products on the platform that embed his unique identification number. The partner earns a commission for any purchases made by users who land on the platform through his link. In particular, Amazon's partners earn a commission, between 1% and 12% depending on the categories of the product and the location of the targeted market (*e.g.,* amazon.com, amazon.de) from all the items bought in the next 24 hours from arrival through the link [28]. Similarly, on eBay, a partner earns a commission between 1% and 4% on the items purchased on the platform in the following 24 hours [129].

Overall, we find 61,170 URLs pointing to Amazon and 1,969 to eBay in the Conspiracy Channel Dataset. To detect URLs belonging to the affiliation program, we search for URLs containing the parameters *tag=* for Amazon and *campid=* for eBay. Surprisingly, we discover that 34,980 (57.18%) of Amazon's URLs and 161 (8.18%) on eBay are affiliated links granting commissions to the conspiracy channels and that 1,870 channels (10.42% of our dataset) leverage this strategy. Unfortunately, since there is no public information available about the partner programs, we can not estimate the gain of the channels with this strategy.

Through our Amazon scraper, we discuss the type of goods sold on this platform. We obtain the item category for 52,117 of the 61,170 Amazon URLs, representing 17,375 different products. Of them the most five advertised products are books with 26,409 (50.6%) URLs and 4,539 (26.7%) different titles, followed by 5,493 (10.5%) URLs of health & personal care with 1,795 (10.3%) distinct products, 3,750 (7.1%) URLs related to electronic devices of which 2,182 (12.5%) unique, 3,194 (6.1%) URLs belonging to the category of home & kitchen with 1,897 (10.9%) different items and finally 2,487 URLs of fashion products with 1,795 (10.3%) different goods.

7.3.2 Donation platforms

The second monetization strategy is asking the subscribers for donations to support the channel and its activity. Also in this case, we look for donation platforms that conspiracy channels can utilize. Since Similarweb and equivalent services do not provide a category for donation platforms, we manually collect them by analyzing the results of Google queries such as: *top donation platforms.* Given the presence of language specific communities, we we conducted country-specific queries containing different languages and keywords. Moreover, to avoid search results on Google from being affected by our browsing history or geo-location, we conduct queries using

Table 7.4. Summary of metrics about donation platforms. Gain with (*) indicates monthly earnings.

Platform	**URLs**	**Profiles**	**Gain**	**Donors**
Paypal/donate	88,572	1,634	-	-
Patreon	33,691	480	$261,822 *	100,657
SubScribeStar	3,482	140	$5,947 *	5,746
BuyMeACoffee	3,383	131	$810,372	15,668
Ko-Fi	1,774	67	$99,162	8,759
DonorBox	1,732	59	-	-
Total	133,896	2,522	1,177,303	130,830

a VPN from different vantage points in countries that roughly correspond to the languages of our conspiracy communities (Spain, Portugal, Germany, UK, Brazil, USA). In this way, we collect 31 donation services.

Then, we analyze the presence of donation platforms in the URLs shared in the Extended Conspiracy URLs Dataset. We detect 133,896 URLs belonging to 15 different services, shared by 5,804 channels, accounting for 32.3% of our dataset. The most used platforms by number of URLs are Paypal/donate [291] with 88,572 URLs (accounting for 65.6% of the donation URLs), Patreon with 33,691 URLs (about 25% of the donation URLs), SubScribeStar with 3,482 (2.6%), BuyMeACoffee with 3,383 (2.5%), Ko-Fi with 1,774 (1.3%), and DonorBox with 1,732 (1.3%).

During the exploration phase, we discover messages sharing donation URLs with the intent to discredit genuine content creators' donation campaigns rather than promote them. To discard these cases, we examine the donation URLs shared both in the conspiracy communities and the rest of Telegram–*i.e.,* the communities not flagged as conspiracy. Then, three independent researchers analyzed the related Telegram messages to ascertain whether the conspiracy channels were promoting or discrediting the donation campaign. At the end of the process, we only find 79 donation URLs, of which we discard 62.

To estimate the potential earnings of the Conspiracy Channel Dataset on these platforms, we use the Selenium Framework [325] to create a custom crawler for the six most used donation platforms in our dataset, covering 90.05% of the URLs. Unfortunately, PayPal/donate and DonorBox do not provide information about donation amounts and the number of donors. Thus, by scraping the web pages of these services, we could only determine the number of unique profiles still reachable. Instead, we obtained information regarding the number of donors and donated amounts for most profiles on the other services. Specifically, for Patreon and SubscribeStar, we retrieve the number of donors and the total amount of money donated in the last month. As for BuyMeACoffee and Ko-Fi, we collect data on the number of donors and the total amount of money raised by the campaigns' creators.

Tab 7.4 shows the number of URLs we find for the six most used services, the number of created campaigns, the amount of money donated, and the number of donors. Among the profiles we examined, the one that raised the most money through this strategy is *QAnon Anonymous*[8], which received an impressive $87,955.63 from 20,103 subscribers using Patreon in September 2023 alone.

Table 7.5. Summary of metrics about crowdfunding platforms.

Platform	URLs	Projects	Funds	Backers
Givesendgo	14,976	167	$9,495,176	197,947
Paypal/pools	7,201	43	$116,322	3,579
GoFundMe	6,559	1,566	**$57,593,611**	594,119
DonorBox	1,139	31	$191,625	4,280
Fundly	310	14	$331,989	5,156
Kickstarter	306	86	$15,610,337	106,654
Fundrazr	235	38	$2,589,506	34,376
Indiegogo	136	41	$5,989,013	42,931
Total	30,862	1,986	$91,917,579	985,820

7.3.3 Crowdfunding and Fundraising services

Finally, we analyze the crowdfunding campaigns. Also in this case, we manually build a list of popular crowdfunding services performing Google queries such as *best crowdfunding services*. Similar to our earlier approach, we perform multiple queries in different languages and from different vantage points. At the end of the process, we collect a list of 49 crowdfunding or fundraising websites. Looking for URLs containing the domain of these platforms in the Extended Conspiracy URLs Dataset, we find 30,887 URLs from 18 of them, shared by 3,531 channels, covering 22.8% of our dataset. The most used services are GiveSendGo with 14,976 URLs, followed by Paypal/pools with 7,201 URLs, GoFundMe with 6,559 URLs, and DonorBox with 1,139 URLs. As for the donation links, we find evidence that conspiracy channels run misinformation campaigns against genuine projects [1]. Thus, we use the same approach as the previous subsection to discover these cases. In this way, we find 65 URLs in common between the two parts of the dataset, of which we discard 46.

As done for the other strategies, we implemented a scraper with the Selenium Framework for each platform. This scraper allows us to collect information about each campaign's earnings and analyze their status. Indeed, a campaign can be completed or ongoing. Kickstarter enables creators to access the funds only if they reach a predefined target funding at the end of the campaign. Instead, on other platforms, the campaign creator can access the funds raised while the campaign is ongoing. An exception is Indiegogo, which allows creators to choose between the two options when starting a campaign. Upon examining the campaigns' statuses, we discover eight campaigns on Kickstarter and one on Indiegogo that concluded without reaching their fundraising goal. Thus, we do not include the funds raised from these campaigns.

Tab. 7.5 reports the number of URLs, different projects, money raised, and number of backers for the top eight services by number of URLs. As it is possible to note, this strategy is the most remunerative, collectively funding conspiracy theorists with over $90M. In the following we describe some of the most popular campaigns in detail.

Crowdfunding campaigns analysis

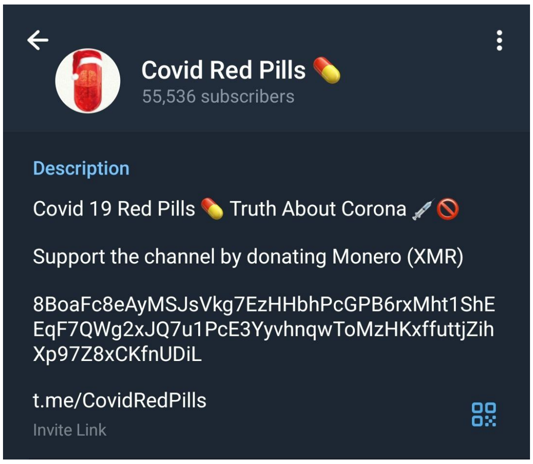
In this section, we analyze some of the crowdfunding campaigns that are shared most frequently in the conspiracy communities. Focusing on the top 5 platforms that collect the highest funds, we discover that several campaigns fall into the following categories:

Campaigns supporting far-right. We have identified several fundraising campaigns associated with far-right projects. Many of these campaigns [210, 306, 205] are focused on gathering funds to cover the legal costs of individuals who participated in the January 6th Capitol Hill riot. These campaigns alone collected an astonishing $160,816 with 2,891 backers. Interestingly, we find these campaigns more frequently on the Givesendgo platform. However, we discover this platform is well known to promote extremist content [15].

Campaigns about COVID-19. We find that GiveSendGo campaigns are also frequently linked to COVID-19. In particular, we have identified campaigns aimed at raising funds for the Freedom Convoy 2022 [176], a movement of protesters against government COVID-19 policies, including restrictions and vaccine mandates. Interestingly, we detect similar campaigns also on GoFundMe. However, their links redirect to a refund page, as GoFundMe prohibits support for content promoting violence and harassment. [170] Looking at the news, we discover that one of these campaigns was the biggest gainer of the platform, with over $10 million from over 120,000 donors [51]. Finally, we find projects on the Indiegogo and Kickstarter platforms that aim at funding documentaries to unveil the truth behind the COVID-19 pandemic [92, 341, 377]. These platforms have stricter policies on campaign content. However, moderating campaigns involving potentially pseudo-scientific content is more complex as it raises concerns about freedom of expression.

Scam campaigns. We detect crowdfunding campaigns that are outright scams. An example is a campaign [254] on Fundrazor claiming to collect funds for starving children in Venezuela to donate to the Save the Children charity organization. The campaign is now closed, and a banner on the page states that Save The Children has confirmed they have no association with it. Unluckily, the campaign was able to amass $39,500 from 602 donors before it was shut down. Another project [14], on Indiegogo, involves fundraising for a portable air cleaner that reportedly contains a high-frequency generator that purifies air from pathogens. The campaign has been closed after raising over $7.4K, as the creator claimed that Silicon Valley internet companies have hindered the project.

Campaigns against the establishment. Finally, we find campaigns that collect funds to challenge government policies or influential individuals. An example is a campaign [31] that collects funds to counter the educational indoctrination of U.S. schools. The campaign is promoted by a non-profit organization called American Education Defenders, Inc., and proposes an educational program called "America’s 52 Videos". This program shares true stories from American history and teaches life lessons to motivate young people and instill American values. Another campaign [70] is about the "Pyramid of Power", a documentary series exploring the individuals and institutions manipulating and controlling the world. These two campaigns gained more than $37K from 477 contributors.



Covid Red Pills 💊

55,536 subscribers

Description

Covid 19 Red Pills 💊 Truth About Corona 💉🚫

Support the channel by donating Monero (XMR)

8BoaFc8eAyMSJsVkg7EzHHbhPcGPB6rxMht1ShEEqF7QWg2xJQ7u1PcE3YyvhnqwToMzHKxffuttjZihXp97Z8xCKfnUDiL

t.me/CovidRedPills

Invite Link

Figure 7.3. A channel displaying in its description the address of a blockchain (Monero).

7.3.4 Other Strategies

In addition to the previously analyzed monetization strategies, we find conspiracy channels can exploit other sources of revenue.

Other Amazon features. The first one is related to the Amazon Influencer Program [30]. This program enables Influencers to create an Amazon web page with some selected products, earning commissions on sales. These pages are easily detectable because they contain the */shop/* string in their URL. Looking for this pattern in our dataset, we find 25 different shops.

Another possibility is using Amazon's wish lists, essentially lists of products a user desires. These lists can be private (visible only to the creator) or public. In the latter case, anyone with access to the wish list link can gift a product to the list's creator. To identify wish lists in our dataset, we extract URLs containing the */wishlist/* string in their path. This process led us to 119 URLs, pointing to 32 distinct public lists. The inspection of these lists revealed that 12 are no longer accessible, and the others contain a wide variety of products, including underwear, vitamins, survival kits, and prepaid cards.

Blockchain addresses. In our analysis, we discover conspiracy channels asking for cryptocurrency donations in their descriptions. Fig. 7.3 shows the description of the Covid Red Pills channel, where the administrator requests financial support for the channel through the donation of Monero coins. We use regular expressions to extract wallet addresses of the most popular blockchains (Bitcoin and Ethereum), as well as the prominent privacy-preserving blockchains (Monero and Zcash). We identified these blockchains' addresses in 40 channels, and through manual verification, we confirmed that 29 of them are used for donations. Analyzing the BTC wallets, we find they received 115 transactions, totaling 0.5 BTC ($\approx \$13,000$). Performing the same analysis on Ethereum, we find that the wallets received 42 transactions, amounting to 5.5 ETH ($\approx \$9,000$).

Custom websites. Analyzing the URLs and examining messages within the conspiracy channels, we observed frequent promotion of custom e-commerce sites or personal websites. Inspecting these websites, we discover that many of them feature dedicated donation sections with blockchain addresses or various payment options. Thus, we perform a raw analysis to estimate the magnitude of the phenomenon. In particular, we look for URLs containing the words: *shop, products, store, produkt, collections, donate, donations or support* as third-level domain or that have these words in the URL's path. As a result, we find 39,592 URLs (12,664 unique) matching our definition. Unfortunately, we can not validate or analyze this huge amount of websites since it requires a heavy manual effort or to build custom parsers.

Drive traffic to video hosting services. The URLs analysis also revealed that conspiracy channels share a considerable number of links to popular video hosting services such as YouTube (3,634,894 URLs) or BitChute [360] (277,033 URLs). While some channels may share videos as a resource to confirm their theories, it is also well known from previous work [45], that some of them leverage such platforms to monetize their content through Partner Programs (*e.g.,* YouTube Partner Program [394]). These programs allow content creators to earn money by placing advertisements in their videos and paying them proportionally to the time they are viewed. While the in-depth analysis of this phenomenon falls outside the scope of this work, we believe that the dataset of URLs we release can be a valuable resource for future research in this area.

Channel ads. Finally, conspiracy channel administrators could also monetize by publishing sponsored messages to their subscribers. There are mainly two methods to implement this strategy. The first relies on a feature recently introduced on Telegram, the Sponsored Messages [351]. This functionality enables channel owners to share sponsored messages to receive a share of the advertising revenue. However, it is worth noting that this feature is relatively new and still in the beta phase. The second approach involves using external services like *telega.io* [349], which act as intermediaries between channel administrators and advertisers or establish private deals directly between advertisers and channel administrators. However, this kind of sponsored message is likely impossible to detect when products are deceptively promoted into the content and storyline of the channel.

7.4 Discussion

Our analysis reveals a clear distinction between the products promoted by conspiracy theory channels on Amazon compared to other online marketplaces. Indeed, while Amazon features standard products like books, masks, and water filters, we find a range of questionable products (*e.g.,* 5G shields, EMF stone protectors, and healing wands) on eBay, Etsy, and Teespring. The distinction is likely attributed to the different content policies of these platforms. Indeed, Amazon upholds a more rigorous content policy compared to the other services. Nevertheless, it is important to emphasize that these products are not inherently harmful. Indeed, the concern lies in the narratives and promotions associated with these items. For example, ordinary substances like Sodium Chloride and Chlorine Dioxide, commonly found in pharmacies, are promoted in four channels as essential components to prepare the so-called "Miracle Mineral Supplement", which is reported as a miraculous cure for various diseases, including cancer and HIV. Similarly, we discover 35 channels sharing links to buy seemingly innocuous white pine needles at an exorbitant price of

$150 on Etsy. The concern lies in the accompanying message, that presents a guide for COVID-19 survival. This guide discourages seeking medical care in hospitals and suggests homemade remedies, including tea prepared with the costly pine needles mentioned above.

Although Amazon has a rigorous policy about the items sold and actively operates to ban QAnon merchandising [277], it has a less strict policy about book content. Quoting Amazon policies: *As a bookseller, we believe that providing access to the written word is important, including content that may be considered objectionable.* [29]. The combination of this less strict policy and the simplicity of self-publishing on Amazon allowed conspiracy theorists to spread their ideas and monetize through book sales. Indeed, nearly 50% of the links point to books and almost half of them include an affiliate tag. Looking at the most frequently occurring authors of books promoted using non-affiliate links, we discover three relatively obscure German writers: VEIBZ (3,092 occurrences), MERKSAM (1,492 occurrences), and EBURD (819 occurrences). Their books delve into several conspiracies with the goal of revealing the truth on subjects such as: *"NASA & Elon Musk – They lie & cover-up", "CERN & its satanic roots", "HAARP & CERN use Alien Tech"*. Fig. 7.4 shows the cover of the four most shared books.

Asking for financial support through donation platforms or the use of referral links is not illegal. Indeed, web content creators often use these tools to finance their activities. Similarly, crowdfunding platforms can be used for good purposes, such as creating new products or promoting noble actions. Problems arise when the raised money is used to finance borderline, if not illegal, activities that can threaten society.

Concerning the promoted items, there are also problems related to the transparency of the activity and compliance with the referral programs. Indeed, according to the partnership agreement of Amazon and Ebay [28] [129], a partner—a participant that can generate referral links and earn commissions—has to clearly disclose their partnership. Moreover, near each affiliated link should be a disclosure such as *"(paid link)", "#ad", or "#CommissionsEarned"*. This information is needed to inform the customers that a conflict of interest exists on the promoted item.

7.5 Mitigation

On the basis of the concerns reported in the previous section, we identify the following points that should be addressed to mitigate the issues:

1. The user should be aware that the information provided in the channel is questionable or pseudo-science.

2. The user should be aware that the medical advice posted on these channels does not come from the official medical sources and could be dangerous to their health.

3. The user should be aware that there is a potential conflict of interest on the promoted items.

VEIBZ
BOTSCHAFT DER GALAKTISCHEN FÖDERATION 1
Erstkontakt - Der Zeitpunkt der Offenlegung ist da
Keine Verschwörung der Kabale kann diese Enthüllung & den Aufstieg
In die 5D Matrix mehr aufhalten
MERKSAM
BOTSCHAFT DER PLEJADEN 1
Energieübertragende Channelings Zur Matrix Transformation
des Gefängnisplanet Erde Spiritueller Ratgeber zur
Lichtarbeit für den 5D Aufstieg ins Goldene Zeitalter
VEIBZ
BOTSCHAFT DER GALAKTISCHEN FÖDERATION 2
Die wahre Menschheitsgeschichte Verborgene Geheimnisse des Homo Anunnaki Nibiruensis Alien Botschaften zur Selbstfindung
VEIBZ
BOTSCHAFT DER GALAKTISCHEN FÖDERATION 3
Reptilian mRNA Genesis
Nachricht der großen Blonden
Nordics von Lyra, Venus, Protykon & Sirius

Figure 7.4. The covers of the four most shared books by conspiracy channels available for sale on Amazon.

4. The user that navigates on a crowdfunding campaign should be aware that it is promoted by a conspiracy theory channel.

In the following subsections, we present some solutions that implement the above-mentioned points.

7.5.1 The Channel Checker Bot

To mitigate the concerns, we implement The Channel Checker Bot. It is a Telegram Bot that receives a channel name from the user as input and provides the user with information about the input channel.

The Channel Checker Bot comprises four distinct components. First, there is a Telegram chatbot implemented in Python 3. The second component is the Bot Engine, a remote server hosted on our machine. It is written in Python 3 and Flask [7] and handles the Telegram Bot's logic. Specifically, it queries and updates the database, parses Telegram channels, and analyzes URLs to find matches in our datasets. The third component is the database, which stores information about analyzed channels. Finally, there is a headless Telegram client used to join channels not present in our database and retrieve their content via Telegram APIs. The architecture of the Channel Checker Bot is illustrated in Fig. 7.5.

To interact with the bot, a Telegram user has to search for it through the client search bar and start a conversation with it. The conversation starts with the user who writes into the chat the name of the target channel ①. The chatbot interface parses the chat and forwards to the Bot Engine the channel name ②. The Bot Engine looks up on the database if the target channel is already present ③. If it is, the Bot Engine retrieves all the needed information ④ and forwards them to the bot ⑦. The bot formats the data to produce a visually appealing representation (see Fig. 7.6) and shows it in the user's chat ⑧. If the channel is not present in the database, the Bot Engine instantiates the headless Telegram client ⑤, which, via

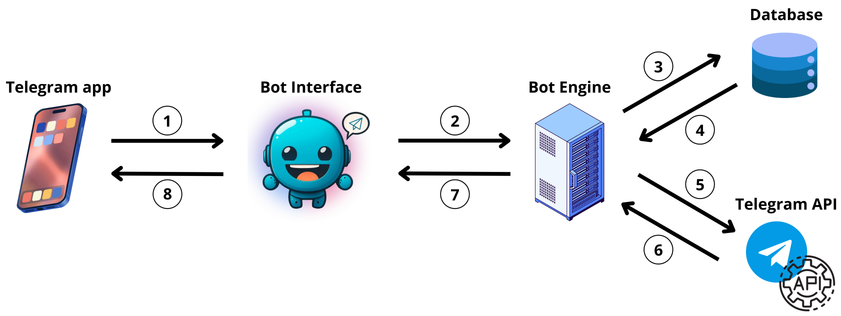


Figure 7.5. Channel Checker Bot architecture.

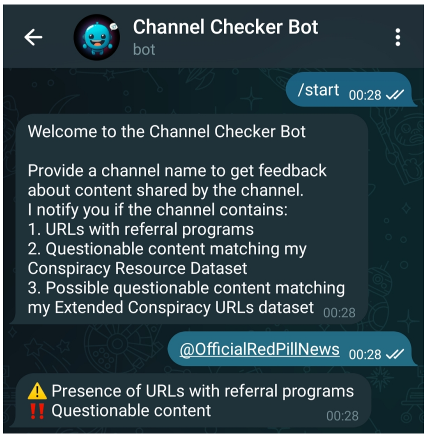
Telegram APIs, joins the channel and retrieves all the contents shared into it ⑥. The channel's content is provided back to the Bot Engine ⑦, which parses it and extracts the links. Finally, each link is analyzed. Firstly, the engine searches if links contain one of the referral program tags. In the case there is the presence of a referral program, the engine labels the channel accordingly. Then, the Bot Engine looks for links matching the Conspiracy Resource Dataset, if matching URLs are present, the Bot Engine labels the channel as *"Questionable content"*, and adds all the links in the Extended Conspiracy URLs dataset. If none of the URLs match the Conspiracy Resource Dataset, but there are matching URLs with the Extended Conspiracy URLs datasets, the Bot Engine labels the channel as *"Possible questionable content"*. Once finished with all the analysis, the Bot Engine updates the database ③ and provides to the Bot the response ⑦ ⑧.

The Channel Checker Bot plays a dual role. It delivers a service to users while consistently updating the list of the conspiracy channels and the Extended URLs dataset. We believe that delving deeper into analyses of the Extended Conspiracy URLs dataset has the potential to unveil heretofore unknown resources disseminating content of questionable credibility.

Lastly, it is crucial to emphasize that our system retains no direct information about users who engage with it (such as IP addresses) or any details that could be employed to associate a request with a user at a later time (*e.g.,* timestamps or Telegram client versions). The system exclusively stores information related to the channels involved.

Matching rule	Message shown by the plug-in
Marketplace URL present in our dataset	This item is promoted by Telegram channels endorsing conspiracy theories. Prior to purchase, conduct independent research and consult a doctor for medical use.
Campaign URL present in our dataset	This campaign is promoted by Telegram channels endorsing conspiracy theories. It is potentially a scam or questionable fund utilization. Before contributing, conduct thorough research.
Video or Text content URL present in our dataset	This content is promoted by Telegram channels endorsing conspiracy theories. This page could contain questionable content, unreliable medical indications, or promoting pseudo-science.

Table 7.6. Messages displayed by the ConspiracyAlert plug-in when a user lends on a matching URL.



Channel Checker Bot
bot

/start 00:28

Welcome to the Channel Checker Bot

Provide a channel name to get feedback about content shared by the channel.
I notify you if the channel contains:
1. URLs with referral programs
2. Questionable content matching my Conspiracy Resource Dataset
3. Possible questionable content matching my Extended Conspiracy URLs dataset 00:28

@OfficialRedPillNews 00:28

⚠️ Presence of URLs with referral programs
‼️ Questionable content 00:28

Figure 7.6. An instance of the Channel Checker Bot chat.

7.5.2 ConspiracyAlert: A Browser Plug-in

The Channel Checker Bot serves as a warning system for Telegram users navigating channels within the platform. However, with it, it is not possible to warn users who browse the web and lend to questionable products, content, or crowdfunding campaigns. To address this gap, we developed a browser plug-in named *ConspiracyAlert*. It is a plug-in for the Chrome and Firefox browsers written in Typescript. Once installed on the browser, the plug-in monitors the navigation of the user. Upon the complete loading of a webpage, it captures the URL and transmits it to a remote server. This server accesses both the Conspiracy Resource dataset and a dataset containing all the URLs identified in Sections 7.3.1, 7.3.2 and 7.3.3, searching for matching URLs. In case of a match, the remote server generates an informative message to be displayed by the plug-in. Tab. 7.6 shows the warning messages displayed to the users.

Similar to the Channel Checker Bot, the ConspiracyAlert plug-in upholds a commitment to user privacy by not retaining any information about the user or their browsing history.

7.5.3 Fully Integrated Mitigation Systems

In the previous subsections, we depicted solutions to mitigate issues raised by conspiracy theories channels. However, when it comes to alerting users outside the Telegram ecosystem, we find the ConspiracyAlert plug-in to be the most effective and efficient solution. Instead, with respect to the Channel Checker Bot, we propose a more efficient and transparent solution for the user, in particular from the usability point of view. Indeed, it is possible to integrate the flow of the Channel Checker Bot directly into the Telegram's Client. In this scenario, when a user joins a channel, the client automatically queries our systems and displays a pop-up warning about potential issues associated with the newly joined channel. This seamless integration enhances user experience and ensures proactive awareness without the need for manual intervention. Although it is possible to develop a custom client, given the open-source nature of Telegram, we believe that widespread adoption of such a solution would be effective only if integrated with the official client release.

As a last solution, there is the introduction of new tags within the existing tag system employed by Telegram. Currently, Telegram utilizes the *"Verified"* tag, a blue mark that is granted to channels and accounts whose owner verified his identity. Then, there are the *"Scam"* and *"Fake"* tags. These two tags are respectively applied to channels that pretend to be VIPs or engage in fraudulent activities. To the best of our knowledge, Telegram applies these two tags on the basis of the users' reports. We envision that the introduction of a new tag, for instance, the *"Questionable"* one, could benefit the Telegram ecosystem. This tag could serve to alert users about potential threats in the contents of specific channels.

7.6 Limitations

As mentioned in Sec. 7.2.2, we build the *Conspiracy Resources Dataset* gathering information from previous work focused on conspiracy theories. However, most of the scientific literature focuses on analyzing English content. This limitation could introduce a bias, as conspiracy communities operating in other languages, such as Russian or Indian, might evade detection due to their use of non-English sources. Moreover, other platforms not considered in our study (*e.g.,* Parler [46]) are known to host conspiracy-related content. Unfortunately, we could not find works providing conspiracy-labeled datasets suitable for our study. Throughout our investigation, we do not attempt to infer the direct link between the channel's administrator and the ultimate recipient of funds. In certain situations, this connection is clear, such as when the channel's name matches a profile on an external platform or in the case of affiliate program campaigns. However, in other instances, such as crowdfunding campaigns, it proves challenging to discern the ultimate objective of the channel's administrator. However, it is clear that someone is profiting and that the channels have a key role in fueling the conspiracy theories' money machine.

7.7 Ethical considerations

The dataset we analyze does not contain personal information like phone numbers or any media that could include adult content or copyrighted material. Furthermore, the channels mentioned in our study are publicly accessible and represent widely recognized public figures or entities. In our data collection process, we scraped web pages of the analyzed platforms. We adopt a careful approach to prevent flooding and ensure a minimal impact on these services by limiting the volume of requests submitted.

7.8 Conclusion

In this chapter, we focused on understanding and quantifying how conspiracy theories raise funds by exploiting Telegram. We started by identifying the conspiracy theory-related channels, analyzing a novel dataset we built by collecting previously validated resources from an extensive literature review.

This study revealed the alarming finding that more than 15% of all Telegram channels in the TGDataset (17,806 channels) are linked to conspiracy theories. Then, we discover that conspiracy theory-related channels actively seek to profit from their subscribers. We provide a taxonomy of all the diverse monetization strategies we find in our dataset and dive into the analysis of the three most common. Our study shows that conspiracy theories raised funds for $90 million by arranging crowdfunding campaigns. As a future work, we believe it is interesting to conduct a more comprehensive analysis of the monetization strategies reported in 7.3.4 to get deeper insights into the impact of monetization. Concerning conspiracy communities, a potential investigation is thoroughly examining the diverse monetization strategies these distinct communities adopt. Finally, another possible direction is analyzing the channels that use the same affiliate program ID and those that share identical funding projects. This study could highlight the collaborative patterns presented by these channels and enable the identification of more fine-grained sub-communities.

Chapter 8

Final remarks and future direction

As digital systems become increasingly integral to our daily lives, it is critical to recognize and address the potential risks accompanying these innovations. This thesis investigates several platforms, analyzing their ecosystems to uncover vulnerabilities and risks for their users. Our research on ASN lifetimes revealed the presence of squatting, often associated with malicious activities like spamming and prefix hijacking. We identified 3,051 suspicious events using a filtering technique, confirming 76 as malicious through external sources. The analysis of the blockchain ecosystem highlights the prevalence of pump and dump schemes orchestrated via Telegram. We compiled a dataset of over 900 events and developed a classifier that effectively identifies these fraudulent activities using market data alone. We extended our blockchain analysis to include token and liquidity pool frauds on the BNB Smart Chain and Ethereum. We found that a small fraction of addresses created a disproportionate number of tokens, primarily used to perform rapid rug pulls. Our analysis identified over 290,000 potential rug pulls, revealing that organizers amassed approximately $240 million through these scams. A particular type of trader bot, called a sniper bot, is often involved in these activities. We found that these bots are highly active and significantly affect the AMM ecosystems. We analyzed how they work and studied their market impact, finding notable differences between Ethereum and BSC regarding success rates and required investments. Finally, we investigated instant messaging platforms, particularly Telegram, uncovering a substantial presence of conspiracy theory-related channels. These channels actively monetize their activities, raising significant funds through various strategies, including crowdfunding and selling merchandise.

The insights gained from this thesis highlight the need for enhanced monitoring and more robust security measures to protect users from emerging threats. By understanding specific vulnerabilities and attack vectors in different environments, stakeholders can implement more effective countermeasures. The evolving digital landscape presents both new opportunities and challenges, particularly with the advent of advanced AI techniques. AI can play a dual role, aiding analyses while potentially facilitating malicious activities, such as the promotion of conspiracy theories. Therefore, understanding how AI can be used to improve security measures and mitigate its misuse by attackers will be a key area of future research. Another promising direction involves identifying networks of malicious actors engaged in complex, collaborative fraudulent activities. Finally, the awareness of potential threats gained from this thesis can be used to educate users about the risks associated with digital platforms, promoting better security practices.

Bibliography

[1] 2020. Donations to Black Lives Matter Group Don't Go to DNC. https://www.factcheck.org/2020/06/donations-to-black-lives-matter-group-dont-go-to-dnc/.

[2] 2020. Telegram, the powerful COVID-19 choice of communications by many governments. https://www.channelnewsasia.com/commentary/coronavirus-covid-19-government-telegram-whatsapp-fake-news-info-936061.

[3] 2021. Telegram Analytics. https://tgstat.com/.

[4] 2021. *Telethon's Documentation.*

[5] 2023. *Coinmarketcap Fastest Alerts.* https://t.me/CMC_fastest_alerts

[6] 2023. Fake News Corpus. https://github.com/several27/FakeNewsCorpus.

[7] 2023. Flask. https://flask.palletsprojects.com/en/3.0.x/.

[8] 2023. Qanon Anonymous. https://www.patreon.com/qanonanonymous.

[9] 2023. SimilarWeb. https://www.similarweb.com.

[10] France 24. 2022. *Germany weighs ban on Telegram, tool of conspiracy theorists.* https://www.france24.com/en/live-news/20220126-germany-weighs-ban-on-telegram-tool-of-conspiracy-theorists

[11] J. Abley and W. Sotomayor. May 2015. *RFC7534: AS112 Nameserver Operations.* https://tools.ietf.org/html/rfc7534

[12] Hayden Adams, Noah Zinsmeister, and Dan Robinson. 2020. Uniswap v2 Core. https://uniswap.org/whitepaper.pdf.

[13] adamsnipes. 2022. Pancakeswap Bot & Uniswap Bot. https://adamsnipes.io/home.html.

[14] ADL. 2020. *AntiVirBag-Portable Air Cleaner, Ionizer, Ozonizer.* https://www.indiegogo.com/projects/antivirbag-portable-air-cleaner-ionizer-ozonizer--2#/updates/all

[15] ADL. 2023. *ADL Crowdfunding Report: How Bigots and Extremists Collect and Use Millions in Online Donations.* https://www.adl.org/resources/report/adl-crowdfunding-report-how-bigots-and-extremists-collect-and-use-millions-online

[16] Anti-Defamation League (ADL). 2023. *Fall of the Cabal.* https://www.adl.org/glossary/fall-cabal

[17] Anti-Defamation League (ADL). 2023. *Goyim Defense League.* https://www.adl.org/resources/backgrounder/goyim-defense-league

[18] Anti-Defamation League (ADL). 2023. *The Goyim Know/Shut It Down.* https://www.adl.org/resources/hate-symbol/goyim-knowshut-it-down

[19] AfriNIC. [n. d.]. *AfriNIC ftp.* Retrieved 2020-05-17 from ftp://ftp.afrinic.net/pub/stats/afrinic/

[20] AfriNIC. Sep. 2015. *AFRINIC - SPEARHEADING AFRICA'S INTERNET SINCE 2005.* Retrieved 2020-05-11 from https://afrinic.net/ast/pdf/afrinic-10years-ab-sept-2015.pdf

[21] Wasim Ahmed, Josep Vidal-Alaball, Joseph Downing, Francesc López Seguí, et al. 2020. COVID-19 and the 5G conspiracy theory: social network analysis of Twitter data. *Journal of medical internet research* 22, 5 (2020), e19458.

[22] Shiroq Al-Megren, Shada Alsalamah, Lina Altoaimy, Hessah Alsalamah, Leili Soltanisehat, Emad Almutairi, et al. 2018. Blockchain use cases in digital sectors: A review of the literature. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData).* IEEE, 1417–1424.

[23] Hayder M Alash and Ghaidaa A Al-Sultany. 2021. Enhanced Twitter Community Detection using Node Content and Attributes. In *2021 1st Babylon International Conference on Information Technology and Science (BICITS).* IEEE, 5–10.

[24] Max Aliapoulios, Emmi Bevensee, Jeremy Blackburn, Barry Bradlyn, Emiliano De Cristofaro, Gianluca Stringhini, and Savvas Zannettou. 2021. A large open dataset from the Parler social network. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 15. 943–951.

[25] Tejasvi Alladi, Vinay Chamola, Joel JPC Rodrigues, and Sergei A Kozlov. 2019. Blockchain in smart grids: A review on different use cases. *Sensors* 19, 22 (2019), 4862.

[26] Franklin Allen and Douglas Gale. 1992. Stock-price manipulation. *The Review of Financial Studies* 5, 3 (1992), 503–529.

[27] AltCensored. 2023. *AltCensored.* https://altcensored.com/

[28] Amazon. 2023. *Associates Program Standard Commission Income Statement.* https://affiliate-program.amazon.com/help/node/topic/GRXPHT8U84RAYDXZ

[29] Amazon. 2023. *Content Guidelines for Books.* https://www.amazon.com/gp/help/customer/display.html?nodeId=201995150

[30] Amazon. 2023. *Monetize your content with the Amazon Influencer Program.* https://affiliate-program.amazon.com/influencers

[31] Inc American Education Defenders. 2021. *Help Protect Our Kids Against The Raw Sewage Of CRT And Other Indoctrinations.* fundrazr.com/ourfuture

[32] Cisco and/or its affiliates. 2020. *BGPmon.* https://bgpmon.net/

[33] Demetris Antoniades, Iasonas Polakis, Georgios Kontaxis, Elias Athanasopoulos, Sotiris Ioannidis, Evangelos P Markatos, and Thomas Karagiannis. 2011. we. b: The web of short URLs. In *Proceedings of the 20th international conference on World Wide Web.* 715–724.

[34] ApeSwap. 2022. ApeSwap. https://apeswap.finance/.

[35] APNIC. 2020. *APNIC ERX resources.* https://www.apnic.net/manage-ip/manage-historical-resources/erx-project/erx-asn-transfer/

[36] APNIC. 2020. *APNIC ftp.* Retrieved 2020-05-17 from https://ftp.apnic.net/stats/apnic/

[37] Ehsan Arabnezhad, Massimo La Morgia, Alessandro Mei, Eugenio Nerio Nemmi, and Julinda Stefa. 2020. A light in the dark web: Linking dark web aliases to real internet identities. In *2020 ieee 40th international conference on distributed computing systems (icdcs).* IEEE, 311–321.

[38] ARIN. 2020. *ARIN ftp.* Retrieved 2020-05-17 from ftp://ftp.arin.net/pub/stats/arin/

[39] ARIN. 2020. *ERX Resources.* ftp://ftp.arin.net/erx/asn/erx-asns.txt

[40] ARIN. 2021. *WhoWas Service.* https://www.arin.net/reference/research/whowas/

[41] Jon Arnold, Olaf Maennel, Ashley Flavel, Jeremy McMahon, and Matthew Roughan. 2008. Quantitative analysis of incorrectly-configured bogon-filter detection. In *2008 Australasian Telecommunication Networks and Applications Conference.* IEEE, 10–15.

[42] Number Resource Organization (ASO). 2010. *RIR Comparative Policy Overview 2010-03.* https://www.nro.net/rir-comparative-policy-overview-2010-03/#1-3-3

[43] Osato Avan-Nomayo. 2022. PancakeSwap DEX reportedly set to block users from Iran. https://www.theblockcrypto.com/linked/133904/pancakeswap-dex-reportedly-set-to-block-users-from-iran.

[44] BakerySwap. 2022. BakerySwap. `https://www.bakeryswap.org`.

[45] Cameron Ballard, Ian Goldstein, Pulak Mehta, Genesis Smothers, Kejsi Take, Victoria Zhong, Rachel Greenstadt, Tobias Lauinger, and Damon McCoy. 2022. Conspiracy brokers: understanding the monetization of YouTube conspiracy theories. In *Proceedings of the ACM Web Conference 2022*. 2707–2718.

[46] Dominik Bär, Nicolas Pröllochs, and Stefan Feuerriegel. 2023. Finding Qs: Profiling QAnon supporters on Parler. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 17. 34–46.

[47] Massimo Bartoletti, Salvatore Carta, Tiziana Cimoli, and Roberto Saia. 2020. Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact. *Future Generation Computer Systems* 102 (2020), 259–277.

[48] Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 14. 830–839.

[49] Dirk G Baur and Thomas Dimpfl. 2018. Asymmetric volatility in cryptocurrencies. *Economics Letters* 173 (2018), 148–151.

[50] BBC. 2021. *Twitter suspends 70,000 accounts linked to QAnon.* `https://www.bbc.com/news/technology-55638558`

[51] BBC. 2022. *Freedom Convoy: GoFundMe seizes funds of Canada 'occupation'.* `https://www.bbc.com/news/world-us-canada-60267840`

[52] BBC. 2023. *The Light: Inside the UK's conspiracy theory newspaper that shares violence and hate.* `https://www.bbc.com/news/uk-65821747`

[53] Michael Bernstein, Andrés Monroy-Hernández, Drew Harry, Paul André, Katrina Panovich, and Greg Vargas. 2011. 4chan and/b: An Analysis of Anonymity and Ephemerality in a Large Online Community. In *Proceedings of the international AAAI conference on web and social media*, Vol. 5. 50–57.

[54] bgpmon. Sep. 2014. *Using BGP data to find Spammers.* `https://bgpmon.net/using-bgp-data-to-find-spammers/`

[55] Binance. 2018. *Public Rest API for Binance.* `https://github.com/binance-exchange/binance-official-api-docs/blob/master/rest-api.md`

[56] Binance. 2022. Binance Chain Docs - JSON-RPC Endpoint. `https://docs.binance.org/smart-chain/developer/rpc.html`.

[57] Binance. 2022. BNB Chain Documentation. `https://docs.bnbchain.world/docs/learn/intro`.

[58] Binance. 2022. PancakeSwap Integrates Token Contract Scanning Directly on Its Swap Page. `https://www.binance.com/en/news/flash/7193825`.

[59] Binance. 2022. Proof of Authority Explained. `https://academy.binance.com/en/articles/proof-of-authority-explained`.

[60] Bitcointalk. 2018. *Bitcointalk official forum of Bitcoin.* https://bitcointalk.org

[61] BitTrex. 2018. *What are my trade limits?* https://support.bittrex.com/hc/en-us/articles/115003004171

[62] Ivan Blekanov, Svetlana S Bodrunova, and Askar Akhmetov. 2021. Detection of hidden communities in twitter discussions of varying volumes. *Future Internet* 13, 11 (2021), 295.

[63] Hudson Borges, Andre Hora, and Marco Tulio Valente. 2016. Understanding the Factors That Impact the Popularity of GitHub Repositories. In *2016 IEEE International Conference on Software Maintenance and Evolution (ICSME).* 334–344. https://doi.org/10.1109/ICSME.2016.31

[64] Charles Bovaird. 2021. *XRP Plunged More Than 50% After Its Latest Pump.* https://www.forbes.com/sites/cbovaird/2021/02/01/xrp-plunged-more-than-50-after-its-latest-pump

[65] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. 2007. On modularity clustering. *IEEE transactions on knowledge and data engineering* 20, 2 (2007), 172–188.

[66] bread. 2022. *Bread (BRD).* https://github.com/breadwallet

[67] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[68] Steven Brock. 2021. Scalping in Ecommerce: Ethics and Impacts. *Available at SSRN 3793357* (2021).

[69] Ryan Browne. 2021. *Tweets from Elon Musk and other celebrities send dogecoin to a record high.* https://www.cnbc.com/2021/02/08/tweets-from-elon-musk-and-celebrities-send-dogecoin-to-a-record-high.html

[70] Derek Broze. 2021. *Help Us FINISH The Pyramid of Power Documentary Series!* fundrazr.com/pyramidofpowerdoc

[71] THE XRP BULLY. 2021. *FEBRUARY 1ST, 2021 PUMP & HOLD XRP DISCLAIMER.* https://www.youtube.com/watch?v=cv9eVZMbwt0&t=8s

[72] Daniel Bumblauskas, Arti Mann, Brett Dugan, and Jacy Rittmer. 2020. A blockchain use case in food distribution: Do you know where your food has been? *International Journal of Information Management* 52 (2020), 102008.

[73] Vitalik Buterin et al. 2014. A next-generation smart contract and decentralized application platform. *white paper* 3, 37 (2014), 2–1.

[74] Trend Calendar. 2021. *Trending words on 30th January, 2021.* https://us.trend-calendar.com/trend/2021-01-30.html

[75] Yi Cao, Yuhua Li, Sonya Coleman, Ammar Belatreche, and Thomas Martin McGinnity. 2015. Detecting wash trade in financial market using digraphs and dynamic programming. *IEEE transactions on neural networks and learning systems* 27, 11 (2015), 2351–2363.

[76] Zhenfeng Cao, Minzhang Zheng, Yulia Vorobyeva, Chaoming Song, and Neil F Johnson. 2017. Dynamical patterns in individual trajectories toward extremism. *arXiv preprint arXiv:1706.01594* (2017).

[77] V. Cerf. Aug. 1990. *RFC 1174: IAB Recommended Policy on Distributing Internet Identifier Assignment and IAB Recommended Policy Change to Internet "Connected" Status.* `https://tools.ietf.org/html/rfc1174`

[78] Vint Cerf. Oct. 1969. *RFC 20: ASCII format for Network Interchange.* `https://tools.ietf.org/html/rfc20`

[79] Federico Cernera, Massimo La Morgia, Alessandro Mei, Alberto Maria Mongardini, and Francesco Sassi. 2023. Ready, Aim, Snipe! Analysis of Sniper Bots and their Impact on the DeFi Ecosystem. In *Companion Proceedings of the ACM Web Conference 2023*. 1093–1102.

[80] Federico Cernera, Massimo La Morgia, Alessandro Mei, and Francesco Sassi. 2023. Token Spammers, Rug Pulls, and SniperBots: An Analysis of the Ecosystem of Tokens in Ethereum and the Binance Smart Chain (BNB). In *32th USENIX Security Symposium (USENIX Security 23)*.

[81] Rakesh Vidya Chandra and Bala Subrahmanyam Varanasi. 2015. *Python requests essentials*. Packt Publishing Birmingham, UK.

[82] Jian Chang, Krishna K Venkatasubramanian, Andrew G West, Sampath Kannan, Boon Thau Loo, Oleg Sokolsky, and Insup Lee. 2011. AS-TRUST: A trust quantification scheme for autonomous systems in BGP. In *International Conference on Trust and Trustworthy Computing*. Springer, 262–276.

[83] Carl Chapman and Kathryn T Stolee. 2016. Exploring regular expression usage and context in Python. In *Proceedings of the 25th International Symposium on Software Testing and Analysis*. 282–293.

[84] Jing Chen and Silvio Micali. 2019. Algorand: A secure and efficient distributed ledger. *Theoretical Computer Science* 777 (2019), 155–183.

[85] Ting Chen, Yufei Zhang, Zihao Li, Xiapu Luo, Ting Wang, Rong Cao, Xiuzhuo Xiao, and Xiaosong Zhang. 2019. Tokenscope: Automatically detecting inconsistent behaviors of cryptocurrency tokens in ethereum. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*. 1503–1520.

[86] Weimin Chen, Xinran Li, Yuting Sui, Ningyu He, Haoyu Wang, Lei Wu, and Xiapu Luo. 2021. Sadponzi: Detecting and characterizing ponzi schemes in ethereum smart contracts. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 5, 2 (2021), 1–30.

[87] Weili Chen, YueJin Xu, Zibin Zheng, Yuren Zhou, Jianxun Eileen Yang, and Jing Bian. 2019. Detecting" pump & dump schemes" on cryptocurrency market using an improved apriori algorithm. In *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*. IEEE, 293–2935.

[88] Weili Chen, Tuo Zhang, Zhiguang Chen, Zibin Zheng, and Yutong Lu. 2020. Traveling the token world: A graph analysis of ethereum erc20 token ecosystem. In *Proceedings of The Web Conference 2020*. 1411–1421.

[89] Scott Chipolina. 2021. *XRP Petition to the White House Hits 35,000 Signatures*. https://decrypt.co/53326/american-public-asks-government-to-deem-xrp-a-currency

[90] Shinyoung Cho, Romain Fontugne, Kenjiro Cho, Alberto Dainotti, and Phillipa Gill. 2019. BGP hijacking classification. In *2019 Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 25–32.

[91] Miyoung Chong. 2019. Discovering fake news embedded in the opposing hashtag activism networks on Twitter:# Gunreformnow vs.# NRA. *Open Information Science* 3, 1 (2019), 137–153.

[92] Robert Cibis. 2020. *CORONA.FILM*. https://www.indiegogo.com/projects/corona-film

[93] Sam Clark and Anna Zaitsev. 2020. Understanding YouTube communities via subscription-based channel embeddings. *arXiv preprint arXiv:2010.09892* (2020).

[94] Steve Clarke. 2019. Conspiracy theories and conspiracy theorizing. In *Conspiracy Theories*. Routledge, 77–92.

[95] Jamie Cleland. 2020. Charismatic leadership in a far-right movement: an analysis of an English defence league message board following the resignation of Tommy Robinson. *Social Identities* 26, 1 (2020), 48–60.

[96] Jay Clython. 2018. *Customer Advisory: Beware Virtual Currency Pump-and-Dump Schemes*. https://news.bitcoin.com/cobinhood-delists-six-tokens-susceptible-to-pump-and-dump-limits-tether-pairs/

[97] cniperbot. 2023. sniperbot. https://github.com/cniperbot/sniperbot.

[98] CNN. 2019. *The flat-Earth conspiracy is spreading around the globe. Does it hide a darker core?* https://edition.cnn.com/2019/11/16/us/flat-earth-conference-conspiracy-theories-scli-intl

[99] CoinDetect. 2018. *CoinDetect*. https://coindetect.org

[100] CoinGecko. 2021. *CoinGecko API*. https://www.coingecko.com/en/api

[101] Coingecko. 2021. *Dogecoin chart*. https://www.coingecko.com/it/monete/dogecoin/usd

[102] CoinGecko. 2023. *CoinGecko.* https://www.coingecko.com

[103] Patrick Thompson CoinGeek. 2021. Solana sees first rug pull: Luna Yield disappears with $6.7M in digital currency. https://coingeek.com/solana-sees-first-rug-pull-luna-yield-disappears-with-6-7m-in-digital-currency/.

[104] CoinMarketCap. 2022. CoinMarketCap. https://coinmarketcap.com/.

[105] Coinmarketcap. 2023. *TrustSwap.* https://coinmarketcap.com/currencies/trustswap/

[106] Yashu Gola Cointelegraph. 2021. Game over! Squid Game-inspired crypto scam collapses as price crashes from $2.8K to zero. https://cointelegraph.com/news/game-over-squid-game-inspired-crypto-scam-collapses-as-price-crashes-from-2-8k-to-zero.

[107] CoinTool. 2023. CoinTool. https://cointool.app/createToken/bsc.

[108] Ben Cox. 2021. *Hunting down the stuck BGP routes.* Retrieved 2021-05-25 from https://blog.benjojo.co.uk/post/bgp-stuck-routes-tcp-zero-window

[109] Rhys Crilley, Marie Gillespie, Bertie Vidgen, and Alistair Willis. 2022. Understanding RT's audiences: Exposure not endorsement for Twitter followers of Russian state-sponsored media. *The International Journal of Press/Politics* 27, 1 (2022), 220–242.

[110] Crypto.com. 2022. Cronos docs. https://cronos.org/docs/getting-started/.

[111] CryptoCompare. 2021. *CC API.* https://min-api.cryptocompare.com

[112] Eustace Cryptus. 2018. *YoBit Inflates PutinCoin in Blatant Pump and Dump Promotion.* https://bitcoinist.com/yobit-inflates-putincoin-in-blatant-pump-and-dump-promotion/

[113] Anthony Cuthbertson. 2021. *Dogecoin: GameStop frenzy takes crypto market over $1 trillion as Reddit stock investors switch to bitcoin rival.* https://www.independent.co.uk/life-style/gadgets-and-tech/dogecoin-price-stock-buy-gamestop-reddit-bitcoin-b1794695.html

[114] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. 2020. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 910–927.

[115] Alberto Dainotti, Karyn Benson, Alistair King, Bradley Huffaker, Eduard Glatz, Xenofontas Dimitropoulos, Philipp Richter, Alessandro Finamore, and Alex C Snoeren. 2016. Lost in space: improving inference of IPv4 address space utilization. *IEEE Journal on Selected Areas in Communications* 34, 6 (2016), 1862–1876.

[116] damartripamungkas. 2022. BOTDEXDAMAR. `https://github.com/damartripamungkas/botdexdamar`.

[117] Michal Mimino Danilak. 2023. *langdetect*. `https://pypi.org/project/langdetect/`

[118] Chris Dannen. 2017. *Introducing Ethereum and solidity*. Vol. 1. Springer.

[119] Daniel De Zeeuw, Sal Hagen, Stijn Peeters, and Emilija Jokubauskaite. 2020. Tracing normiefication: A cross-platform analysis of the QAnon conspiracy theory. *First Monday* (2020).

[120] Geoff Desreumaux. 2018. *Sorry Facebook, Reddit Is Now The Third Most Popular Site In The US*. `https://wersm.com/sorry-facebook-reddit-is-now-the-third-most-popular-site-in-the-us/`

[121] Anirudh Dhawan and Tālis J Putniņš. 2020. A new wolf in town? Pump-and-dump manipulation in cryptocurrency markets. *Pump-and-dump manipulation in cryptocurrency markets (August 10, 2020)* (2020).

[122] Monika Di Angelo and Gernot Salzer. 2021. Identification of token contracts on Ethereum: standard compliance and beyond. *International Journal of Data Science and Analytics* (2021), 1–20.

[123] Cambridge Dictionary. 2023. *Conspiracy theory definition*. `https://dictionary.cambridge.org/dictionary/english/conspiracy-theory`

[124] D Dietrich. 2005. Bogons and bogon filtering. In *33rd meeting of the North American Network Operator's Group (NANOG 33)*.

[125] Christoph Dietzel, Anja Feldmann, and Thomas King. 2016. Blackholing at ixps: On the effectiveness of ddos mitigation in the wild. In *International Conference on Passive and Active Network Measurement*. Springer, 319–332.

[126] Chris HQ Ding, Hongyuan Zha, Xiaofeng He, Parry Husbands, and Horst D Simon. 2004. Link analysis: hubs and authorities on the World Wide Web. *SIAM review* 46, 2 (2004), 256–268.

[127] Morris J Dworkin et al. 2015. SHA-3 standard: Permutation-based hash and extendable-output functions. (2015).

[128] dYdX. 2023. dYdX. `https://dydx.exchange`.

[129] eBay. 2023. *Global rate card*. `https://partnernetwork.ebay.com/our-program/rate-card`

[130] Adam M Enders, Joseph E Uscinski, Michelle I Seelig, Casey A Klofstad, Stefan Wuchty, John R Funchion, Manohar N Murthi, Kamal Premaratne, and Justin Stoler. 2021. The relationship between social media use and beliefs in conspiracy theories and misinformation. *Political behavior* (2021), 1–24.

[131] Kristen Engel, Yiqing Hua, Taixiang Zeng, and Mor Naaman. 2022. Characterizing reddit participation of users who engage in the qanon conspiracy theories. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW1 (2022), 1–22.

[132] EtherDelta. 2023. EtherDelta. `https://etherdelta.com`.

[133] Ethereum. 2022. Contract ABI Specification. `https://docs.soliditylang.org/en/v0.8.13/abi-spec.html`.

[134] Ethereum. 2022. Ethereum Virtual Machine (EVM). `https://ethereum.org/it/developers/docs/evm/`.

[135] ExportData.io. 2021. *Twitter Data Export & Analysis Tool.* `https://www.exportdata.io/trends/`

[136] Vitalik Buterin Fabian Vogelsteller. 2015. EIP-20: Token Standard. `https://eips.ethereum.org/EIPS/eip-20`.

[137] Emily Fales, Lauryn Lintner, Mason Runkel, and Paola Ariza. 2020. The Moon Landing Hoax. (2020).

[138] Gowhar Farooq. 2017. Politics of fake news: How WhatsApp became a potent propaganda tool in India. *Media Watch* 9, 1 (2017), 106–117.

[139] Nick Feamster, Jaeyeon Jung, and Hari Balakrishnan. 2005. An empirical study of" bogon" route advertisements. *ACM SIGCOMM Computer Communication Review* 35, 1 (2005), 63–70.

[140] Philip M Fernbach and Jonathan E Bogard. 2023. Conspiracy Theory as Individual and Group Behavior: Observations from the Flat Earth International Conference. *Topics in Cognitive Science* (2023).

[141] Romain Fontugne, Esteban Bautista, Colin Petrie, Yutaro Nomura, Patrice Abry, Paulo Gonçalves, Kensuke Fukuda, and Emile Aben. 2019. BGP zombies: An analysis of beacons stuck routes. In *International Conference on Passive and Active Network Measurement.* Springer, 197–209.

[142] Center for Applied Internet Data Analysis based at the University of California's San Diego Supercomputer Center. 2021. *AS Relationships.* `https://www.caida.org/catalog/datasets/as-relationships/`

[143] American Registry for Internet Numbers. 2004. *New Statistics Format Available.* Retrieved 2020-05-05 from `https://www.arin.net/vault/announcements/2004/20040108.html`

[144] American Registry for Internet Numbers. 2009. *Extended Allocation and Assignment Report for RIRs.* Retrieved 2020-05-05 from `https://www.arin.net/reference/research/statistics/nro_extended_stats_format.pdf`

[145] Fantom Foundation. 2022. Fantom Whitepaper. `https://fantom.foundation/research/wp_fantom_v1.6.pdf`.

[146] Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* 55, 1 (1997), 119–139.

[147] Laura Frieder and Jonathan Zittrain. 2007. Spam works: Evidence from stock touts and corresponding market activity. *Hastings Comm. & Ent. LJ* 30 (2007), 479.

[148] Michael Fröwis, Andreas Fuchs, and Rainer Böhme. 2019. Detecting token systems on ethereum. In *International conference on financial cryptography and data security*. Springer, 93–112.

[149] Milla Frühling. 2020. *The Conspiracy Empire of Oliver Janich*. https://www.belltower.news/social-media-the-conspiracy-empire-of-oliver-janich-106913/

[150] Vince Fuller, Tony Li, Jessica Yu, and Kannan Varadhan. 1993. *Classless inter-domain routing (CIDR): an address assignment and aggregation strategy*. Technical Report.

[151] Adityawardhan Gaikwad and Sushil Mavale. 2021. The Impact of Cryptocurrency Adoption as a Legal Tender in El Salvador. *International Journal of Engineering and Management Research* 11, 6 (2021), 112–115.

[152] Neil Gandal, JT Hamrick, Tyler Moore, and Tali Oberman. 2018. Price manipulation in the Bitcoin ecosystem. *Journal of Monetary Economics* 95 (2018), 86–96.

[153] Bingyu Gao, Haoyu Wang, Pengcheng Xia, Siwei Wu, Yajin Zhou, Xiapu Luo, and Gareth Tyson. 2020. Tracking counterfeit cryptocurrency end-to-end. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4, 3 (2020), 1–28.

[154] Amanda Garry, Samantha Walther, Rukaya Rukaya, and Ayan Mohammed. 2021. QAnon conspiracy theory: examining its evolution and mechanisms of radicalization. *Journal for Deradicalization* 26 (2021), 152–216.

[155] Chaim Gartenberg. 2021. *XRP Posted Biggest Single-Day Gain in 3 Years in a Coordinated Buying Attack*. https://www.theverge.com/2021/1/28/22254102/robinhood-gamestop-bloc-stock-purchase-amc-reddit-wsb

[156] Mustafa Gatollari. 2021. *Angela White Tweeted About Dogecoin and Its Value Skyrocketed 125 Percent*. https://www.distractify.com/p/angela-white-dogecoin

[157] Huston Geoff. [n. d.]. *Ipv4 stats*. Retrieved 2020-06-29 from https://www.potaroo.net/tools/ipv4/index.html

[158] Huston Geoff. Aug. 2005. *ASN stats*. Retrieved 2020-06-29 from https://www.potaroo.net/ispcol/2005-08/as.pdf

[159] Huston Geoff. Dec. 2003. *IPv4 - How long do we have?* Retrieved 2020-06-29 from https://www.potaroo.net/papers/ipj/2003-v6-n4-ipv4/ipv4.html

[160] Huston Geoff. Dec. 2008. *RFC 5398:Autonomous System (AS) Number Reservation for Documentation Use.* https://tools.ietf.org/html/rfc5398

[161] Huston Geoff. Jul. 2003. *ASN IPs stats.* Retrieved 2020-06-29 from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.5361

[162] Huston Geoff. May. 2021. *The 32-bit AS Number Report.* Retrieved 2021-05-22 from https://www.potaroo.net/tools/asn32/

[163] Huston Geoff. Oct. 2008. *Confronting IPv4 Address Exhaustion.* Retrieved 2020-06-29 from https://www.potaroo.net/ispcol/2008-10/v4depletion.pdf

[164] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th symposium on operating systems principles.* 51–68.

[165] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. 2007. Youtube traffic characterization: a view from the edge. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement.* 15–28.

[166] Vasileios Giotsas, Georgios Smaragdakis, Christoph Dietzel, Philipp Richter, Anja Feldmann, and Arthur Berger. 2017. Inferring BGP blackholing activity in the internet. In *Proceedings of the 2017 Internet Measurement Conference.* 1–14.

[167] Github. 2023. Github. https://github.com/.

[168] Omkar Godbole. 2021. *CXRP Pump Fails to Materialize as Price Crashes 40% From Day's High.* https://www.coindesk.com/xrp-pump-fails-to-materialize-as-price-crashes-40-from-days-high

[169] Omkar Godbole. 2021. *XRP Posted Biggest Single-Day Gain in 3 Years in a Coordinated Buying Attack.* https://www.coindesk.com/xrp-ripple-price-3-year-high-telegram-groups

[170] GoFundMe. 2022. *UPDATE: GoFundMe to refund all Freedom Convoy 2022 donations (2/5/2022).* https://gofundme.com/f/taking-back-our-freedom-convoy-2022

[171] Yashu Gola. 2021. Shiba Inu could surpass Dogecoin after a 700% SHIB price rally in October. https://cointelegraph.com/news/shiba-inu-could-surpass-dogecoin-after-a-700-shib-price-rally-in-october.

[172] Dan Goodin. Apr. 2018. *Suspicious event hijacks Amazon traffic for 2 hours, steals cryptocurrency.* Retrieved 2020-06-29 from https://arstechnica.com/information-technology/2018/04/suspicious-event-hijacks-amazon-traffic-for-2-hours-steals-cryptocurrency/

[173] Google. 2018. *Cloud Natural Language.* https://cloud.google.com/natural-language/

[174] Google. 2018. *The Hunt for 3ve Taking down a major ad fraud operation through industry collaboration.* https://services.google.com/fh/files/blogs/3ve_google_whiteops_whitepaper_final_nov_2018.pdf

[175] Google. 2021. *Google trend.* https://trends.google.com/trends/explore?date=2021-01-20%202021-02-04&geo=US&q=xrp

[176] Todd Gordon. 2022. The Freedom Convoy, the resurgence of the far Right, and the crisis of the petty bourgeoisie. *Studies in Political Economy* 103, 3 (2022), 280–293.

[177] The Graph. 2022. The Graph: APIs for a vibrant decentralized future. https://thegraph.com/en/.

[178] B.A. Green. 2020. *Gamestop Closing 1,000 More Stores By April.* https://www.thathashtagshow.com/2020/12/09/gamestop-closing-1000-more-stores-by-april

[179] Andy Gregory. 2021. *Dogecoin, Elon Musk and the 'joke' cryptocurrency showing the power of memes.* https://www.indy100.com/tech/dogecoin-elon-musk-cryptocurrency-meme-b1797746

[180] The Guardian. 2021. Unmasked: man behind cult set to replace QAnon. https://www.theguardian.com/us-news/2021/mar/20/revealed-man-behind-fast-growing-cult-becoming-the-new-qanon-sabmyk-network.

[181] The Guardian. 2023. *8chan: the far-right website linked to the rise in hate crimes.* https://www.theguardian.com/technology/2019/aug/04/mass-shootings-el-paso-texas-dayton-ohio-8chan-far-right-website

[182] Pedro A Aranda Gutiérrez. 2010. Collateral damage in the last big internet storm. In *2010 Sixth Advanced International Conference on Telecommunications*. IEEE, 468–473.

[183] J. Haas and J. Mitchell. Jul. 2014. *RFC 7300: Reservation of Last Autonomous System (AS) Numbers.* https://tools.ietf.org/html/rfc7300

[184] Martin Haferkorn and Josué Manuel Quintana Diaz. 2014. Seasonality and interconnectivity within cryptocurrencies-an analysis on the basis of bitcoin, litecoin and namecoin. In *International Workshop on Enterprise Applications and Services in the Finance Industry*. Springer, 106–120.

[185] JT Hamrick, Farhang Rouhi, Arghya Mukherjee, Amir Feder, Neil Gandal, Tyler Moore, and Marie Vasek. 2018. The economics of cryptocurrency pump and dump schemes. (2018).

[186] Hans WA Hanley, Deepak Kumar, and Zakir Durumeric. 2022. No calm in the storm: investigating QAnon website relationships. In *Proceedings of the international AAAI conference on Web and social media*, Vol. 16. 299–310.

[187] Layla Harding. 2018. *Yobit Pump and Dump Scheme: Everything you need to know.* `https://coinnounce.com/yobit-pump-and-dump-scheme-everything-you-need-to-know/`

[188] HashDit. 2023. Risk Level Description. `https://hashdit.github.io/hashdit/docs/risk-level-description`.

[189] J. Hawkinson and T. Bates. Mar. 1996. *RFC 1930: Guidelines for creation, selection, and registration of an Autonomous System (AS).* `https://tools.ietf.org/html/rfc1930`

[190] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, Genevieve Bartlett, and Joseph Bannister. 2008. Census and survey of the visible internet. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement.* 169–182.

[191] Urs Hengartner, Sue Moon, Richard Mortier, and Christophe Diot. 2002. Detection and analysis of routing loops in packet traces. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment.* 107–112.

[192] Mohamad Hoseini, Philipe Melo, Fabricio Benevenuto, Anja Feldmann, and Savvas Zannettou. 2023. On the globalization of the QAnon conspiracy theory through Telegram. In *Proceedings of the 15th ACM Web Science Conference 2023.* 75–85.

[193] Packet Clearing House. 2021. *Packet Clearing House.* `https://www.pch.net/`

[194] IANA. 2020. *Special-Purpose Autonomous System (AS) Numbers Created.* `https://www.iana.org/assignments/iana-as-numbers-special-registry/iana-as-numbers-special-registry.xhtml`

[195] IANA. Aug. 2015. *Special-Purpose Autonomous System (AS) Numbers Created.* Retrieved 2020-05-17 from `https://www.iana.org/assignments/iana-as-numbers-special-registry/iana-as-numbers-special-registry.xhtml`

[196] IDEX. 2023. IDEX. `https://idex.io/`.

[197] Mustafa Ilhan. 2021. *A project that keeps history of trending topics on Twitter.* `https://tt-history.appspot.com/`

[198] Roland Imhoff and Pia Lamberty. 2020. A bioweapon or a hoax? The link between distinct conspiracy beliefs about the coronavirus disease (COVID-19) outbreak and pandemic behavior. *Social Psychological and Personality Science* 11, 8 (2020), 1110–1118.

[199] Vincenzo Imperati, Massimo La Morgia, Alessandro Mei, Alberto Maria Mongardini, and Francesco Sassi. 2023. The Conspiracy Money Machine: Uncovering Telegram's Conspiracy Channels and their Profit Model. *arXiv preprint arXiv:2310.15977* (2023).

[200] Discord Inc. 2018. *Discord.* `https://discordapp.com/`

[201] Reddit Inc. 2021. *Reddit.* `https://www.reddit.com`

[202] Infura. 2022. Infura. `https://infura.io/`.

[203] Walter Isaacson. 2014. *The innovators: How a group of inventors, hackers, geniuses and geeks created the digital revolution.* Simon and Schuster.

[204] Quentin Jacquemart. 2015. *Towards uncovering BGP hijacking attacks.* Ph. D. Dissertation. Télécom ParisTech.

[205] April Jensen. 2021. *God Bless America, Free my J6er.* `https://www.givesendgo.com/G26FY`

[206] Guolin Jiang, Paul G Mahoney, and Jianping Mei. 2005. Market manipulation: A comprehensive study of stock pools. *Journal of Financial Economics* 77, 1 (2005), 147–170.

[207] Yong Jiang, Björn Pehrson, and Runtong Zhang. 2003. Measuring and evaluating the current BGP policy model. In *Proceedings of the Eighth IEEE Symposium on Computers and Communications. ISCC 2003.* IEEE, 1167–1171.

[208] Rebecca Henschke Joel Gunter and Astudestra Ajengrastri. 2023. Global network of sadistic monkey torture exposed by BBC. https://www.bbc.com/news/world-65951188.

[209] Don Johnson, Alfred Menezes, and Scott Vanstone. 2001. The elliptic curve digital signature algorithm (ECDSA). *International journal of information security* 1, 1 (2001), 36–63.

[210] Sommer B Johnson. 2021. *Stand with Paul.* `fundly.com/stand-4-paul#gallery/3`

[211] Demitri Kalogeropoulos. 2020. *Why GameStop Stock Dropped 52% in 2019.* `https://www.fool.com/investing/2020/01/07/why-gamestop-stock-dropped-52-in-2019.aspx`

[212] Josh Kamps and Bennett Kleinberg. 2018. To the moon: defining and detecting cryptocurrency pump-and-dumps. *Crime Science* 7, 1 (2018), 18.

[213] Daniel Karrenberg, Gerard Ross, Paul Wilson, and Leslie Nobile. 2001. Development of the Regional Internet Registry System. *The Internet Protocol Journal* 4, 4 (Dec. 2001), 17–29. `https://www.nro.net/development-of-the-regional-internet-registry-system/`

[214] Arjun Kharpal. 2021. *Reddit frenzy pumps up Dogecoin, a cryptocurrency started as a joke.* `https://cnb.cx/3j10ZOv`

[215] Asim Ijaz Khwaja and Atif Mian. 2005. Unchecked intermediaries: Price manipulation in an emerging stock market. *Journal of Financial Economics* 78, 1 (2005), 203–241.

[216] S. Kirkpatrick, M. Stahl, and M. Recker. Jul. 1990. *RFC 1166: Internet numbers.* https://tools.ietf.org/html/rfc1166

[217] Jon M Kleinberg. 1999. Hubs, authorities, and communities. *ACM computing surveys (CSUR)* 31, 4es (1999), 5–es.

[218] Maria Konte, Roberto Perdisci, and Nick Feamster. 2015. ASwatch: An AS Reputation System to Expose Bulletproof Hosting ASes. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication - SIGCOMM '15.* ACM Press, London, United Kingdom, 625–638. https://doi.org/10.1145/2785956.2787494

[219] P.C. Kotsias. 2020. pcko1/etherscan-python. https://github.com/pcko1/etherscan-python. https://doi.org/10.5281/zenodo.4306855

[220] P.C. Kotsias. 2021. pcko1/bscscan-python. https://github.com/pcko1/bscscan-python. https://doi.org/10.5281/zenodo.4781726

[221] Peter M Krafft, Nicolás Della Penna, and Alex Sandy Pentland. 2018. An experimental study of cryptocurrency market dynamics. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems.* ACM, 605.

[222] David B Kramer. 2005. The Way It Is and the Way It Should Be: Liability Under § 10 (b) of the Exchange Act and Rule 10b-5 Thereunder for Making False and Misleading Statements as Part of a Scheme to" Pump and Dump" a Stock. *University of Miami Business Law Review* 13, 2 (2005), 243.

[223] Abhinandan Kulal. 2021. Followness of Altcoins in the Dominance of Bitcoin: A Phase Analysis. *Macro Management & Public Policies* 3, 3 (2021).

[224] W. Kumari, R. Bush, H. Schiller, and K. Patel. Aug. 2015. *RFC7607:Codification of AS 0 Processing.* https://tools.ietf.org/html/rfc7607

[225] James F Kurose and Keith W Ross. 2012. Computer Networking: A Top-Down Approach . 6th. *Harlow, UK: Pearson Education Ltd* (2012).

[226] Albert S Kyle and S Viswanathan. 2008. How to define illegal price manipulation. *American Economic Review* 98, 2 (2008), 274–79.

[227] Massimo La Morgia, Alessandro Mei, and Alberto Maria Mongardini. 2023. It's a Trap! Detection and Analysis of Fake Channels on Telegram. In *2023 IEEE International Conference on Web Services (ICWS).* IEEE.

[228] Massimo La Morgia, Alessandro Mei, and Alberto Maria Mongardini. 2023. TGDataset. https://zenodo.org/record/7640712#.Y-9PjNLMKXI.

[229] Massimo La Morgia, Alessandro Mei, and Alberto Maria Mongardini. 2023. TGDataset: a Collection of Over One Hundred Thousand Telegram Channels. *arXiv preprint arXiv:2303.05345* (2023).

[230] Massimo La Morgia, Alessandro Mei, Alberto Maria Mongardini, and Jie Wu. 2021. Uncovering the Dark Side of Telegram: Fakes, Clones, Scams, and Conspiracy Movements. *arXiv preprint arXiv:2111.13530* (2021).

[231] M. La Morgia, A. Mei, S. Raponi, and J. Stefa. 2018. Time-Zone Geolocation of Crowds in the Dark Web. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. 445–455. `https://doi.org/10.1109/ICDCS.2018.00051`

[232] Massimo La Morgia, Alessandro Mei, Francesco Sassi, and Julinda Stefa. 2020. Pump and Dumps in the Bitcoin Era: Real Time Detection of Cryptocurrency Market Manipulations. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 1–9.

[233] Massimo La Morgia, Alessandro Mei, Francesco Sassi, and Julinda Stefa. 2023. The doge of wall street: Analysis and detection of pump and dump cryptocurrency manipulations. *ACM Transactions on Internet Technology* 23, 1 (2023), 1–28.

[234] LACNIC. 2020. *LACNIC ftp*. Retrieved 2020-05-17 from `https://ftp.lacnic.net/pub/stats/lacnic/`

[235] lacnog. Oct. 2013. *Secuestro de ruta*. Retrieved 2020-06-29 from `https://mail.lacnic.net/pipermail/lacnog/2013-October/002622.html`

[236] Solidity Lang. 2022. Contract ABI Specification. `https://docs.soliditylang.org/en/v0.5.3/abi-spec.html`.

[237] Mark Ledwich and Anna Zaitsev. 2020. Algorithmic extremism: Examining YouTube's rabbit hole of radicalization. *First Monday* (2020).

[238] Barry M Leiner, Vinton G Cerf, David D Clark, Robert E Kahn, Leonard Kleinrock, Daniel C Lynch, Jon Postel, Larry G Roberts, and Stephen Wolff. 2009. A brief history of the Internet. *ACM SIGCOMM computer communication review* 39, 5 (2009), 22–31.

[239] Tao Li, Donghwa Shin, and Baolian Wang. 2018. Cryptocurrency pump-and-dump schemes. *Available at SSRN* (2018).

[240] Tao Li, Donghwa Shin, and Baolian Wang. 2020. Cryptocurrency pump-and-dump schemes. *Available at SSRN 3267041* (2020).

[241] Xigao Li, Anurag Yepuri, and Nick Nikiforakis. 2023. Double and Nothing: Understanding and Detecting Cryptocurrency Giveaway Scams. In *Network and Distributed Systems Security (NDSS) Symposium.*

[242] Yun Li. 2020. *Melvin Capital, hedge fund targeted by Reddit board, closes out of GameStop short position.* `https://cnb.cx/2YibdR1` Accessed on 2021-01-28..

[243] Bailey Lipschultz. 2021. *WallStreetBets Briefly Goes Dark After Fueling GameStop's Surge.* `https://www.bloomberg.com/news/articles/2021-`

01-26/gamestop-extends-gains-premarket-as-eye-popping-surge-continues Accessed on 2021-01-28.

[244] Deborah E Lipstadt. 2012. *Denying the Holocaust: The growing assault on truth and memory.* Simon and Schuster.

[245] Defi Llama. 2022. Defi Llama. https://defillama.com/.

[246] Telegram LLC. 2018. *Telegram - a new era of messaging.* https://telegram.org/

[247] Andrew W Lo and Jasmina Hasanhodzic. 2010. *The evolution of technical analysis: financial prediction from Babylonian tablets to Bloomberg terminals.* Vol. 96. John Wiley & Sons.

[248] Austin P Logan, Phillip M LaCasse, and Brian J Lunday. 2023. Social network analysis of Twitter interactions: a directed multilayer network approach. *Social Network Analysis and Mining* 13, 1 (2023), 65.

[249] Elizabeth Lopatt. 2021. *SEC says third-largest cryptocurrency was sold all wrong.* https://www.theverge.com/2020/12/22/22196064/ripple-sec-cryptocurrency-security-currency-xrp

[250] K. Lougheed and Y. Rekhter. Jun. 1989. *RFC 1105: A Border Gateway Protocol (BGP).* https://tools.ietf.org/html/rfc1105

[251] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas, and KC Claffy. 2013. AS relationships, customer cones, and validation. In *Proceedings of the 2013 conference on Internet measurement conference.* 243–256.

[252] Youcef Maouchi, Lanouar Charfeddine, and Ghassen El Montasser. 2022. Understanding digital bubbles amidst the COVID-19 pandemic: Evidence from DeFi and NFTs. *Finance Research Letters* 47 (2022), 102584.

[253] Bruno Mazorra, Victor Adan, and Vanesa Daza. 2022. Do Not Rug on Me: Leveraging Machine Learning Techniques for Automated Scam Detection. *Mathematics* 10, 6 (2022), 949.

[254] Team McAfee. 2021. *Help Team McAfee Give Back to SaveTheChildren Venezuela Thanksgiving Feast4Food.* https://fundrazr.com/Team_McAfee

[255] Mdex. 2022. Mdex. https://mdex.com.

[256] Theo Meder et al. 2021. Online coping with the first wave: Covid humor and rumor on Dutch social media (March–July 2020). *Folklore: Electronic Journal of Folklore* 82 (2021), 135–158.

[257] Evgeny Medvedev and the D5 team. 2018. Ethereum ETL. https://github.com/blockchain-etl/ethereum-etl.

[258] Jianping Mei, Guojun Wu, and Chunsheng Zhou. 2004. Behavior based manipulation: theory and prosecution evidence. *Available at SSRN 457880* (2004).

[259] Amin Mekacher and Antonis Papasavva. 2022. "I Can't Keep It Up." A Dataset from the Defunct Voat. co News Aggregator. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 16. 1302–1311.

[260] Xiaoqiao Meng, Zhiguo Xu, Beichuan Zhang, Geoff Huston, Songwu Lu, and Lixia Zhang. 2005. IPv4 address allocation and the BGP routing table evolution. *ACM SIGCOMM Computer Communication Review* 35, 1 (2005), 71–80.

[261] MetaMask. 2022. A crypto wallet & gateway to blockchain apps. `https://metamask.io/`.

[262] Sarah E Michigan. 2021. Sneaker bots & Botnets: malicious digital tools that harm rather than help e-commerce. *Rutgers Bus. LJ* 17 (2021), 169.

[263] J. Mitchell. Jul. 2013. *RFC6996: Autonomous System (AS) Reservation for Private Use.* `https://tools.ietf.org/html/rfc6996`

[264] Shaheed N Mohammed. 2019. Conspiracy theories and flat-earth videos on YouTube. *The Journal of Social Media in Society* 8, 2 (2019), 84–102.

[265] Bhabendu Kumar Mohanta, Soumyashree S Panda, and Debasish Jena. 2018. An overview of smart contract and use cases in blockchain technology. In *2018 9th international conference on computing, communication and networking technologies (ICCCNT).* IEEE, 1–4.

[266] Mohd Razman Achmadi Muhammad and Noor Nirwandy. 2021. A study on Donald Trump Twitter remark: a case study on the attack of Capitol Hill. *Journal of Media and Information Warfare (JMIW)* 14, 2 (2021), 75–104.

[267] MyEtherWallet. 2022. MyEtherWallet. `https://www.myetherwallet.com/`.

[268] Shalini Nagarajan. 2021. *Dogecoin, a digital token that started as a joke, spikes 140% after traders in a crypto-themed Reddit forum trigger Wall Street Bets copycat rally.* `https://markets.businessinsider.com/currencies/news/dogecoin-price-crypto-reddit-traders-satoshistreetbets-trigger-copycat-wsb-rally-2021-1-1030015774`

[269] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review* (2008).

[270] Nanog. Aug. 2017. *Hijack Factories: AS203418, AS205944, and AS203040.* Retrieved 2020-06-29 from `https://mailman.nanog.org/pipermail/nanog/2017-August/191858.html`

[271] Nanog. Jan. 2018. *Spectrum prefix hijacks.* Retrieved 2020-06-29 from `https://mailman.nanog.org/pipermail/nanog/2018-January/193573.html`

[272] Inc. NANOG. 2021. *NANOG.* https://www.nanog.org/resources/nanog-mailing-list/nanog-mailing-lists/

[273] RIPE NCC. 2021. *Routing Information Service (RIS).* Retrieved 2021-05-25 from https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris

[274] Eugenio Nerio Nemmi, Francesco Sassi, Massimo La Morgia, Cecilia Testart, Alessandro Mei, and Alberto Dainotti. 2021. The parallel lives of autonomous systems: ASN allocations vs. BGP. In *Proceedings of the 21st ACM Internet Measurement Conference.* 593–611.

[275] CBS News. 2020. *Twitter bans Zero Hedge after it posts coronavirus conspiracy theory.* https://www.cbsnews.com/news/twitter-bans-zero-hedge-coronavirus-conspiracy-theory/

[276] NBC News. 2020. *Google bans website ZeroHedge from its ad platform over comments on protest articles.* https://www.nbcnews.com/tech/tech-news/google-bans-two-websites-its-ad-platform-over-protest-articles-n1231176

[277] NBC News. 2021. *Amazon removes QAnon merchandise from its marketplace.* https://www.nbcnews.com/business/business-news/amazon-removes-qanon-merchandise-its-marketplace-n1253937

[278] Lawrence Lewitinn Nikhilesh De, Zack Seward. 2021. *Coinbase to Suspend XRP Trading Following SEC Suit Against Ripple.* https://www.coindesk.com/coinbase-suspends-xrp-trading

[279] Leonardo Nizzoli, Serena Tardelli, Marco Avvenuti, Stefano Cresci, Maurizio Tesconi, and Emilio Ferrara. 2020. Charting the landscape of online cryptocurrency manipulation. *IEEE Access* 8 (2020), 113230–113245.

[280] Number Resource Organization (NRO). 2021. *RIR Comparative Policy Overview.* https://www.nro.net/policy/regional/rir-comparative-policy-overview/

[281] Ricardo Oliveira, Dan Pei, Walter Willinger, Beichuan Zhang, and Lixia Zhang. 2009. The (in) completeness of the observed internet AS-level structure. *IEEE/ACM Transactions on Networking* 18, 1 (2009), 109–122.

[282] OpenZeppelin. 2023. OpenZeppelin. https://forum.openzeppelin.com/.

[283] Chiara Orsini, Alistair King, Danilo Giordano, Vasileios Giotsas, and Alberto Dainotti. 2016. BGPStream: a software framework for live and historical BGP data analysis. In *Proceedings of the 2016 Internet Measurement Conference.* 429–444.

[284] Caitlin Ostroff. 2021. *What Is Dogecoin, How to Say It, and Why It's No Longer a Joke.* https://www.wsj.com/articles/what-is-dogecoin-how-to-say-it-and-why-its-no-longer-a-joke-thanks-elon-11612820776

[285] Jedidajah Otte. 2021. *'Sending a message': GameStop investors on why they bought shares.* `https://www.theguardian.com/business/2021/jan/28/sending-a-message-gamestop-investors-on-why-they-bought-shares`

[286] PADL. 2018. *PADL - Crypto Pump and Dump Groups -Signals - List.* `https://padl.mine.nu`

[287] Antonis Papasavva, Max Aliapoulios, Cameron Ballard, Emiliano De Cristofaro, Gianluca Stringhini, Savvas Zannettou, and Jeremy Blackburn. 2021. The gospel according to Q: Understanding the QAnon conspiracy from the perspective of canonical information. In *AAAI International Conference on Web and Social Media.*

[288] Antonis Papasavva, Jeremy Blackburn, Gianluca Stringhini, Savvas Zannettou, and Emiliano De Cristofaro. 2021. "Is it a qoincidence?": An exploratory study of QAnon on Voat. In *Proceedings of the Web Conference 2021*. 460–471.

[289] Antonis Papasavva, Savvas Zannettou, Emiliano De Cristofaro, Gianluca Stringhini, and Jeremy Blackburn. 2020. Raiders of the lost kek: 3.5 years of augmented 4chan posts from the politically incorrect board. In *Proceedings of the international AAAI conference on web and social media*, Vol. 14. 885–894.

[290] Pujan Paudel, Jeremy Blackburn, Emiliano De Cristofaro, Savvas Zannettou, and Gianluca Stringhini. 2021. Soros, child sacrifices, and 5G: understanding the spread of conspiracy theories on web communities. *arXiv preprint arXiv:2111.02187* (2021).

[291] Paypal. 2023. *Donate Button.* `https://www.paypal.com/donate/buttons`

[292] Sebastian Pellejero and Marco Quiroz-Gutierrez. 2020. *BlackBerry, AMC and Other Reddit YOLO Favorites That Aren't GameStop.* `https://www.wsj.com/articles/blackberry-amc-and-other-reddit-yolo-favorites-that-arent-gamestop-11611681716` Accessed on 2021-01-28..

[293] Shruti Phadke, Mattia Samory, and Tanushree Mitra. 2021. What makes people join conspiracy communities? role of social factors in conspiracy engagement. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW3 (2021), 1–30.

[294] Shruti Phadke, Mattia Samory, and Tanushree Mitra. 2022. Pathways through conspiracy: the evolution of conspiracy radicalization through engagement in online conspiracy discussions. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 16. 770–781.

[295] David Araújo Pinheiro, Mariana Gomes Leitão De Araújo, Keilla Barbosa De Souza, BDS Campos, EM De Oliveira, RSM Lima, GA Ferreira, ACA De Freitas, CB Toledo, GB De Souza, et al. 2020. Sharing fake news about health in the cross-platform messaging app WhatsApp during the COVID-19 pandemic: A pilot study. *Int. J. Sci. Res. Manag* 8 (2020), 403–410.

[296] Jason Carve Piper Merriam. 2022. Web3.py. https://web3py.readthedocs.io/en/stable/.

[297] Amy Cheng The Washington Post. 2021. 'Squid Game'-inspired cryptocurrency that soared by 23 million percent now worthless after apparent scam. https://www.washingtonpost.com/world/2021/11/02/squid-game-crypto-rug-pull/.

[298] J. Postel and J. Vernon. Jan. 1983. *RFC 820: Assigned Numbers.* https://tools.ietf.org/html/rfc820

[299] Nicholas Proferes, Naiyan Jones, Sarah Gilbert, Casey Fiesler, and Michael Zimmer. 2021. Studying reddit: A systematic overview of disciplines, approaches, methods, and ethics. *Social Media+ Society* 7, 2 (2021), 20563051211019004.

[300] The Spamhaus Project. 2021. *Spamhaus.* https://www.spamhaus.org/

[301] Valerio Puggioni. 2022. Crypto rug pulls: What is a rug pull in crypto and 6 ways to spot it. https://cointelegraph.com/explained/crypto-rug-pulls-what-is-a-rug-pull-in-crypto-and-6-ways-to-spot-it.

[302] PumpBot. 2022. Sniper Bot Crypto: Chain sniper- The all in one Sniper Bot, DEX Bot, Pinksale Bot. https://pump-bot.com/crypto-bots/sniper-bot-frontrunner-chainsniper-dexbot.

[303] pushshift.io. 2021. *Learn about Big Data ans Social Media Ingest and Analysis.* https://redditsearch.io/

[304] Kaihua Qin, Liyi Zhou, and Arthur Gervais. 2022. Quantifying blockchain extractable value: How dark is the forest?. In *2022 IEEE Symposium on Security and Privacy (SP).* IEEE, 198–214.

[305] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. 2004. Defining and identifying communities in networks. *Proceedings of the national academy of sciences* 101, 9 (2004), 2658–2663.

[306] Jodi Reffitt. 2021. *Reffitt Family fund.* https://www.givesendgo.com/G23DE

[307] Y. Rekhter, T. Li, and S. Hares. January 2006. *RFC 4271:A Border Gateway Protocol 4 (BGP-4).* https://www.rfc-editor.org/info/rfc4271

[308] Philipp Richter, Mark Allman, Randy Bush, and Vern Paxson. 2015. A Primer on IPv4 Scarcity. *ACM SIGCOMM Computer Communication Review* 45, 2 (April 2015), 21–31. https://doi.org/10.1145/2766330.2766335

[309] Philipp Richter, Georgios Smaragdakis, David Plonka, and Arthur Berger. 2016. Beyond Counting: New Perspectives on the Active IPv4 Address Space. In *Proceedings of the 2016 Internet Measurement Conference.* ACM, Santa Monica California USA, 135–149. https://doi.org/10.1145/2987443.2987473

[310] RIPE. 2020. *Legacy Resources.* https://www.ripe.net/manage-ips-and-asns/legacy-resources/erx

[311] RIPE. 2020. *RIPE ERX Resources.* https://www.ripe.net/manage-ips-and-asns/legacy-resources/erx/erx-transfer-of-as-number-registrations

[312] RIPE. 2020. *RIPE ftp.* Retrieved 2020-05-17 from https://ftp.ripe.net/pub/stats/ripencc/

[313] Team Rocket, Maofan Yin, Kevin Sekniqi, Robbert van Renesse, and Emin Gün Sirer. 2019. Scalable and probabilistic leaderless BFT consensus through metastability. *arXiv preprint arXiv:1906.08936* (2019).

[314] Eric C. Rosen. Oct. 1982. *RFC 827: Exterior Gateway Protocol (EGP).* https://tools.ietf.org/html/rfc827

[315] Matthew Roughan, Walter Willinger, Olaf Maennel, Debbie Perouli, and Randy Bush. 2011. 10 lessons from 10 years of measuring and modeling the internet's autonomous systems. *IEEE Journal on Selected Areas in Communications* 29, 9 (2011), 1810–1821.

[316] RouteViews. 2021. *RouteViews Routing Table Archive.* Retrieved 2021-05-22 from http://www.routeviews.org

[317] Brian Rudick. 2021. DeFi Summer 2.0: Don't Call it a Comeback. https://www.gsr.io/insights/chart-of-the-week-defi-summer-2-0-dont-call-it-a-comeback/.

[318] RugDoc. 2023. *RugDoc API.* https://rugdoc.io/

[319] saantiaguilera. 2022. AX-50 Liquidity Sniper. https://github.com/saantiaguilera/liquidity-sniper.

[320] S Rasoul Safavian and David Landgrebe. 1991. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics* 21, 3 (1991), 660–674.

[321] Tim Sauer. [n. d.]. Hacking the Internet: How BGP paves the way for attackers. ([n. d.]).

[322] Elizabeth Schumacher. 2023. *Disclose.TV: English disinformation made in Germany.* https://www.dw.com/en/disclosetv-english-disinformation-made-in-germany/a-60694332

[323] Jon Seidel. 2023. Feds' child porn sweep on Telegram app leads to arrest of Chicago man, more than a dozen others. https://chicago.suntimes.com/2023/2/2/23582748/child-porn-telegram-app-operation-swipe-left-homeland-security-stauffer-adam-hageman-verastigui.

[324] Kevin Sekniqi, Daniel Laine, Stephen Buttolph, and Emin G¨un Sirer. 2020. *Avalanche Platform.* Vol. 1. online.

[325] Selenium. 2023. *Selenium.* https://www.selenium.dev

[326] Arijit Sarkar The New York State Senate. 2021. The Secret To The Success Of 'Squid Game,' Explained. https://www.forbes.com/sites/danidiplacido/2021/10/06/the-secret-to-squid-games-success-explained/?sh=83a56ea224cf.

[327] Arijit Sarkar The New York State Senate. 2022. NY Sen. Thomas proposes to criminalize rug pulls and other crypto frauds. https://www.nysenate.gov/newsroom/in-the-news/kevin-thomas/ny-sen-thomas-proposes-criminalize-rug-pulls-and-other-crypto.

[328] Pavlos Sermpezis, Vasileios Kotronis, Alberto Dainotti, and Xenofontas Dimitropoulos. 2018. A survey among network operators on BGP prefix hijacking. *ACM SIGCOMM Computer Communication Review* 48, 1 (2018), 64–69.

[329] Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas Dimitropoulos, Danilo Cicalese, Alistair King, and Alberto Dainotti. 2018. ARTEMIS: Neutralizing BGP hijacking within a minute. *IEEE/ACM Transactions on Networking* 26, 6 (2018), 2471–2486.

[330] Michael Siering. 2013. All Pump, No Dump? The Impact Of Internet Deception On Stock Markets.. In *ECIS*. 115.

[331] Georgos Siganos and Michalis Faloutsos. 2004. Analyzing BGP policies: Methodology and tool. In *IEEE INFOCOM 2004*, Vol. 3. IEEE, 1640–1651.

[332] Vasilios A Siris and Fotini Papagalou. 2004. Application of anomaly detection algorithms for detecting SYN flooding attacks. In *IEEE Global Telecommunications Conference, 2004. GLOBECOM'04.*, Vol. 4. IEEE, 2050–2054.

[333] Rebecca Skippage. 2022. Telegram: Where women's nudes are shared without consent. https://www.bbc.com/news/world-60303769.

[334] TUF sniperbot. 2023. TUF sniperbot. https://tufsniperbot.com/.

[335] Joe Sommerlad. 2021. Sabmyk Network: Founder of bizarre new religion targeting QAnon believers 'unmasked' by Hope Not Hate. https://www.independent.co.uk/news/world/europe/sabmyk-network-qanon-conspiracy-theories-b1820639.html.

[336] Anusha Sriraman, Kevin RB Butler, Patrick D McDaniel, and Padma Raghavan. 2007. Analysis of the ipv4 address space delegation structure. In *2007 12th IEEE Symposium on Computers and Communications*. IEEE, 501–508.

[337] Ethereum StackExchange. 2023. Ethereum StackExchange. https://ethereum.stackexchange.com/.

[338] Carl Stempel, Thomas Hargrove, and Guido H Stempel III. 2007. Media use, social structure, and belief in 9/11 conspiracy theories. *Journalism & Mass Communication Quarterly* 84, 2 (2007), 353–372.

[339] Eva Su. 2020. *Digital Assets and SEC Regulation.* Congressional Research Service.

[340] S Shyam Sundar, Maria D Molina, and Eugene Cho. 2021. Seeing is believing: Is video modality more powerful in spreading fake news via online messaging apps? *Journal of Computer-Mediated Communication* 26, 6 (2021), 301–319.

[341] Maria Susana. 2020. *Crimes Against Humanity.* indiegogo.com/projects/crimes-against-humanity?create_edit=true

[342] Martin Holst Swende and Marius van der Wijden. 2022. EIP-3155: EVM trace specification. https://eips.ethereum.org/EIPS/eip-3155.

[343] SystemsLab-Sapienza. 2021. *Pump and dump dataset.* https://github.com/SystemsLab-Sapienza/pump-and-dump-dataset

[344] SystemsLab-Sapienza. 2021. *XRP crowd pump chat.* https://github.com/SystemsLab-Sapienza/gme-pump-xrp-telegram

[345] SystemsLab-Sapienza. 2023. Regex for token smart contract mechanisms that hinder sniper bots. https://doi.org/10.5281/zenodo.7604918.

[346] Amelia Tait. 2016. *Pizzagate: How a 4Chan conspiracy went mainstream.* https://www.newstatesman.com/science-tech/2016/12/pizzagate-how-4chan-conspiracy-went-mainstream

[347] Cristina Tardáguila, Fabricio Benevenuto, and Pablo Ortellado. 2018. Fake News Is Poisoning Brazilian Politics. WhatsApp Can Stop It. *International New York Times* (2018), NA–NA.

[348] Spamhaus Team. 2020. *Suspicious network resurrections.* Retrieved 2021-05-25 from https://www.spamhaus.org/news/article/802/suspicious-network-resurrections

[349] Telega.io. 2023. *Telegram ads platform: trusted and effective channels and bots.* https://telega.io

[350] Telegram. 2023. *CoinGecko & CoinMarketCap Listing Alerts Premium.* https://t.me/CMC_CG_listing_alerts

[351] Telegram. 2023. *Telegram Ad Platform.* https://promote.telegram.org

[352] Telegram. 2023. *Telegram FAQ.* https://telegram.org/faq#q-what-is-telegram-what-do-i-do-here

[353] Cecilia Testart, Philipp Richter, Alistair King, Alberto Dainotti, and David Clark. 2019. Profiling BGP Serial Hijackers: Capturing Persistent Misbehavior in the Global Routing Table. In *Proceedings of the Internet Measurement Conference.* 420–434.

[354] TokenByGen. 2023. TokenByGen. https://tokensbygen.com/.

[355] Andree Toonk. Dec. 2017. *Popular destinations rerouted to russia.* Retrieved 2020-06-29 from https://bgpmon.net/popular-destinations-rerouted-to-russia/

[356] Christof Ferreira Torres, Ramiro Camino, et al. 2021. Frontrunner jones and the raiders of the dark forest: An empirical study of frontrunning on the ethereum blockchain. In *30th USENIX Security Symposium (USENIX Security 21)*. 1343–1359.

[357] Christof Ferreira Torres, Mathis Steichen, et al. 2019. The art of the scam: Demystifying honeypots in ethereum smart contracts. In *28th USENIX Security Symposium (USENIX Security 19)*. 1591–1607.

[358] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific reports* 9, 1 (2019), 5233.

[359] Trading-Tiger. 2022. Pancakeswap BSC Sniper Bot. `https://github.com/Trading-Tiger/Pancakeswap_BSC_Sniper_Bot`.

[360] Milo Trujillo, Maurício Gruppi, Cody Buntain, and Benjamin D Horne. 2020. What is bitchute? characterizing the. In *Proceedings of the 31st ACM conference on hypertext and social media*. 139–140.

[361] Cynthia Turcotte. 2018. *Yobit Crypto Exchange Announces Public Pump and Dump Scheme.* `https://www.livebitcoinnews.com/yobit-crypto-exchange-announces-public-pump-and-dump-scheme/`

[362] Marc Tuters, Emilija Jokubauskaitė, and Daniel Bach. 2018. Post-truth protest: How 4chan cooked up the Pizzagate bullshit. *M/c Journal* 21, 3 (2018).

[363] UniCrypt. 2023. Liquidity Lockers - UniCrypt. `https://docs.unicrypt.network/liquidity-lockers/general-concept`.

[364] Uniswap. 2022. Uniswap v2 License. `https://github.com/Uniswap/v2-core/blob/master/LICENSE`.

[365] Uniswap. 2022. Uniswap v3 License. `https://github.com/Uniswap/v3-core/blob/main/LICENSE`.

[366] Ravi Vaidyanathan, Abhrajit Ghosh, Yukiko Sawaya, and Ayumu Kubota. 2012. On the use of enhanced bogon lists (EBLs) to detect malicious traffic. In *2012 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 1–6.

[367] Jan-Willem Van Prooijen and Karen M Douglas. 2017. Conspiracy theories as part of history: The role of societal crisis situations. *Memory studies* 10, 3 (2017), 323–333.

[368] Marie Vasek and Tyler Moore. 2019. Analyzing the bitcoin ponzi scheme ecosystem. In *Financial Cryptography and Data Security: FC 2018 International Workshops, BITCOIN, VOTING, and WTSC, Nieuwpoort, Curaçao, March 2, 2018, Revised Selected Papers 22*. Springer, 101–112.

[369] The Verge. 2018. Hackers emptied Ethereum wallets by breaking the basic infrastructure of the internet. https://www.theverge.com/2018/4/24/17275982/myetherwallet-hack-bgp-dns-hijacking-stolen-ethereum.

[370] The Verge. 2023. *Reddit has banned the QAnon conspiracy subreddit r/GreatAwakening.* https://www.theverge.com/2018/9/12/17851938/reddit-qanon-ban-conspiracy-subreddit-greatawakening

[371] Pierre-Antoine Vervier, Quentin Jacquemart, Johann Schlamp, Olivier Thonnard, Georg Carle, Guillaume Urvoy-Keller, Ernst Biersack, and Marc Dacier. 2014. Malicious BGP hijacks: appearances can be deceiving. In *2014 IEEE International Conference on Communications (ICC).* IEEE, 884–889.

[372] VICE. 2022. *Germany's 'Biggest QAnon Mouthpiece' Arrested in the Philippine.* https://www.vice.com/en/article/n7zexk/oliver-janich-germany-philippines

[373] Friedhelm Victor and Tanja Hagemann. 2019. Cryptocurrency Pump and Dump Schemes: Quantification and Detection. In *2019 International Conference on Data Mining Workshops (ICDMW).* IEEE, 244–251.

[374] Friedhelm Victor and Bianca Katharina Lüders. 2019. Measuring ethereum-based erc20 token networks. In *International Conference on Financial Cryptography and Data Security.* Springer, 113–129.

[375] Q. Vohra and E. Chen. Dec. 2012. *RFC6793: BGP Support for Four-Octet Autonomous System (AS) Number Space.* https://tools.ietf.org/html/rfc6793

[376] VSQUARE. 2023. *TELEGRAM, THE FREE ZONE FOR DISINFORMATION AND CONSPIRACIES.* https://vsquare.org/telegram-the-free-zone-for-disinformation-and-conspiracies/

[377] W. 2021. *THE BIG RESET.* kickstarter.com/projects/thebigreset/the-big-reset

[378] Charlotte Wagnsson. 2023. The paperboys of Russian messaging: RT/Sputnik audiences as vehicles for malign information influence. *Information, communication & society* 26, 9 (2023), 1849–1867.

[379] Trust Wallet. 2022. Trust Wallet. https://trustwallet.com/.

[380] Nick Paton Walsh and Salma Abdelaziz. 2018. US assault rifles are being sold on the Telegram messaging app in Syria. https://edition.cnn.com/2018/02/20/middleeast/us-weapons-telegram-syria-intl/index.html.

[381] Kaili Wang, Qinchen Wang, and Dan Boneh. 2022. ERC-20R and ERC-721R: Reversible Transactions on Ethereum. *arXiv preprint arXiv:2208.00543* (2022).

[382] Janith Weerasinghe, Bailey Flanigan, Aviel Stein, Damon McCoy, and Rachel Greenstadt. 2020. The pod people: Understanding manipulation of social media popularity via reciprocity abuse. In *Proceedings of The Web Conference 2020*. 1874–1884.

[383] Amy Whitaker. 2019. Art and blockchain: A primer, history, and taxonomy of blockchain use cases in the arts. *Artivate* 8, 2 (2019), 21–46.

[384] Wikipedia. Jun. 2020. *Dot-com Bubble.* Retrieved 2020-06-29 from `https://en.wikipedia.org/wiki/Dot-com_bubble`

[385] Per-Olof H Wikström and Noémie Bouhana. 2016. Analyzing radicalization and terrorism: A situational action theory. *The handbook of the criminology of terrorism* (2016), 175–186.

[386] Rene Wilhelm and Henk Uijterwaal. Oct. 2005. *ASN Missing In Action.* Retrieved 2020-06-29 from `https://www.ripe.net/publications/docs/ripe-353`

[387] Skip Willman. 1998. Traversing the Fantasies of the JFK Assassination: Conspiracy and Contingency in Don Delillo's" Libra". *Contemporary Literature* 39, 3 (1998), 405–433.

[388] Gavin Wood. 2018. Ethereum yellow paper: A formal specification of Ethereum, a programmable blockchain. 2018. *URL https://github. com/ethereum/yellowpaper* (2018).

[389] Turner Wright. 2021. *DOGE is 'not a bad look' for crypto users, says Mark Cuban as price surges another 8%.* `https://cointelegraph.com/news/doge-is-not-a-bad-look-for-crypto-users-says-mark-cuban-as-price-surges-another-8`

[390] Pengcheng Xia, Haoyu Wang, Bingyu Gao, Weihang Su, Zhou Yu, Xiapu Luo, Chao Zhang, Xusheng Xiao, and Guoai Xu. 2021. Trade or Trick? Detecting and Characterizing Scam Tokens on Uniswap Decentralized Exchange. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 5, 3 (2021), 1–26.

[391] Jiahua Xu and Benjamin Livshits. 2019. The anatomy of a cryptocurrency pump-and-dump scheme. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*. 1609–1625.

[392] Jiahua Xu, Krzysztof Paruch, Simon Cousaert, and Yebo Feng. 2021. Sok: Decentralized exchanges (dex) with automated market maker (amm) protocols. *Comput. Surveys* (2021).

[393] Ahmet S Yayla and Anne Speckhard. 2017. Telegram: The mighty application that ISIS loves. *International Center for the Study of Violent Extremism* 9 (2017).

[394] Youtube. 2023. *Monetization Policies.* `https://www.youtube.com/intl/en_us/howyoutubeworks/policies/monetization-policies/`

[395] Jing Zeng and Mike S Schäfer. 2021. Conceptualizing “dark platforms”. Covid-19-related conspiracy theories on 8kun and Gab. *Digital Journalism* 9, 9 (2021), 1321–1343.

[396] Dirk A Zetzsche, Douglas W Arner, and Ross P Buckley. 2020. Decentralized finance. *Journal of Financial Regulation* 6, 2 (2020), 172–203.

[397] Peng Zhang, Douglas C Schmidt, Jules White, and Gunther Lenz. 2018. Blockchain technology use cases in healthcare. In *Advances in computers*. Vol. 111. Elsevier, 1–41.

[398] Liyi Zhou, Kaihua Qin, Christof Ferreira Torres, Duc V Le, and Arthur Gervais. 2021. High-frequency trading on decentralized on-chain exchanges. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 428–445.