DeFiIntel: A Dataset Bridging On-Chain and Off-Chain Data for DeFi Token Scam Investigation

Iori Suzuki*,Yin Minn Pa Pa†, Nguyen Thi Van Anh†, Katsunari Yoshioka*†
*Graduate School of Environment and Information Sciences, Yokohama National University
Email: suzuki-iori-zv@ynu.jp, yoshioka@ynu.ac.jp
†Institute of Advanced Sciences, Yokohama National University
Email: yinminn-papa-jp@ynu.ac.jp, nguyen-anh-xd@ynu.ac.jp

***Abstract*—Decentralized Finance (DeFi) token scams have become one of the most prevalent forms of fraud in Web-3 technology, generating approximately $241.6 million in illicit revenue in 2023 [1]. Detecting these scams requires analyzing both on-chain data, such as transaction records on the blockchain, and off-chain data, such as websites related to the DeFi token project and associated social media accounts. Relying solely on one type of data may fail to capture the full context of fraudulent activities. While on-chain data is publicly accessible due to the transparency inherent in blockchain technology, off-chain data often disappears alongside DeFi scam campaigns, making it difficult for the security community to study these scams. To address this challenge, we propose a dataset comprising more than 550 thousand archived web and social media data as off-chain data, in addition to on-chain data related to 32,144 DeFi tokens deployed on Ethereum blockchain from September 24, 2024 to January 14, 2025. This dataset aims to support the security community in studying and detecting DeFi token scams. To illustrate its utility, our case studies demonstrated the potential of the dataset in identifying patterns and behaviors associated with scam tokens. These findings highlight the dataset's capability to provide insights into fraudulent activities and support further research in developing effective detection mechanisms.**

I. Introduction

Decentralized Finance (DeFi) token projects have rapidly gained traction in the Web-3 ecosystem, providing innovative financial services such as lending, borrowing, and staking without relying on traditional intermediaries like banks [2]. These projects are designed to attract users by offering rewards, letting people take part in important decisions through special voting tokens (governance tokens), or turning real-world assets into digital tokens that are easier to trade and access [3].

Despite these legitimate use cases, the DeFi ecosystem has also become a hotspot for fraudulent activities. Scams such as rug pulls—where developers withdraw liquidity and abandon projects—have caused significant financial losses [4]. Similarly, pump-and-dump schemes artificially inflate token prices before a sudden sell-off, leaving investors with worthless assets [5]. Some projects even create tokens as a means to facilitate money laundering, complicating regulatory oversight [6]. According to the 2024 Crypto Crime Report by Chainalysis, approximately 24.4% of all new tokens launched on Ethereum in 2023 were suspected to be scams, particularly pump-and-dump schemes, which generated illicit profits of $241.6 million.

This dual nature of DeFi token projects—as platforms for financial innovation and as targets for scams—underscores the need for effective detection mechanisms. Such measures are essential to protect investors, maintain trust, and ensure the sustainable growth of decentralized financial ecosystems.

Effectively detecting scams within DeFi token projects requires a comprehensive approach that leverages both on-chain data, such as blockchain transaction records, and off-chain data, including information from associated websites and social media accounts. On-chain data provides a transparent and immutable record of transactions, but it often lacks the context needed to understand the intent or legitimacy behind those activities. Reports and studies have emphasized the importance of combining these data sources, as relying solely on on-chain data is insufficient for early fraud detection [7]. Off-chain sources often reveal fraudulent activities that on-chain data cannot capture, especially when scammers leverage short-lived websites or fake social media profiles to promote their schemes. Despite the recognized need for integrating these data types, the absence of a database that combines both on-chain and off-chain data poses a significant challenge for the security community.

The REKT Database [8] is a curated repository documenting over 3,000 crypto scams and hacks, providing details like contract addresses, exploit summaries, and references to off-chain URLs. However, it does not archive the content of these URLs completely, leading to the loss of critical off-chain data when the URLs become inaccessible, limiting its long-term utility for comprehensive scam analysis.

In contrast, CryptoScamDB [9], an open-source initiative by MyCrypto, focuses on documenting cryptocurrency scams across various platforms, with a particular emphasis on off-chain data such as phishing websites, scam domains, and malicious social media campaigns. While CryptoScamDB excels at identifying fraudulent off-chain domains and URLs, it does not directly integrate on-chain data, such as token transactions or contract behaviors, which limits its capacity

Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb) 2025
28 February 2025, San Diego, CA, USA
ISBN 979-8-9894372-9-0
https://dx.doi.org/10.14722/madweb.2025.23019
www.ndss-symposium.org

for a holistic analysis of scams that bridge on-chain and off-chain activities. Furthermore, CryptoScamDB does not archive the content of websites, only recording URLs, which may become inaccessible over time. This limitation reduces its utility for long-term scam analysis and comprehensive contextual research.

Palaiokrassas et al. [10] developed a dataset using on-chain transaction data from 23 DeFi protocols across 12 blockchains to detect fraudulent activities. While comprehensive in capturing blockchain interactions, the dataset lacks off-chain data, such as website and social media content, which limits its ability to provide contextual insights into scams that rely heavily on external manipulation or promotion. Clough et al. [11] built a dataset focusing on pump-and-dump schemes, integrating on-chain data, such as transaction records and token interactions, with off-chain data primarily from Telegram channels. While this dataset provides valuable insights into the lifecycle and impact of pump-and-dump schemes across 765 coins, it is limited in scope, as it heavily relies on Telegram for off-chain data and lacks the inclusion of other off-chain sources, such as websites or broader social media platforms.

To address the limitations of existing datasets, we developed a dataset to detect and analyze DeFi token scams by integrating on-chain and off-chain data. Our method exhaustively monitors the Ethereum blockchain, analyzing all deployed tokens during the observation period and collecting smart contract information using RPC endpoints [12] and tools like Etherscan [13]. URLs embedded in smart contract source code are systematically extracted to identify relevant off-chain data, such as website content and social media activity from platforms like X (Twitter) and Telegram. This off-chain data is archived in WARC format to preserve ephemeral and dynamic information, enabling a thorough analysis of both on-chain activity and off-chain promotional strategies.

Our dataset comprises more than 550 thousand archived web and social media data as off-chain data, in addition to on-chain data related to 32,144 DeFi tokens deployed on Ethereum blockchain from September 24, 2024 to January 14, 2025. Case studies confirmed the dataset's potential to reveal reliance on off-chain promotional efforts, and the dynamics of fraudulent behavior in the decentralized finance ecosystem.

Following are the contributions of this study:

- **Integrated Data Collection System:** We present an integrated data collection system that systematically collects and integrates both on-chain data (e.g., source files, metadata) and off-chain data (e.g., URLs, website content, and social media activity) to provide DeFi token activities. The system also archives off-chain data, addressing the challenge of ephemeral information disappearing over time (Section IV).
- **Dataset for Scam Detection and Research:** This study introduces a dataset that bridges the gap between blockchain-based records and off-chain promotional activities. The dataset includes on-chain data such as DeFi token related information and its related off-chain data such as contract source code, web archives, enabling the security community to analyze fraudulent behaviors effectively. We are going to share this dataset to verified researchers. By making this dataset accessible, the work supports scam detection and future research efforts (Section V).
- **Case Studies Validating Dataset Utility:** Through case studies, we showcase the dataset's effectiveness in detecting fraudulent activities, such as inconsistencies in contract addresses advertised on websites and the use of domains that impersonate legitimate projects. These examples illustrate how our dataset uncovers scam dynamics and provides actionable insights for the security community (Section VI).

II. Preliminaries

A. DeFi Token

A DeFi token is a type of cryptocurrency used within decentralized finance (DeFi) platforms to facilitate financial services like lending, staking, trading, and governance. These tokens, typically based on the Ethereum Request for Comment 20 (ERC-20) standard on Ethereum, are managed by smart contracts and are interoperable with wallets and exchanges.

B. Blockchain Explorer

A blockchain explorer is a web-based tool that retrieves, aggregates, and displays information about transactions, blocks, and smart contracts within a blockchain network. For the Ethereum blockchain, Etherscan [13] is one of the most widely used blockchain explorers. Etherscan supports contract verification. Verification typically requires access to the contract's source code and constructor arguments, which are usually known only to the developers.

C. Verified Smart Contract

When deploying a smart contract on Ethereum, developers write the source code in a high-level language like Solidity, compile it into bytecode, and deploy it on the blockchain. A verified contract is one where the uploaded source code and the compiled bytecode match. Once verified, the source code is publicly available, allowing users to compare it with the deployed bytecode. This transparency helps with bug discovery, security auditing, and provides users a clearer understanding of the contract's functionality.

D. Understanding Embedded URLs in Smart Contract

The source code of verified smart contracts often contains embedded URLs, which act as a bridge between the on-chain functionality and the off-chain context of DeFi token projects. Developers include these URLs to establish connections between the technical implementation of the token and its broader ecosystem. These URLs typically serve 2 main purposes: 1. They link to official project websites, which serve as the central hub for information about the token, including its purpose, goals, purchase instructions, and the official contract address. 2. They direct users to social media platforms, such as

Telegram, Twitter (X), or Discord, where developers share announcements, updates on token performance, and community discussions. These platforms occasionally host scam reports and feedback, offering critical insights into the legitimacy of the project. 3. Some URLs point to external data sources, such as function documentation or third-party services, which provide insights into the token's operations.

These project-specific URLs are often included at the beginning of the source code, typically as comments, to provide additional context without interfering with the smart contract's execution.

E. On-Chain and Off-Chain Data

On-chain data refers to information recorded directly on the blockchain, which is immutable and publicly accessible. Examples include smart contract address, bytecode, and contract deployment timestamp.

Off-chain data encompasses dynamic and temporary information external to the blockchain, often used in token promotion and project activities. Examples include project-specific URLs embedded in smart contract source code, website content, and social media activity (e.g., posts and group messages).

III. Related works

A. DeFi and Scams Detection

Xia et al. [14] and Mazorra et al. [15] conducted studies on scam token detection and rug-pull scams within the decentralized finance (DeFi) ecosystem, particularly focusing on Uniswap. They introduced a hybrid approach combining guilt-by-association heuristics and machine-learning techniques to detect scam tokens on Uniswap V2. They analyzed over 20 million transactions and identified more than 10,000 scam tokens, which accounted for nearly 50% of all tokens listed on Uniswap. Their findings highlighted the prevalence of rug-pull scams, where attackers remove liquidity after luring victims into buying worthless tokens. They also uncovered the use of collusion addresses and smart contract backdoors, demonstrating profits exceeding $16 million from around 40,000 victims.

Mazorra et al. [15] expanded upon Xia's work by constructing a dataset of 27,588 tokens and improving scam detection accuracy through advanced machine-learning techniques. By introducing features such as the Herfindahl-Hirschman Index (HHI) to analyze token distribution and clustering coefficients for transaction graphs, they demonstrated the ability to predict rug-pull scams before they occur. Their models achieved a remarkable accuracy of 0.9936, with a recall of 0.9540 and a precision of 0.9838. They also criticized the reliance on lock contracts like Unicrypt, showing that 90% of tokens employing such mechanisms eventually became scams.

These studies primarily focus on detecting scam tokens using on-chain data, such as transaction behaviors, token distributions, and liquidity patterns. However, they do not incorporate off-chain information, such as social media discussions or project announcements, which may provide additional context for identifying scams at earlier stages.

To emphasize scam research within the context of social media, Xu and Livshits analyzed 412 pump-and-dump activities orchestrated through messages in Telegram channels, leveraging both price-related information and non-price-related factors such as coin listing status [16]. Their study primarily focuses on understanding pump-and-dump schemes using a dataset of confirmed scams but does not address monitoring emerging contracts for newer scams.

B. Datasets for Blockchain Analysis

Datasets focusing on blockchain and DeFi data have been proposed for scam detection [10], [11]. Palaiokrassas et al. [10] introduced a dataset leveraging on-chain transaction data from 23 DeFi protocols across 12 blockchains to detect fraudulent activities. While it extensively covers blockchain interactions, it omits off-chain data. Similarly, [11] developed a dataset targeting pump-and-dump schemes, integrating on-chain data such as transaction records and token interactions with Telegram channel messages. These datasets represent snapshots of data from specific points in time and do not include live updates. For ongoing, continuously updated databases, projects like REKT Database and CryptoScamDB are notable examples.

The REKT Database [8] is a manually documented database with over 3,000 crypto scams, DeFi hacks, exchange exploits, and phishing attacks. It includes key details such as contract addresses, exploit summaries, technical breakdowns, and references to off-chain URLs like project websites and social media accounts. While this database serves as a valuable source for understanding crypto-related scams, it has significant limitations: it does not archive the content of off-chain URLs. As a result, critical contextual information is lost when these URLs become inaccessible, reducing its long-term utility for comprehensive scam analysis and security research.

CryptoScamDB [9] is an open-source project by MyCrypto aimed at tracking and documenting cryptocurrency scams. It primarily focuses on off-chain data, such as phishing websites, scam domains, and malicious social media campaigns, shedding light on fraudulent activities that may not be detected through on-chain analysis. The database is publicly accessible, supports API integration, and allows community contributions, ensuring timely updates on scams. However, it lacks integration with on-chain data, such as blockchain transactions or smart contract interactions, and does not archive the content of scam-related websites, only recording URLs that could later become inaccessible. These limitations make it less suitable for long-term and comprehensive scam analysis, even though it plays a key role in identifying and preventing crypto-related fraud.

IV. Data collection methodology

This section details the methodology for dataset collection. Our objective is to design a systematic and automated framework for DeFi smart contract and token data collection, encompassing both on-chain and off-chain sources. The on-chain data includes immutable contract information retrieved

from the Ethereum blockchain, while the off-chain data covers temporary and dynamic information such as promotional campaign messages on social media and webpage content.

Our data collection process, illustrated in Figure 1, consists of five steps. First, smart contracts are extracted from the Ethereum blockchain (Step 1). Next, additional details such as source code and metadata are gathered using the Etherscan API (Step 2). The retrieved smart contracts are then validated to identify the corresponding tokens (Step 3). For tokens identified as DeFi tokens, the process proceeds to collect off-chain data. This involves retrieving token-related URLs from the smart contract source code (Step 4), which typically leads to the token's official website, social media accounts, or other associated platforms. Finally, the content from these URLs, including web pages, DeFi token promotional materials, and social media posts, is retrieved and stored (Step 5).

An overview of the collection process is depicted in Fig. 1.

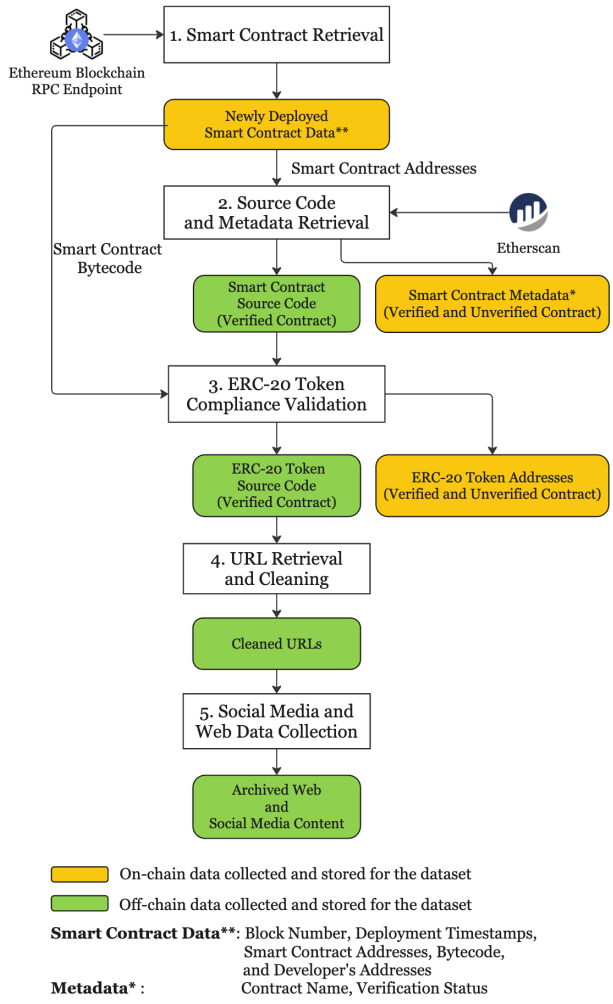

Fig. 1. Data Collection Process

Step 1: Smart Contract Retrieval

Newly deployed smart contracts, defined as contracts created through specific transactions on the Ethereum blockchain, are identified by analyzing transactions in newly created blocks. A contract creation transaction is recognized by the absence of a recipient address (since no existing account is being interacted with) and the presence of bytecode in the transaction's data field. These transactions are retrieved from the Ethereum blockchain using public RPC endpoints [12], allowing real-time detection of newly deployed contracts.

The collected data includes key details about each smart contract to provide a comprehensive view of its deployment and origins. Specifically, it includes the **block number**, which identifies the blockchain block where the contract was deployed. This block contains a timestamp, enabling the determination of the **deployment timestamp**, which records the exact date and time of the contract's activation. The **smart contract address** is the unique identifier assigned to the contract, facilitating interactions with it. The **bytecode** represents the compiled code executed by the Ethereum Virtual Machine (EVM), defining the contract's functionality. Additionally, the **developer's address**, which is the origin of the contract creation transaction, offers insights into its provenance. Collectively, these elements enable a comprehensive analysis of smart contracts, from their origins to their functionality.

Step 2: Source Code and Metadata Retrieval

For each smart contract collected in Step 1, additional on-chain information is obtained using the Etherscan API [17]. Developers often upload the source code of their smart contracts to Etherscan, allowing these contracts to be classified as verified contracts. Verified contracts provide access to their source code, enhancing transparency and enabling deeper analysis. In addition to the source code, metadata such as the contract name and verification status (i.e., whether or not the source code has been uploaded by the developer) is also retrieved from Etherscan. This metadata serves as valuable contextual information about the smart contract.

Our dataset incorporates metadata for both verified and non-verified contracts to facilitate a comprehensive analysis. For non-verified contracts, the smart contract's metadata is stored. For verified contracts, the source code, along with associated metadata, is included in the dataset. This dual approach ensures that the dataset provides a robust foundation for analyzing smart contracts, irrespective of their verification status.

Step 3: ERC-20 Token Compliance Validation

In this step, we validate whether a given smart contract corresponds to an ERC-20 token (DeFi token) or not. A two-fold method is employed based on the verification status of the smart contract.

For verified contracts obtained in Step 2, the source code is examined to identify the presence of required ERC-20 standard functions, such as `transfer`, `approve`, and `balanceOf`. These functions are essential components of the ERC-20 specification and indicate compliance with the standard.

For unverified contracts, where the source code is unavailable, the bytecode collected in Step 1 is analyzed

for specific function signatures. These include `18160ddd` (representing `totalSupply`) and `70a08231` (representing `balanceOf`), which are derived from the Ethereum Application Binary Interface (ABI) encoding standard [18]. The detection of these function signatures within the bytecode serves as an indicator of ERC-20 functionality.

This dual approach ensures that both verified and unverified contracts are systematically analyzed for their potential relation to ERC-20 tokens, providing a comprehensive validation framework.

Step 4: URL Retrieval and Cleaning

As explained in the preliminary section, "Understanding Embedded URLs in Smart Contracts" II-D, the source code of verified smart contracts often includes project-specific URLs, typically found at the beginning of the source file and often added as comments. These URLs serve as critical off-chain links for understanding token projects. To create a focused dataset for analysis, we adopt a structured approach for URL retrieval and cleaning.

All URLs embedded in the source code collected in Step 3 are extracted. A distinction is made between: - **Project-Specific URLs**, such as links to official websites and social media platforms, which are retained for further analysis. - **General-Purpose URLs**, such as links to development guides or libraries (e.g., `soliditylang.org`, `ethereum.org`), which are excluded as they do not provide meaningful insights into the token project.

This filtering ensures that the dataset highlights URLs directly relevant to token projects, enabling an understanding of their off-chain activities and potential security risks.

Step 5: Social Media and Web Data Collection

The URLs retrieved in Step 4 serve as the foundational input for this process. Project-specific websites are archived to preserve key information, such as token descriptions, purchase instructions, and contract addresses. These websites provide critical references for understanding token characteristics and are stored in the Web ARChive (WARC) format, ensuring long-term accessibility and reproducibility. Web pages were accessed at two depth levels: Depth 1, where only the first URL was accessed, and Depth 2, where secondary URLs linked from the initial page were also archived. To handle dynamic content, tools such as `pywb` [19] and `Playwright` [20] are used. These tools enable the archiving of JavaScript-rendered pages that traditional methods like `wget` cannot process effectively. Importantly, project-specific websites are often deleted or altered after scams conclude, making real-time archiving essential to retain this volatile data.

Similarly, data from social media platforms, including Telegram, Twitter (X), and Discord, is collected and archived. These platforms host valuable updates from developers, community discussions, and, in some cases, scam reports from users. Like project websites, social media data is often removed or altered following the conclusion of scams, further emphasizing the need for timely data collection. Social media platforms provide crucial insights into the off-chain activities of DeFi token projects and the dynamics of their associated communities. For collecting this data, `Tweepy` [21] is employed to extract tweets, while `Telethon` [22] is used for scraping group chats and announcements from Telegram.

For certain Telegram groups, where direct API access is insufficient, the `DrissionPage` [23] web automation tool is utilized to automate the process of joining groups and accessing content. This combination of tools ensures comprehensive and efficient collection of both static and dynamic data from websites and social media platforms before they are permanently removed or altered.

V. Results

This section presents the results of our data collection process, which combines on-chain and off-chain data for DeFi tokens. Using the methodology described in Section IV, we followed five key steps: retrieving smart contracts, collecting source code and metadata, validating ERC-20 compliance, extracting Project-Specific URLs, and archiving off-chain data. All the results from Steps 1 to 4 are based on experiments conducted between September 24, 2024, and January 14, 2025 (JST), while Step 5 covers data collection from November 12, 2024, to January 14, 2025 (JST). Although the results presented here are limited to these periods, the data collection process is ongoing, and the dataset continues to grow. Table I summarizes the outputs of each step, providing a clear overview of the collected data.

Step 1: Smart Contract Retrieval

In Step 1, we collected 137,111 smart contracts deployed on the Ethereum blockchain from September 24, 2024, to January 14, 2025. The average of newly deployed smart contracts is 1,504 contracts per day. The data collected, as detailed in Table I, includes the block number, contract address, creator address, deployment timestamp, and bytecode. These foundational elements provide essential insights into the life-cycle and origin of DeFi projects. The large number of contracts reflects the high activity in the Ethereum ecosystem during the study period, forming a substantial dataset for subsequent analysis of legitimate and fraudulent activities.

Step 2: Source Code and Metadata Retrieval

Building on the contracts identified in Step 1, Step 2 retrieved source code and metadata using the Etherscan API. This step collected source codes and metadata for 42,785 verified contracts and metadata only for 94,326 unverified contracts, as shown in Table II. Metadata fields include contract names and verification status, where verified contracts have names and a "True" verification status, while unverified contracts have "None" as the name and "False" for verification status. Verified contracts provide transparency and enable deeper analysis, while the large number of unverified contracts underscores the need for greater transparency in the DeFi ecosystem.

TABLE I
OUTPUT ITEMS FOR EACH PROCESS

Step	Process	Process Output	Output Items	Description
1.	Smart Contract Retrieval	Smart Contract Basic Information	Block Number	The blockchain block where the contract was deployed.
			Contract Address	The address of the contract.
			Creator Address	The wallet address of the contract creator.
			Creation Date	The timestamp of the contract deployment.
			Bytecode	The bytecode of the contract.
2.	Source Code and Metadata Retrieval	Smart Contract Metadata	Contract Name	The name of the contract.
			Verification Status	True or False status of contract verification.
		Smart Contract Source Code	Source Code	The source code of the contract.
3.	ERC-20 Token Compliance Validation	ERC-20 Token Source Code	ERC-20 Source Code	The source code of the ERC-20 token contract.
		ERC-20 Token Address	ERC-20 Token Address	The address of the ERC-20 token contract.
4.	URL Retrieval and Cleaning	Cleaned URLs	Project-Specific URLs	URLs targeted for archiving in Step 5.
5.	Social Media and Web Data Collection	Archived Web and Social Media Content	Web Archive	Website Archive (WARC) itself and Website Information
			X (Twitter) Post Archive	X (Twitter) post's tweet_id, text post content, status, created_at, author_id, metrics, url, media_url.
			X (Twitter) Metadata	X account's user_id, name, username, status, creation_date, description, urls, media, verified.
			Telegram Message Archive	Telegram groups messages related to the token.
			Telegram Metadata	Telegram account's telegram_username, title, type, telegram id (10 digit integer), creation date, chat_id.

Step 3: ERC-20 Token Compliance Validation

In Step 3, we validated each contract's compliance with the ERC-20 standard, which defines core DeFi token functionality. Using source code analysis for verified contracts and bytecode analysis for unverified contracts, we identified 32,144 verified ERC-20 tokens and 8,823 unverified ERC-20 tokens, as shown in Table II. The number of source files for verified ERC-20 tokens is 248,393, also detailed in Table II.

TABLE II
RESULTS OF EACH STEP OF THE DATA COLLECTION PROCESS

Step	Process	Result	Count
1.	Smart Contract Retrieval	Contracts	: 137,111
2.	Souce Code and Metadata Retrieval	Verified Contracts	: 42,785
		Source Files of Verified Contracts	: 322,356
3.	ERC-20 Token Compliance Validation	Verified ERC-20 Tokens	: 32,144
		Unverified ERC-20 Tokens	: 8,823
		ERC-20 Tokens Source Files	: 248,393
4.	URL Retrieval and Cleaning	Cleaned URLs	: 30,716
5.	Web Archive and Social Media Content	Web Archive Data	: 14,879
		X Post and Metadata	: 2,059
		Telegram Message and Metadata	: 539,325

Step 4: URL Extraction and Cleaning

Step 4 involved extracting and cleaning URLs embedded within the source code of verified contracts. These URLs often link to project websites, social media accounts, or other off-chain resources, providing crucial context for each token. From 18,145 verified contracts that include URLs in it, 30,716 project-specific URLs are extracted. A breakdown of the types of extracted URLs is provided in Table III, highlighting platforms like Twitter (X) and Telegram as the most popular mediums for advertising and community engagement. Telegram URLs are public username URLs (Example: `t.me/<username>`) and Chat Invite URLs (Example: `t.me/+<hash>`). Other URLs in Table III include token project-specific website URLs.

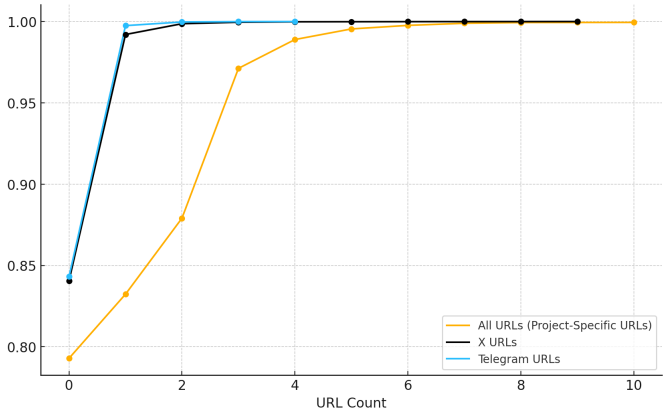

Fig. 2. CDF Graph of URL Count Per Verified Token Address

The Figure 2 presents the cumulative distribution function (CDF) of ERC-20 tokens based on the number of project-specific URLs included in their source code, which link to resources such as official websites, Telegram groups, or Twitter pages. Each line in the graph represents a different category of URLs. The orange line represents the overall distribution of tokens based on the total count of project-specific URLs. The black line indicates tokens containing URLs linking specifically to "X" project pages, while the blue

line represents tokens containing URLs linking specifically to Telegram project pages. For all ERC-20 tokens, represented by the orange line, nearly 80% of tokens do not include any project-specific URLs in their source code (URL count = 0). By the time the URL count reaches 1, less than 83% of tokens have either 0 or 1 URL, implying that only about 3% of tokens include exactly 1 URL. Tokens with multiple URLs are even rarer, highlighting that most ERC-20 tokens lack detailed project-specific metadata in their source code. This suggests limited transparency or incomplete information for many tokens. The black line, representing "X" URLs, shows that approximately 85% of tokens lack any URLs linking to "X" project pages (URL count = 0). At URL count = 1, there is only a minimal increase, indicating that fewer than 2% of tokens include exactly one "X" URL. Beyond URL count = 1, the curve flattens, confirming that tokens with more than one "X" URL are extremely rare. Similarly, the blue line, representing Telegram URLs, shows a similar pattern. Around 85% of tokens lack Telegram project page URLs, and fewer than 2% include exactly one Telegram URL. The curve remains flat for URL counts greater than 1, suggesting that multiple Telegram URLs are also exceptionally uncommon. In summary, the vast majority of ERC-20 tokens do not include Project-Specific URLs in their source code, and the inclusion of URLs linking to X or Telegram project pages is even rarer. Tokens with exactly 1 URL make up a small fraction, and those with multiple URLs are exceptionally rare. This analysis underscores the general lack of project-specific metadata in the source code of most ERC-20 tokens, potentially limiting their transparency and accessibility.

TABLE III
BREAKDOWN OF PROJECT-SPECIFIC URLS

The Number of Project-Specific URLs	Count
X URLs	9,582
Post(Tweet) URLs	2,011
Profile Page URLs	7,571
Telegram URLs	10,582
Public Username URLs	10,529
Chat Invite URLs	53
Web Archive URLs	10,556
Medium URLs	198
Discord URLs	69
YouTube URLs	94
Reddit URLs	52
Facebook URLs	19
Other URLs	10,124
Total	30,716

Step 5: Social Media and Web Data Collection

In Step 5, social media and web data collection was conducted between November 12, 2024, and January 14, 2025, as summarized in Table IV. The process focused on archiving web pages and collecting social media content linked to DeFi tokens. The web pages are archived only one time. This effort resulted in 5,721 archived pages, of which 2,092 were accessible, and 9,158 additional pages, with 5,509 accessible pages. These archives provide a comprehensive capture of off-chain promotional and informational content.

TABLE IV
BREAKDOWN OF ARCHIVE

Archive Type	Collection Period	Breakdown	Count
Web Archive	2024-11-12 to 2025-01-14	Depth=1	Total: 5,721 Accessible: 2,092
		Depth=2	Total: 9,158 Accessible: 5,509
X (Twitter)	2024-12-09 to 2025-01-14	Metadata	1,789
		Post Archive	270
Telegram	2024-12-20 to 2025-01-14	Metadata	4,052
		Message Archive	535,273

Fig. 3. Token Project's Specific Website

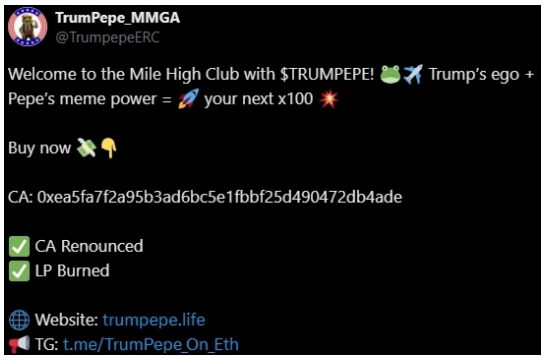

Fig. 4. X Post Advertising Trumpepe DeFi Token

Social media data was gathered from various platforms, with a particular emphasis on Twitter (X) and Telegram due to their prominent presence in the dataset. The Twitter dataset includes 1,789 metadata records and 270 post archives, containing details like user information (e.g., user ID, display name, username, account status, account creation date, description, URLs, and media data) and post content (e.g., tweet ID, text, status, creation date, author's ID, metrics, and shared media URLs). Similarly, the Telegram data includes 4,052 metadata records and 535,273 messages, featuring details such as Telegram username, Telegram ID, creation date, chat title,

and chat ID. Each message record includes attributes like the message date, ID, sender name, content, and shared media.

Figures 3 and 4 show examples of archived off-chain data. Figure 3 presents a screenshot of an archived website promoting a DeFi token, highlighting key promotional messages and structural design. Figure 4 displays a sample X post from the archive, showcasing metrics such as engagement (likes and reposts), as well as the accompanying content and media. These figures emphasize the diversity of data collected and its utility in analyzing off-chain promotional strategies.

VI. Case Studies

In this section, we present case studies demonstrating how our dataset can be leveraged to enhance the understanding of DeFi tokens and detect potential scam projects. Through various analytical approaches, we identified multiple real-time tokens exhibiting indicators of fraudulent activity. We mask the last four characters before the dot and top-level domain (TLD) in domain names explained in this section by replacing them with '****' to anonymize sensitive information. For example, "example.com" becomes "exa****.com".

A. Analysis of Contract Address (CA) Consistency

1) Objective: Token developers or promoters often prominently display the contract address on promotional or community websites as a key identifier for their tokens. This analysis aims to assess the consistency between contract addresses retrieved from the blockchain and those listed on associated websites. Such consistency is a vital indicator of the reliability and authenticity of the projects. In contrast, discrepancies, such as a token's source code containing a URL that promotes a different token linked to another address, may signal potential fraudulent activity.

2) Analysis methodology: We analyzed the consistency of contract addresses (CA) between project web pages and the Ethereum blockchain. First, from the collected data, we extract the Project-Specific URLs from the source code of each verified contract. Using these URLs, we access and scrape the associated web pages. We then verify whether the contract address on the blockchain matches the address found on the page. If they match, we can confirm that the web page is associated with the authentic contract.

3) Findings: We perform this analysis on 25,695 Project-Specific URLs, extracted from the Verified Contracts collected between November 12, 2024 and December 31, 2024.

- **Match CAs: 336 / 25,695** URLs were found to have consistent contract addresses, with **426** from web pages and **25** from Telegram.
- **Not Contain CAs: 8,872 / 25,695** (**2,960** from web pages, **2,439** from X, and **3,473** from Telegram) URLs did not contain CA information on the accessed page.
- **Inaccessible URLs: 16,487 / 25,695** URLs were inaccessible and thus, could not be verified.

The high proportion of URLs that either do not contain CA or are inaccessible underscores potential systemic challenges in data availability and transparency within blockchain projects. This finding reveals that only a small fraction of URLs from DeFi projects contain CA matching the original contract, limiting the number of authentic projects. This creates a significant vulnerability, allowing scammers to easily set up fraudulent project websites that link to counterfeit tokens.

A mismatch in CAs is a strong indicator of scam tokens. One common sign of fraudulent tokens is when the CA listed on a webpage differs from the one deployed on the blockchain. Oftentimes, a fraudulent contract still includes a link to a legitimate project in the source code, to mislead users. We investigated several cases where mismatched CAs were observed.

Marutaro. We identified a contract created on October 1, 2024 at 18:30:59, with source code associated with the Marutaro coin from `marutar****.xyz`. Upon investigating the Telegram messages and website content, we found that the CA listed on both the website and in the Telegram groups is identical.

The `marutar****.xyz` domain and Telegram group were created around September 2024. The CAs found on the website and in the Telegram groups point to a legitimate project, with price fluctuations indicating active transactions over the past three months since September 30, 2024, demonstrating that Marutaro is a project with an engaged community.

The CA from the collected contract differs from the one listed on the project sites, indicating fraud. This is confirmed by a DEXTools score [24] of 1/99, indicating a very unsafe contract. The fraudulent contract shows only a few buy transactions around October 2024, with just 17 holders compared to 651 holders in the legitimate token. We suspect this purchase was the result of a scam, involving a user misled into making a purchase.

WuTensor. A contract purportedly for WuTensor, created on October 1, 2024 at 17:40:23, shows a mismatch compared to the one published through X. Additionally, there was a warning message on X stating, *"Fake CAs are starting to appear"*. At the time of the investigation, the WuTensor website was no longer accessible.

FOXXY. Similarly, a contract for the FOXXY token, created on October 2, 2024 at 11:06:59, also shows a mismatch compared to the one published through X. This indicates a similar pattern of fraudulent activity.

4) Conclusion and future potential analysis: Our findings emphasize the importance of cross-verifying contract address data across multiple sources. By examining the consistency between on-chain data (contract addresses from the blockchain) and off-chain data (contract addresses published and advertised to users), our analysis reveals the potential for deeper investigations into fraudulent tokens.

Moreover, our results suggest that integrating consistency and accessibility metrics could serve as new indicators to improve existing fraud detection frameworks. For example, discrepancies in contract addresses between the blockchain and promotional websites may signal potential scams, and projects with a high proportion of inaccessible URLs warrant further scrutiny.

B. Domain Clustering

1) Objective: In DeFi projects, websites play a key role in establishing token credibility. It is not uncommon to encounter projects that use multiple similar or identical domains, often due to updates, re-branding, or the creation of additional domains for marketing, regional targeting, or subdomains. However, some fraudulent projects may deliberately use domains similar or identical to those of reputable tokens in an attempt to mislead users into purchasing scam tokens. Multiple similar domains may not necessarily indicate scam projects, but they can raise a red flag. When used in conjunction with other indicators, such as CA consistency checking, token-score risk assessments, domain status analysis, and web content analysis, this can help identify potential fraudulent projects.

The goal of this use case is to identify suspicious domain names collected in the dataset by clustering similar URLs into meaningful groups. By recognizing patterns or textual similarities across domain names, we can collaborate with further investigations using our dataset to detect potential phishing or scam websites that exploit domain-based deception.

2) Method: We first normalize the domain by removing the protocol (e.g., `http://`) and top-level domains (e.g., `.com`, `.xyz`). Next, we represent each normalized domain using TF-IDF features. To construct the vocabulary for the corpus, each domain name is divided into overlapping character sequences or n-grams. For instance, the domain name `example.com` has the suffix `".com"` removed and is then split into n-grams such as `exa`, `xam`, `amp`, and `ple`. Each domain name is considered a "document" within the corpus. Using Term Frequency-Inverse Document Frequency (TF-IDF), each domain name is transformed into a vector of size `V`, where `V` represents the total number of unique sub-strings. Once the domain names are vectorized, we perform Agglomerative Clustering (AGNES) with cosine distance as the similarity metric.

TABLE V
EXAMPLES OF DOMAIN CLUSTERS

Cluster	Sample Size	Example Domains
A	19	strategicinures****.com strategicpeperes****.com strategicmemeres****.xyz ...
B	3	dogcoi****.cc dogcoin****.fun dogc****.community
C	4	S****.io s****.vip s****.pro ...
D	2	fug****.com fug****.vip

3) Findings: We performed the URL clustering for 2,605 domains from our dataset.

- The clustering process resulted in a total of 2,199 clusters.
- When filtering out *General-Purpose URLs Unrelated to the Token Itself*, the remaining clusters, categorized as *Project-Specific URLs*, are concentrated into smaller groups, typically containing 2 to 4 domains per cluster, examples are shown in Table V.

TABLE VI
INFORMATION ON `DOGCOI****.CC` AND `DOGCOIN****.FUN`

Domain	dogcoi****.cc	dogcoin****.fun
Creation Date	2024-10-22	2024-11-04
Whois Server	namecheap	namecheap
Name Server	ns.vercel-dns.com	dns.registar-server.com
Hosting Server	Vercel Inc	Vercel Inc
Safe Score by Third-party	98/99 (Safe)	1/99 (Unsafe)
Price and Transaction History	Active transactions from 2024-10-31 to 2025-01-05	All transaction values at 0
Telegram	Telegram groups with 477 subscribers, newest post date is Nov 11, 2024	Private account
X(Twitter)	Account not existed	Account not existed

While there may be instances where domain names differ but the same web page and content are used for scams, clustering similar domains can assist investigators in narrowing their focus to smaller groups, thereby aiding the scam detection process.

We conducted a preliminary investigation using Whois data, domain ages, and social media content to analyze suspected clusters, uncovering cases of domain similarities, some linked to potential scam projects.

Dogcoin. Through the clustering, we discovered a cluster of three domains `dogcoi****.cc`, `dogcoin****.fun`, and `dog****.community`. Upon checking the web content of the three pages, `dog****.community` contains a different webpage and is accompanied by a CA, which indicates a separate project. We focus our investigation on `dogcoi****.cc` and `dogcoin****.fun`. Interestingly, the first two web pages are identical; however, all links inside the web pages, such as the Buy link (link Uniswap), Telegram group, and X group, are different. The CAs on these web pages align with those from the verified source of contracts linked to these URLs.

Based on Table VI, there are two hypotheses:

- H1: Since the domain `dogcoi****.cc` is created first, we suspect `dogcoin****.fun` is the scam token, which mimics `dogcoi****.cc` to confuse and scam users. Users may be misled into confusing fake tokens with more popular and credible ones, leading them to click on purchase links for fraudulent tokens.
- H2: Both are from the same project. `dogcoin****.fun` can be the backup of `dogcoi****.cc` to start a new scheme.

Fugcoin. Other suspected scam cases are found in the cluster containing `fug****.com` and `fug****.vip`. While `fug****.com` remains active with more than 1,000 followers on X and 449 subscribers on Telegram, the `fug****.vip` domain is inaccessible. Additionally, the Telegram and X links in `fug****.vip` are no longer accessible, further raising suspicion of fraudulent activity.

Besides scam cases, we identified instances where similar domains represent different developmental versions of the same project.

4) Conclusion and potential investigation: URL clustering not only helps group similar domains to narrow the scope of investigations but also holds the potential for automating the scam detection process. By defining a clear set of criteria for identifying suspicious domains and integrating methods such as Whois checks and domain age analysis, this approach can enhance detection efficiency.

Additionally, leveraging existing datasets of reported or known scams, such as the REKT database, offers further opportunities. Applying URL clustering to these datasets could help uncover new scams linked to previously identified fraudulent projects, improving proactive scam detection capabilities.

C. Domain Age Analysis

Strategic Reserve-Related Domains. The dataset uncovered a cluster of domains centered around the theme of a "strategic reserve," leveraging terms such as "strategic," "reserve," and various cryptocurrency-related keywords. A total of 19 domains were identified, including examples such as `strategicinures****.com`, `strategicdogeres****.xyz`, and `strategic cryptores****.com`.

These domains appeared to capitalize on discussions about a U.S. Strategic Bitcoin Reserve, a concept that gained prominence in late 2024, particularly following media reports and political commentary on the topic [25], [26]. Many of these domains were registered or became active between November 7 and November 13, 2024, coinciding with this heightened public interest. The dataset revealed a strong correlation between the timing of these domain activities and the deployment of associated smart contracts.

Active domains like `strategicinures****.com` and `strategicdogeres****.xyz` hosted promotional content, including links to decentralized finance (DeFi) platforms like Uniswap and DEXTools. These sites provided smart contract addresses and in-centivized users to invest in projects marketed as part of a "strategic reserve." While further investigation is needed, the alignment of domain registration dates, contract deployments, and public narratives indicates a possible attempt to leverage the strategic reserve theme in a coordinated manner. This case study illustrates how the dataset can detect scam trends at an early stage by identifying clusters of domains and analyzing their on-chain and off-chain behaviors. The thematic clustering and timing correlations provide actionable insights for researchers and security professionals, enabling proactive scam detection and prevention.

VII. Discussion and Limitations

This study provides a novel dataset that integrates on-chain and off-chain data to enhance the detection and analysis of DeFi token scams. By systematically collecting and archiving ephemeral off-chain data, such as website content and social media activity, alongside immutable blockchain records, this research addresses critical gaps in existing resources. The dataset's utility was demonstrated through case studies that uncovered inconsistencies in contract address disclosures, suspicious clustering of domain names, and coordinated scam behaviors across social media platforms. These findings highlight the dataset's ability to detect fraudulent activities and support the development of more effective detection methodologies.

However, several limitations were encountered. Technical challenges arose in archiving certain web pages, particularly those using *Zstandard* compression [27][28], which the tool *pywb* does not fully support. Dynamic and redirected web pages also posed difficulties despite using additional tools and proxies. Addressing these limitations will be future work for capturing dynamic content.

VIII. Conclusion

This study introduces an integrated dataset that bridges the gap between on-chain and off-chain data to improve the detection and analysis of DeFi token scams. By systematically preserving ephemeral data and addressing key challenges in scam detection, the dataset enables comprehensive analysis of fraudulent activities. Case studies validate its effectiveness in uncovering inconsistencies, identifying domain-based deception, and analyzing scam behaviors on social media. While limitations such as scalability and archiving challenges remain, this work lays a foundation for future research to develop advanced detection frameworks. By addressing these challenges and expanding the dataset's scope, this study contributes to securing and fostering trust in the DeFi ecosystem.

IX. Ethical Considerations

This research adheres to strict ethical principles to ensure responsible collection, storage, and usage of data. All data, whether on-chain or off-chain, was sourced exclusively from publicly accessible platforms, with no access to private or restricted information. Archived data, including social media posts and web pages, was collected in compliance with platform terms of service and stored securely in a reproducible format to support transparency and long-term research utility.

While the nature of the collected off-chain data (e.g., social media posts) generally precludes personally identifiable information (PII) that can identify a specific individual, there is a possibility that some PII might exist within the dataset. To address this, we employed Presidio, an automated open-source tool developed by Microsoft, to identify and filter out any potential PII from the dataset prior to analysis or storage [29]. Any residual PII that could not be filtered is excluded from public dissemination to ensure compliance with ethical guidelines and privacy laws.

The dataset and findings from this study aim solely to enhance the understanding and detection of fraudulent activities in the DeFi ecosystem. Open access to the findings and datasets is restricted to reputable researchers who agree to use the data responsibly, explicitly prohibiting misuse. By maintaining these ethical standards, this study contributes to fostering collaboration and advancing research in DeFi security while ensuring the protection of individuals' privacy.

ACKNOWLEDGMENT

This paper is based on results obtained from a project, JPNP24003, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). This work was supported by JSPS KAKENHI Grant Number 22H03588.

REFERENCES

[1] Chainalysis, "The 2024 Crypto Crime Report," 2024, 104 pages, accessed: 2025-01-03. [Online]. Available: https://www.chainalysis.com/wp-content/uploads/2024/06/the-2024-crypto-crime-report-release.pdf

[2] Openware, "What Are Governance Tokens in DeFi Projects?" accessed: 2025-01-04. [Online]. Available: https://www.openware.com/news/articles/what-are-governance-tokens-in-defi-projects

[3] W. E. Forum, "Tokenization: Blockchain assets finance," accessed: 2025-01-04. [Online]. Available: https://www.weforum.org/stories/2024/12/tokenization-blockchain-assets-finance

[4] Solidus Labs, "Rug Pull Crypto Scams," accessed: 2025-01-04. [Online]. Available: https://www.soliduslabs.com/post/rug-pull-crypto-scams

[5] Uncirculars, "Pump and dump schemes," accessed: 2025-01-04. [Online]. Available: https://uncirculars.com/pump-and-dump-schemes

[6] T. Tokenizer, "Defi and regulation," accessed: 2025-01-04. [Online]. Available: https://thetokenizer.io/2024/11/22/defi-and-regulation

[7] S. Labs, "Off-chain and on-chain analysis," accessed: 2025-01-04. [Online]. Available: https://www.soliduslabs.com/post/off-chain-and-on-chain-analysis

[8] De.Fi, "Biggest Crypto Hacks & Scams - DeFi REKT Database," 2024. [Online]. Available: https://de.fi/rekt-database

[9] CryptoScamDB, "Cryptoscamdb," 2025, accessed: 2025-01-04. [Online]. Available: https://cryptoscamdb.org/

[10] G. Palaiokrassas, "Title missing (add manually if known)," *arXiv*, vol. 2306.07972, 2023, accessed: 2025-01-04.

[11] J. Clough and M. Edwards, "Pump, dump, and then what? the long-term impact of cryptocurrency pump-and-dump schemes," in *2023 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 2023, pp. 1–17.

[12] "Chainlist," 2025, accessed: 2025-01-15. [Online]. Available: https://chainlist.org/chain/1

[13] Ethereum, "Etherscan." [Online]. Available: https://etherscan.io/

[14] P. Xia, H. Wang, B. Gao, W. Su, Z. Yu, X. Luo, C. Zhang, X. Xiao, and G. Xu, "Trade or trick? detecting and characterizing scam tokens on uniswap decentralized exchange," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 5, no. 3, pp. 1–26, 2021.

[15] B. Mazorra, V. Adan, and V. Daza, "Do not rug on me: Leveraging machine learning techniques for automated scam detection," *Mathematics*, vol. 10, no. 6, p. 949, 2022.

[16] J. Xu and B. Livshits, "The anatomy of a cryptocurrency pump-and-dump scheme," in *Proceedings of the 28th USENIX Conference on Security Symposium*, ser. SEC'19. USENIX Association, 2019, p. 1609–1625.

[17] "Etherscan API," Etherscan, 2025, accessed: 2025-01-15. [Online]. Available: https://etherscan.io/apis

[18] F. Vogelsteller and V. Buterin, "Erc-20: Token standard," 2015, accessed: 2025-01-04. [Online]. Available: https://eips.ethereum.org/EIPS/eip-20

[19] Webrecorder Project, "pywb: Python Web Archive Toolkit," 2025, accessed: 2025-01-09. [Online]. Available: https://github.com/webrecorder/pywb

[20] Microsoft, "Playwright: End-to-End Testing for Modern Web Apps," 2025, accessed: 2025-01-09. [Online]. Available: https://playwright.dev/

[21] Harmon, J. Roesslein, and other contributors, "Tweepy." [Online]. Available: https://github.com/tweepy/tweepy

[22] LonamiWebs, "Telethon," 2025, accessed: 2025-01-09. [Online]. Available: https://github.com/LonamiWebs/Telethon

[23] g1879, "Drissionpage," 2023, accessed: 2025-01-15. [Online]. Available: https://github.com/g1879/DrissionPage

[24] DEXTools.io, "Crypto glossary - dextscore," 2024. [Online]. Available: https://info.dextools.io/crypto-glossary/dextscore/

[25] Reuters, "How Would a U.S. Strategic Bitcoin Reserve Work?" 2024, accessed: 2025-01-04. [Online]. Available: https://www.reuters.com/technology/how-would-us-bitcoin-strategic-reserve-work-2024-12-16/

[26] Investopedia, "What Would Be The Point Of A Strategic Bitcoin Reserve?" 2024, accessed: 2025-01-04. [Online]. Available: https://www.investopedia.com/what-would-be-the-point-of-a-strategic-bitcoin-reserve-8763067

[27] Y. Collet, "Zstandard - Fast real-time compression algorithm," https://github.com/facebook/zstd, 2016, accessed: 2025-01-15.

[28] Y. Collet and C. Turner, "Zstandard Compression and the application/zstd Media Type," Internet Requests for Comments, February 2021, https://datatracker.ietf.org/doc/html/rfc8878.

[29] Microsoft, "Presidio: Open-source pii anonymization and redaction tool," 2025, accessed: 2025-01-15. [Online]. Available: https://github.com/microsoft/presidio