IEEE ICBC'24 1570982010

A Multi-Source and Multi-Kernel ML Approach for Hunting DeFi Rug Pulls and Scams

***Abstract*—In the fast-changing world of Decentralized Finance (DeFi), while DeFi projects make finance more accessible and create new ways to use financial services without central authority, the rise of scams and rug pulls is a big problem, causing loss of trust and major financial losses. This paper introduces an innovative, machine learning (ML)-based approach to proactively detect and mitigate fraudulent activities in DeFi projects, a crucial step towards restoring trust in this evolving sector. Our methodology is rooted in a multi-source, multi-kernel ML paradigm, meticulously tailored to assimilate and analyze a wide array of data points from various facets of the DeFi ecosystem. This extensive data aggregation approach ensures a comprehensive, multi-dimensional understanding of DeFi operations, which is essential for effective fraud detection. Our approach features a multi-kernel design, integrating diverse sophisticated ML models. This includes the Prophet forecasting tool, FinBERT tailored for financial Natural Language Processing; the robust XGBoost for advanced classification, CodeBERT, specialized in code understanding and generation; and the Agglomerative Clustering Algorithm for nuanced anomaly detection. These components synergize to form a robust surveillance system against DeFi fraud. A cornerstone of our approach is its emphasis on proactive detection through a continuous feedback loop, which iteratively enhances the predictive capabilities of our models. This dynamic mechanism facilitates real-time identification of potential threats, keeping the system adaptive and vigilant against evolving scam tactics. Furthermore, our strategy incorporates Explainable AI (XAI) visualizations to demystify decision-making, providing valuable insights into scam patterns and behaviors. This visualization (SHAP and LIME) plays a pivotal role in boosting the confidence of DeFi stakeholders, particularly project owners and investors, by enhancing transparency and understanding. The success of our approach is based on a complete data analysis framework and using an ensemble of ML models. This is a big step in restoring trust in DeFi projects and helps create a safe, standardized environment for various financial activities in the DeFi space.**

***Index Terms*—DeFi, Decentralized Finance, Security, Scams, Rug Pull, AI, Machine Learning.**

I. Introduction

Decentralized Finance or DeFi [1], [2], [3] is a groundbreaking financial innovation that harnesses blockchain technology [4] to offer direct access to financial services, bypassing traditional intermediaries. At its core are Smart Contracts: automated, programmed codes that orchestrate transaction flows based on financial agreements. This automation reduces the cost of managing the traditional centralized bank or financial institution. DeFi projects make financial services accessible to a wide range of audiences, and the transparency and immutability of blockchain transactions in DeFi provide trust and increased security in digitized and decentralized financial services. Poised for growth, DeFi is redefining traditional financial services through smart contract-driven transactions and a secure, transparent, decentralized ledger.

Key among DeFi services [5] are *Lending and Borrowing* [6] platforms, which allow users to lend their assets and take out loans directly by interacting through smart contracts. *Decentralized Exchanges* (DEXs) [7], [8] are another prominent service that enables trading cryptocurrencies and digital assets. Within DEXs, Automated Market Makers (AMMs) [9] use smart contracts and algorithms for trading without traditional order books, relying on liquidity pools [10]—crucial DeFi elements representing pooled user funds. *Stablecoins* [11], [12] add stability to the DeFi landscape by minimizing the risk and price volatility by allowing cryptocurrency holders to switch to equivalent U.S. Dollars or gold. Another service is *Yeild Farming* [13], which is a way to earn returns from DeFi projects by depositing tokens in a liquidity pool or trading pool. Finally, *Asset Management* [14] integrates traditional financial services with smart contracts, enhancing transparency in DeFi project funds. This sector includes yield aggregators and digital assets [15] as its key protocols. Figure 1 provides an overview of the DeFi landscape as central to this study.

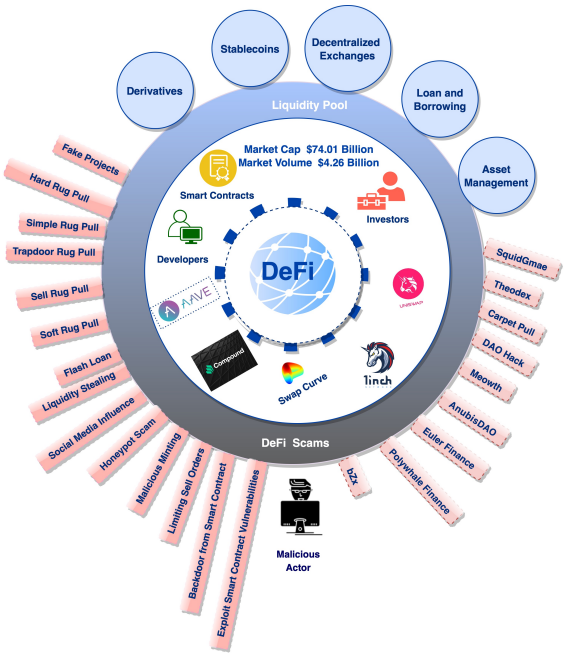

Fig. 1. DeFi Landscape

Despite the allure of high returns drawing many investors to DeFi, making it a lucrative marketplace for trading and investment, it is also increasingly vulnerable to modern scams and Rug Pulls, as noted in various studies [16], [17], [18], [19], [20], [21]. Research work in [22] summarizes the security issues in DeFi protocols. Below, we discuss details of the major category of scams highlighted in Table I.

Baseline scams in DeFi projects are the *Exit Scams* where the scammers apply various techniques to disappear with the funds of DeFi project. *Rug Pull Scams*, very prevalent in the Crypto world, are the scenarios where the project owners or developers or developers employ deceptive tactics, such as hacks and backdoors, to drain the liquid pool funds and vanish with the assets. Within this category, *Soft Rug Pulls* are characterized by more gradual and less immediately obvious scamming techniques. Conversely, *Hard Rug Pulls* are abrupt and have a more drastic impact on the project, significantly devaluing the assets of investors. Another variant, the *Sell Rug Pull* Scam, involves tricking investors with promises of high returns and project hype, only for the scammers to disappear after the sale concludes.

Another famous technique to scam funds of a DeFi project is through smart contracts, which are the heart of a DeFi project. These scams can be orchestrated either by the project's developers or by external parties exploiting weak coding or security vulner-

abilities. One such scam, the *Backdoor from Smart Contracts* is a scam where the developers of the project create undetectable backdoors that could limit the investors from selling their tokens or accessing their funds, also referred to as *Limiting Sell Orders* scam. Another prevalent form of fraud is the *Liquidity Stealing* scam, where developers manipulate smart contracts to covertly transfer investor funds to a single address.

The *Flash Loan Attacks* occur when malicious actors exploit the system by taking out substantial loans in a single transaction, thereby affecting the value of the project. A notable example is the Euler Finance attack incident, as reported in [23]. Similarly, *Fake Projects* involve scammers creating counterfeit projects and tokens that mimic established ones, deceiving investors into funding them. *Malicious Minting* refers to the practice where nefarious individuals excessively mint tokens, consequently destabilizing the token value in the pool. *Honeypot Scam* is a tactic used in DeFi projects, where exaggerated hype lures investors into buying tokens that cannot be sold or traded, effectively trapping their funds indefinitely. Another scam type is *Fake or Deceptive Token Sales*, such as the Squid Game token, resulting in a loss of $3.38 million. *Ponzi Schemes* are also prevalent; for instance, the Russian DeFi platform Forsage was implicated in a $340 million scam. *Exploit Attacks* [24], as another category, include incidents like the Maiar decentralized exchange hack, where attackers exploited a vulnerability to steal tokens, swapping some for Ethereum, leading to a total estimated loss of $113 million. Figure 2 illustrates the monetary losses in recent scam incidents in the DeFi space.

TABLE I
DEFI SCAMS DETAILS

DeFi Scam Name	Description
Hard Rug Pulls	Scammers abruptly withdraw funds from a DeFi project, significantly impacting its value.
Sell Rug Pulls	Scammers attract investors with a hyped DeFi project, sell fake tokens, then vanish post-sale.
Soft Rug Pulls	A gradual siphoning of project funds by scammers, often going unnoticed initially.
Trapdoor Rug Pulls	Scammers trap investors in a DeFi project and exploit the accumulated funds.
Exit Scams	Project owners or creators abscond with the project's funds.
Pump and Dump	Scammers inflate a token's value, then sell at a high price, leaving investors with worthless tokens.
Flash Loan Attacks	Malicious actors borrow massive funds to destabilize a DeFi project.
Honeypot	Investors are prevented from withdrawing their funds by scammers.
Ponzi Scheme	New investor funds are used to pay returns to existing investors, perpetuating a cycle of fraud.
Exploit Attacks	Scams that take advantage of vulnerabilities and security flaws in DeFi projects.
Smart Contract Exploits	Attacks targeting the smart contracts of a DeFi project.
Smart Contract Backdoors	Developers embed backdoors in smart contracts to manipulate funds and defraud investors.
Liquidity Stealing	Malicious actors drain liquidity from a project for personal gain, destabilizing and devaluing it.

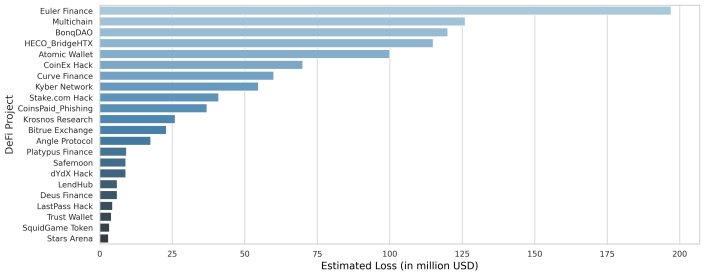

Fig. 2. DeFi Scams Losses

A. Motivation

As of recent data, the market capitalization of DeFi cryptocurrencies is valued at approximately 74.01 billion, accompanied by a substantial total trading volume of around 4.26 billion [25]. The DeFi sector, characterized by its novelty and lack of stringent regulation, has become increasingly susceptible to many security threats, including sophisticated hacks, scams, and fraudulent activities aimed at siphoning digital assets. These incidents pose direct financial risks and significantly impair the overall financial stability and resilience of decentralized finance. The escalating frequency and complexity of scams in DeFi underscore the urgent need for a comprehensive understanding of the inherent vulnerabilities and potential threat vectors. Establishing robust security measures and trust in these emerging financial services is paramount to ensure their sustainable growth and enable a broader spectrum of users to engage safely and profit from the cryptocurrency market.

This research is motivated by the critical need to harness intelligence and automate detection strategies for identifying potential Scams in the DeFi Landscape. By leveraging data-driven insights from historical scams and discerning underlying patterns, our approach aims to identify and neutralize potential future scams proactively. This proactive detection mechanism is crucial for alerting the cryptocurrency community about imminent threats, thereby facilitating informed and safer investment decisions. Our work is pivotal in fortifying the security infrastructure of DeFi platforms and plays a significant role in enhancing investor confidence and fostering a safer, more resilient DeFi ecosystem. Integrating advanced machine learning techniques and analytics in scam detection represents a forward-thinking stride in tackling the evolving challenges in decentralized finance, ultimately contributing to the sector's long-term stability and credibility.

B. Contributions

Our research delineates crucial findings and contributions, advancing the field of DeFi security through the following key achievements:

- *Comprehensive DeFi Projects Dataset Assembly:* We have meticulously curated a comprehensive dataset encompassing many DeFi projects. This dataset serves as the foundational bedrock for our analysis, offering a panoramic view of the DeFi landscape. It includes diverse data types and sources, ensuring a holistic approach to understanding and identifying potential scam vectors in the DeFi space.
- *ML and NLP Powered Smart Contract Audits:* Our approach integrates advanced machine learning techniques with NLP to conduct thorough audits of smart contracts. This fusion of ML and NLP enhances the accuracy of detecting vulnerabilities and potential scam signatures in the Smart Contracts and streamlines the audit process, making it more efficient and comprehensive.
- *Multi-Source, Multi-Kernel ML Framework for DeFi Project Monitoring:* We employ a multi-model ML framework that

synthesizes information from various sources to surveil DeFi projects. This approach leverages the strengths of different ML models to provide a nuanced, multi-faceted perspective on the security and integrity of DeFi projects, enabling proactive detection of fraudulent activities.

- *LLM and BERT NLP for Extracting Insights from Social Media Views:* Utilizing state-of-the-art NLP models, including Large Language Models (LLM) and BERT, our research gains deep insights from social media discussions surrounding DeFi projects. This analysis aids in understanding public sentiment and identifying early warning signs of potential scams circulating in community forums and discussions.
- *Research Scope Beyond DEXs and Uniswap Tokens:* Our research extends beyond the usual focus of Decentralized Exchanges (DEXs) and Uniswap exchange tokens. It encompasses a wider array of DeFi projects, ensuring that our insights and solution are broadly relevant and beneficial across the heterogeneous and rapidly evolving landscape of DeFi.

This paper is structured as follows: Section II discusses related literature. Section III describes our proposed approach that highlights the details of major data sources and feature extraction, ML pipelines, and algorithms. Section IV presents the evaluation results of the proposed approach and discusses the findings. Section V gives the conclusion and the future work.

II. Related Work

The research landscape addressing fraud and scams in DeFi is rapidly evolving, with numerous studies contributing valuable insights and methodologies. This section reviews some key research works that have laid the groundwork for our study. Palaiokrassas et al. [26] employed Covalent APIs to extract data from 23 prominent DeFi projects, including Aave and Compound. Their work involved extracting over 400 features specific to DeFi and cryptocurrencies. By training machine learning models like XGBoost and Neural Networks, they identified fraudulent accounts interacting within the DeFi ecosystem, including data from multiple blockchain networks, notably Ethereum, and integrating DeFi-specific and general features significantly enhanced model performance. In a detailed analysis of scam tokens on the Uniswap exchange, a study [27] expanded an existing dataset to include 18,000 tokens. The research focused on unraveling the tactics behind Rug Pull scams, leveraging machine learning to label data, and utilizing Slither for smart contract audits. However, this study was confined to the DEX protocol of DeFi and exclusively examined Uniswap tokens, indicating a gap in broader DeFi scam detection.

Huynh [28] proposed using ML classifiers, including Random Forest, XGBoost, and LightGBM, to identify trapdoor scam tokens within the Uniswap Exchange. This work comprehensively explored Trapdoor Rug Pull scenarios, including code snippets from smart contracts that permit the buying but restrict the selling of tokens. This study highlighted the intricacies of smart contract manipulation in DeFi scams. Also, Xia et al. [21], in their research on Uniswap, introduced an ML framework based on the guilt-by-association principle. It has been found that over 10K scam tokens are listed on Uniswap, which indicates that around 50% are scam tokens. The approach can act as a whistleblower, identifying scam tokens early on. In the study cited as [29], the authors systematically analyze vulnerabilities in the DeFi ecosystem on the Ethereum platform. Their investigation is multi-tiered, delving into issues at various levels of the DeFi structure. Beyond theoretical analysis, they extend their research to examine real-world attack scenarios, providing an empirical dimension to their study. This comprehensive approach offers a holistic view of the current state of DeFi security and its trajectory toward more robust and resilient financial technologies.

Shifting focus to the NFT space, studies [30], [31] delved into Rug Pull scams in NFT marketplaces like OpenSea. These works analyzed existing Rug Pull scams and identified patterns in the operations of Rug Pull mafia groups. The insights gleaned from these studies are invaluable in understanding the commonalities and behavioral patterns of Rug Pull scams, which can be leveraged to enhance scam detection in the DeFi sector. In another work, DeFiWarder [32], the authors address protecting DeFi Apps by considering token-leaking vulnerabilities. DeFiWarder records the execution logs (traces) of smart contracts and user relations based on role mining. In their study, Luo et al. [33] present a comprehensive survey on AI-powered fraud detection in the DeFi ecosystem, categorizing frauds according to the life cycle stages of DeFi projects. They review various AI detection methods, including statistical modeling and machine learning techniques, and highlight the effectiveness of specific models in different stages. This work serves as a crucial guide for future research, aiming to enhance security in the rapidly evolving DeFi landscape.

Moreover, Dotan et al. [34] examine the vulnerabilities associated with decentralized governance in DeFi, particularly focusing on the use and impact of governance tokens in Decentralized Autonomous Organizations (DAOs). They reveal that governance tokens are often underutilized for voting, affected by gas prices, and lead to centralization in voting. The study also delves into various governance attacks and the manipulative use of governance tokens across different platforms, highlighting the complexities and security concerns in DeFi's decentralized governance model. While existing research has significantly advanced our understanding of DeFi fraud and scams, particularly in the realms of smart contract analysis and specialized token scams, there remains a need for more expansive and diverse studies. Our research aims to fill these gaps by providing a broader DeFi scam detection and prevention solution.

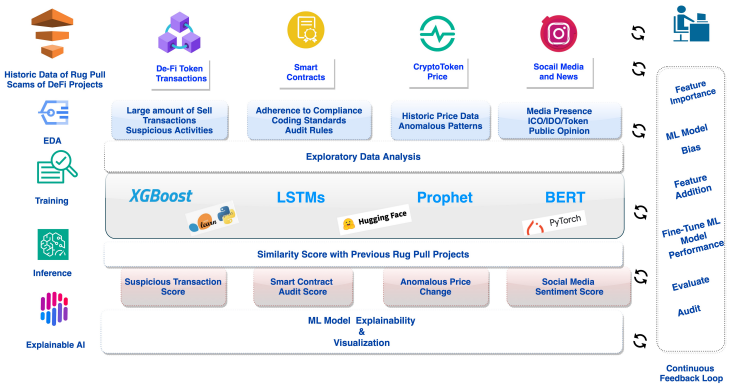

Fig. 3. Machine Learning Models Workflow

III. Proposed Approach

A. Architecture

Our proposed approach considers multiple data sources influencing the DeFi project risk score. We analyze the collected data and derive patterns from historic Rug Pulls and scams of DeFi projects. The derived features are ingested into the training pipeline for a consortium of ML models to learn the pattern in data and gain insights. The combination of trending analytics and ML models outperforms the existing methodologies. The designed feedback loop continuously monitors ML model scores and incorporates human intelligence into the training pipeline. Figure 3 illustrates the architecture of the ML models workflow of our proposed approach. As shown in the figure, our

data collection process for DeFi projects, encompasses Smart Contract Source Code, Token Price Data, Token Transaction Data, and Social Media Content. Another crucial data source is 'Historic Data of Rug Pulls and Scams in DeFi.' This repository contains historical records of fraudulent smart contracts and transactions, along with detailed descriptions of the associated malicious activities. This dataset is vital for our data collection and labeling efforts. We compiled this information by analyzing patterns of fraudulent activities in the DeFi sector over the years. Subsequently, we applied data analytics methodologies to revalidate the analysis through a combination of programmatic and manual code logic verification, assigning labels to each dataset. We then developed a series of training pipelines for machine learning models, aimed at detecting patterns and deriving features and labels using anomaly detection algorithms. For NLP-based applications, such as auditing Smart Contracts source code, we utilized the pre-trained CodeBERT model, further optimized for smart contract code data. Additionally, FinBERT was employed to extract insights from social media content and discussions. We discuss details of each component in the below sections.

Data Sources, Collection, and Feature Extraction: In our methodology for analyzing decentralized finance (DeFi) projects, we harness diverse data sources. These include:

Smart Contracts of DeFi Projects: We focus on smart contracts from DeFi projects with diverse protocols selected for this research, as listed in Table II. Audited and verified on the Ethereum blockchain, these projects provide a foundation for our analysis. By scrutinizing these contracts, we can understand the underlying mechanisms of these platforms.

DeFi Token Transaction Records: Transaction records of DeFi tokens are pivotal. They provide a window into the operational dynamics of these tokens in the market.

Social Media Discussions and News Coverage: To capture the public sentiment and the evolving narrative around DeFi projects, we analyze discussions on social media platforms and news articles from publicly available data sources and the paid commercial APIs, including Twitter (X) and Reddit.

Token Price Statistics: The pricing trends of DeFi tokens offer insights into market perceptions and investor behaviors.

Our methodology involves a structured data collection and feature extraction approach, as illustrated in Figure 4. We begin by analyzing smart contract data from these DeFi projects. This initial phase involves identifying and labeling data as either suspicious or non-suspicious based on the historic fraudulent transactions on smart contracts.

After classifying the smart contract data using the historical data, we delve deeper into extracting transaction data associated with these contracts. This enables us to construct a dataset for transaction data, labeling each transaction with an is_scam tag if it's connected to fraudulent activities. We rely on tools for data collection, including the Etherscan API, GitHub, and CoinGecko API. These tools facilitate access to a wealth of Ethereum Blockchain Network data, encompassing historical and current transactional data. This data forms the backbone of our analysis and helps build robust inference pipelines. In addition to these technical sources, we also explore social media and news outlets for relevant data.While publicly available data is our primary source, we turn to paid membership APIs for more comprehensive insights, especially for commercial purposes. These APIs provide access to larger volumes of on-demand data, enriching our analysis with a broader perspective.

The data collected is further processed to extract meaningful information that can be ingested into the training pipeline of ML models. Below summarizes the steps for dataset collection and feature extraction.

TABLE II
DEFI PROJECTS LIST

Token Name	*DeFi Protocol*	*Website*
AAve	Lending	Link
Compound	Lending	Link
Curve Finance	Lending	Link
Uniswap	DEX	Link
Pancakeswap	DEX	Link
Polkaswap (PSWAP)	DEX	Link
Gluwa Credit	Stablecoins	Link
Dai	Stablecoins	Link
Sai Stableco	Stablecoins	Link

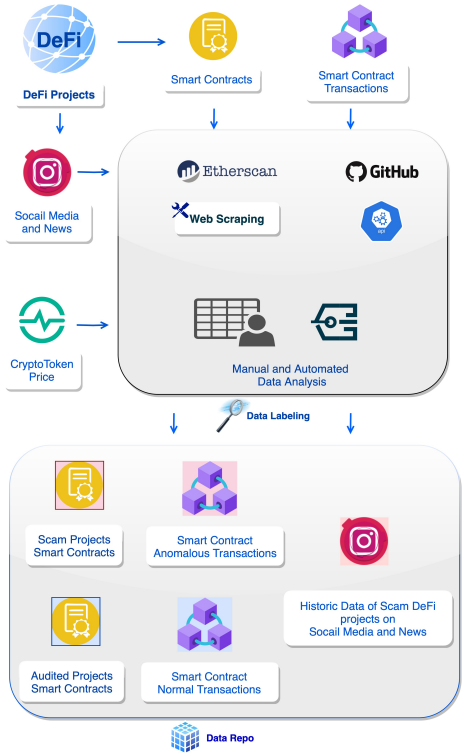

Fig. 4. Feature Extraction

1) List the DeFi projects registered and audited on Ethereum.
2) Scrap the publicly available smart contracts address details of Scam projects.
3) Extract the smart contract source code using Etherscan API.
4) Extract the transactions related to the smart contracts.
5) Extract the transactions with the addresses reported as fraudulent on Social Media and Etherscan Comments section.
6) Scrap the publicly available data of Crypto and DeFi project social media discussions and content.
7) Analyzes the data points and labels per the pre-defined logic coded in pseudo algorithms.

In the next sections, we discuss the details of the training pipeline built with the data sources, the features extracted and derived, and ingested as training data for ML models.

DeFi Transaction Data ML Model: The historical transaction data of fraudulent DeFi projects serves as an exemplary pattern to verify for calculating the risk score of a DeFI project. Algorithm 1 gives an intuition into data extraction, feature analysis, and labeling for DeFi Projects' transaction data.

In the threat model for Lending and Borrowing Protocol Transactions, key malicious actors include Flash Borrowers, who engage in short-term loans, and Collusion Attackers, who manipulate interest rates and asset prices. Similarly, for DEX Protocol

TABLE III
DEX TRADE TRANSACTION DATA FEATURES

Feature	*Description*
Action	Describes the type of transaction or action performed, e.g., 'Buy', 'Sell', 'Swap'.
Amount (Out)	Represents the amount of Cryptocurrency or tokens being sent out of the wallet.
Token (Out)	Specifies the type of token or Cryptocurrency being sent out.
Amount (In)	Indicates the amount of Cryptocurrency or tokens received in the transaction.
Token (In)	Specifies the type of token or Cryptocurrency received.
Swapped Rate	Shows the exchange rate or conversion rate at which the tokens were swapped.
Swapped Pair	Specifies the pair of tokens being swapped, e.g., 'BTC/ETH', 'USDT/DAI'.
DEX	Stands for Decentralized Exchange and indicates the exchange platform where the transaction occurred.
is_scam	A boolean value indicates whether the transaction is considered suspicious or a potential scam.

TABLE IV
DEFI TRANSACTION DATA FEATURES

Feature Name	*Feature Description*
blockNumber	The number of the block in the blockchain.
timeStamp	The timestamp when the block was created.
hash	The unique hash identifier for the block.
nonce	A random number used in mining to create a new block.
blockHash	The hash of the block's data.
transactionIndex	The index of the transaction within the block.
from	The sender's address in a transaction.
to	The recipient's address in a transaction.
value	The value of credit in the transaction.
gas	The amount of gas used in the transaction.
gasPrice	The price of gas in the transaction.
isError	Indicates if there was an error in the transaction.
txreceipt_status	The status of the transaction receipt.
input	The input data of the transaction.
contractAddress	The address of the smart contract
cumulativeGasUsed	The cumulative gas used in all transactions.
gasUsed	The gas used in the current transaction.
confirmations	The number of confirmations for the block.
methodId	The unique identifier for the method called in a contract.
functionName	The name of the function called in a contract.
is_high_value_scam	Indicates if the transaction is a high-value scam.
is_frequent_small_scam	Indicates if the transaction is a frequent small scam.
is_scam	Label that Indicates if the transaction is a scam.

Transactions, the threat model mirrors those found in traditional finance (TradFi) or centralized finance (CeFi) exchanges. Here, the threats include price manipulators and exploiters, front-running traders, and automated trading bots. Common to both protocols are risks like Sybil attacks and smart contract vulnerabilities. The transaction types vary between these protocols,

TABLE V
FEATURES FOR SMART CONTRACTS ML MODELS

Feature	*Description*
Smart Contract Source Code	The Actual Code of Smart Contract in Solidity.
Smart Contract ABI	Application Binary Interface helps to differentiate the smart contracts.
Smart Contract Audit Flag	Indicates whether the smart contract has undergone an audit by a third party to assess security, performance, and compliance with best practices.

TABLE VI
FEATURES FOR NEWS AND SOCIAL MEDIA ML MODELS

Feature	*Description*
Community Engagement Metrics	Gathers data on the project's social media presence and other engagement metrics, reflecting community interest and involvement.
News Sentiment Analysis	Collects information related to the DeFi Token, performs sentiment analysis and provides insights into the overall sentiment and perception of the token in the community.

necessitating different feature sets; Tables IV and III detail the transaction features unique to the Lending and Trading aspects of DeFi projects. Below, we list activities that could potentially leave a project underfunded.

1) Large amounts of suspicious sell transactions.
2) Token has only buy transactions, but no sell transactions are recorded.
3) Large transaction to/from Liquidity Pool.
4) Large amount of loans Borrowed in one transaction.

Smart Contract Audit ML Model: ML for smart contract vulnerabilities is a critical area of research [35]. The historic Rug Pull scams review has emphasized the importance of Smart Contracts audits, which would have prevented the occurrence of many Rug Pull scams [36], [37]. Smart contracts are a crucial part of a DeFi project that orchestrates and dictates how transactions are performed. The unintended and non-malicious design of the DeFi project with improper coding of Smart Contracts without compelling to standards of compliance and risk assessments make the entire DeFi vulnerable to attacks and incur losses. Sudden token transfer to a single address, the smart contract's ability to draw the entire token amount can signify a potential Rug Pull scam. The audit of smart contracts provides insights into the complexity of smart contracts to the developers for better coding and adherence to compliance and risk management standards. The DeFi project with audited and verified smart contracts gains investors' trust and the project owners' reputation by sealing the security and providing an exploit-free environment.

Table VI outlines a feature set for the Smart Contract Audit ML Model. We build an ML pipeline with a large language model optimized with smart contract code and integrate it with a clustering algorithm to identify anomalous patterns in the Smart Contract Code. Algorithm 2 summarizes the steps for deriving the audit score of smart contracts source code.

DeFi Token Price Anomaly Detection ML Model: Anomalous price change is a trend to look out for suspicious activity in any crypto-linked project. Along with DeFi Rug Pull scams, price change is a crucial indicator of malicious activity such as pump and dump schemes. It can be manipulated to gain trust in the project by artificially pumping the value of the token price. In

Algorithm 1. DeFi Transactions Data Preprocessing

```
procedure MAIN
    data_sources ← collect_data_sources()
    raw_data ← collect_data(data_sources)
    cleaned_data ← clean_data(raw_data)
    buy_transactions ← filter_buy_transactions(cleaned_data)
    large_buy_transactions ← filter_large_transactions(buy_transactions)
    store_data(large_buy_transactions)
    maintain_data_continuously(large_buy_transactions)
end procedure
function COLLECT_DATA_SOURCES
    sources ← [source1, source2, ...]
    return sources
end function
function COLLECT_DATA(sources : List)
    combined_data ← DataFrame()
    for each source in sources do
        data ← fetch_data_from_source(source)
        combined_data.append(data)
    end for
    return combined_data
end function
function CLEAN_DATA(data : DataFrame)
    cleaned_data ← remove_inconsistencies(data)
    cleaned_data ← standardize_formats(cleaned_data)
    return cleaned_data
end function
function DEX_FILTER_BUY_TRANSACTIONS(data : DataFrame)
    // Extract only buy transactions from the dataset for a Token
    buy_transactions ← data[data['transaction_type'] ==' buy']
    return buy_transactions
end function
function LENDING_FILTER_LARGE_TRANSACTIONS(transactions : DataFrame)
    large_transaction_threshold ← determine_large_transaction_threshold(transactions)
    large_transactions ← transactions[transactions['amount'] ≥ large_transaction_threshold]
    return large_transactions
end function
function LENDING_DETERMINE_LARGE
_TRANSACTION_THRESHOLD(transactions : DataFrame)
    threshold ← transactions['amount'].quantile(0.90)
    return threshold
end function
function STORE_DATA(data : DataFrame)
    store_to_database(data)
end function
```

Algorithm 2. Semantic Analysis and Clustering of Smart Contracts

```
function PREPROCANDLABEL(smartContracts)
    for each contract in smartContracts do
        preprocContract ← PREPROCCONTRACT(contract)
        label ← CREATELABEL(preprocessedContract)
        Assign label based on DeFi protocol and audit flag
        Store preprocessedContract and label for further processing
    end for
end function
procedure SEMANALYSIS(codeData)
    Utilize Microsoft's CodeBert model
    Embed code data into n-dimensional vector space
    for each codeSnippet in codeData do
        embedding ← CODEBERTEMBEDDING(codeSnippet)
        Store embedding for clustering and similarity analysis
    end for
end procedure
function CALCULATESIMWITHSCAM(embeddings, scamEmbeddings)
    for each embedding in embeddings do
        for each scamEmbedding in scamEmbeddings do
            similarityScore ← COMPUTESIMILARITY(embedding, scamEmbedding)
            Store similarityScore for further analysis
        end for
    end for
end function
procedure AGGLOCLUSTERING(embeddings)
    Apply hierarchical agglomerative clustering
    clusters ← HIERARCHICALCLUSTERING(embeddings)
    for each cluster in clusters do
        Analyze cluster properties
    end for
    Return grouped smart contract clusters
end procedure
smartContracts ← Load smart contract data
scamContracts ← Load scam contract data
preprocData ← PREPROCANDLABEL(smartContracts)
embeddings ← SEMANALYSIS(preprocessedData)
scamEmbeddings ← SEMANALYSIS(scamContracts)
CALCULATESIMWITHSCAM(embeddings, scamEmbeddings)
clusteredData ← AGGLOCLUSTERING(embeddings)
```

most cases, in the Rug Pull scams, the price is manipulated to its peak value before, drawing the entire token worth for higher benefits. The features in the time-series analysis data are TokenId, which represents the Token code of the DeFi project, and open, high, low, close for the price details of the token. We applied Facebook's time-series analytic model to detect the anomalous pattern in the price change of the Token Price data. Trained continuously with the latest price data, the models gain insights into the usual price change trends and score the anomalous factor as high when the price graph shows suspicious activity.

Social Media and News ML Model: DeFi project discussions on social media and news have established a known pattern for Rug Pull and a few other DeFi Scams, where the scammers spend time luring the users with Initial coin offerings of the DeFi project and promoting it as promising high returns on investment and try to create Fear Of Missing Out (FOMO) situations where the investors are made to invest on the project. Scammers extensively leverage social media channels and News to advertise and attract investors. The other use of tracking the DeFi project sentiment score on social media is to quickly pick the negative or red flag discussions, prevent further investments, and save the investors from incurring losses by investing in the project flagged suspicious by other investors.

We applied NLP models [38] to extract meaningful information from public opinion through social media and news data on the internet and the APIs. We train an ML model to identify the familiar pattern of the previous Rug Pull project's social media presence. Additionally, we train an ML model to capture the social media sentiment on the DeFi Projects. We generate features VI and extract the pattern from Rug-pulled DeFi projects for news and social media content data. Another important feature is the social media sentiment on the DeFi projects to capture the discussion and posts on the suspicious activity of a DeFi project.

B. Inference Pipeline

The inference pipeline is used to infer the prediction scores from each ML model and calculate the weighted average for the overall prediction score. Algorithm 3 summarizes the methodology of the inference pipeline for calculating the overall prediction for the possibility of a token being a scam token. The Inference pipeline connects to each ML pipeline to retrieve prediction scores on the specific DeFi project. We derive the overall Rug pull prediction score by performing a weighted average of the scores from each ML model.

C. Feedback loop & Analytics Repo

We gather insights from human intelligence (the auditor) and regularly review scores from machine learning (ML) models.

Algorithm 3. Inference Pipeline

procedure MAIN($token\_id$)
  $SmartContracts\_Scam\_Prediction\_Score \leftarrow$ SMARTCONTRACTS_ML_PIPELINE($token\_id$)
  $TokenPriceData\_Anomalies\_Score \leftarrow$ GETTOKENPRICEDATAANOMALIESSCORE($token\_id$)
  $TransactionData\_Scam_Prediction\_Score \leftarrow$ GETTRANSACTIONDATAANOMALIESSCORE($token\_id$)
  $SocialMediaContent\_Sentiment\_Score \leftarrow$ GETSOCIALMEDIACONTENTSENTIMENTSCORE($token\_id$)
  CALCULATEOVERALLSCORE(SmartContracts_Scam_Score, TokenPriceData_Anomalies_Score, TransactionData_Anomalies_Score, SocialMediaContent_Sentiment_Score)
end procedure
function GETTOKENPRICEDATAANOMALIESSCORE($token\_id$)
  Retrieve and analyze token price data for anomalies for the token with ID: token_id
  Implement your algorithm here
  Calculate and return the anomalies score
end function
function GETTRANSACTIONDATASCAMSCORE($token\_id$)
  Retrieve and analyze transaction data for anomalies for the token with ID: token_id
  Implement your algorithm here
  Calculate and return the anomalies score
end function
function GETSOCIALMEDIACONTENTSENTIMENTSCORE($token\_id$)
  Retrieve and analyze social media content for sentiment for the token with ID: token_id
  Implement your sentiment analysis algorithm here
  Calculate and return the sentiment score
end function
procedure CALCULATEOVERALLSCORE($SmartContracts\_Score$, $TokenPriceData\_Score$, $TransactionData\_Score$, $SocialMediaContent\_Score$)
  $Weight\_SmartContracts \leftarrow 0.4$
  $Weight\_TransactionData \leftarrow 0.3$
  $Weight\_TokenPriceData \leftarrow 0.2$
  $Weight\_SocialMediaContent \leftarrow 0.1$
  $Overall\_Prediction\_Score \leftarrow (SmartContracts\_Score \cdot Weight\_SmartContracts) + (TokenPriceData\_Score \cdot Weight\_TokenPriceData) + (TransactionData\_Score \cdot Weight\_TransactionData) + (SocialMediaContent\_Score \cdot Weight\_SocialMediaContent)$
  Output or store the Overall Prediction Score
end procedure

This process enables us to reassess the importance of features, reducing the risks of model degradation and bias. The Auditor receives detailed scores and insights about DeFi projects. To better understand ML model predictions, we utilize explainability tools such as SHAP and LIME [39]. The findings from these tools are analyzed to refine feature importance and update our models. To support the workflow, we maintain a repository with key data points, including: Importance of each data feature, ML model inference data for continuous improvement and future training, Explainability analysis of ML model predictions, and Code embeddings of Smart Contracts identified as scams

IV. Evaluation and Results

This section provides an empirical evaluation of the proposed approach. We outline the experimental setup, describe the datasets and model properties, and compare the results.

A. Experimental setup

To evaluate the performance of our proposed approach, we have created a development environment using Google Colab Pro plus T4 GPU with 160 GB Hard disk, 50 GB System RAM, and 15 GB GPU RAM and a MacBook Pro GPU with 500 GB Hard disk and 32 GB RAM. The data extraction and preprocessing are coded in Python 3.10.12. For distributing the work to worker nodes and improving workflow efficiency, we have integrated the methods with ray [40], which efficiently upgrades the processing speed and training time. Training pipelines and inference pipeline are implemented using PyTorch Deep Learning Framework. For We have used a combination of Streamlit, Matplotlib, Seaborn and Plotly Python packages for visualization.

ML models: In this section, we discuss the consortium of ML models to gain insights from the data sources and the features covered in Section III-A. This includes creating a series of machine-learning model pipelines that connect with the data sources for continuous learning and an inference pipeline for retrieving the scores. Below is the list of models trained.

- ***Time series: Prophet***
  We utilized Facebook's Prophet model [41] for the time series analysis of token price data, calculating the model's predictive values to identify anomalous patterns. Our evaluation of various time-series models, including ARIMA, LSTMs [42], [43], [44], [45], and GRUs [45], led us to select the Prophet model. This choice was driven by its lesser time consumption and lower computational demands, which effectively counterbalance the resource-intensive nature of our NLP pipelines.
- ***LLM:Finbert***
  We selected the finBert Large Language Model (LLM) [46] from the Hugging Face model repository [38], specifically trained on financial data, to analyze finance-related news and discussions on social media. We further refined finBert to tailor it to the unique content of DeFi projects. Our approach included developing a model to monitor social media sentiment regarding DeFi projects. Additionally, we trained another model on the same dataset to detect patterns characteristic of previous scam projects on social media.
- ***Gradient Boosting: XGBoost***
  For classification problems of transaction data [47], we used XGBoost to train the transaction data to get the prediction score for the transactions on the DeFi network. We chose the Gradient Boosting algorithm XGBoost for its capability to deal with large amounts of data efficiently in less time.
- ***LLM:CodeBert***
  Microsoft's CodeBert [48] shows proficiency in various programming languages. We utilized the embeddings from this LLM model to explore semantic similarities in code, represented in n-dimensional vector space. CodeBert, which is based on the BERT architecture (a transformer-based method for NLP tasks), is trained across multiple programming languages, aiding in code search and summarization.
- ***Hierarchical Clustering: Agglomerative algorithm***
  We continue the pipeline of smart contract data embeddings by grouping them into clusters based on the distance metric using [49] hierarchical Unsupervised clustering algorithm. We opted for this unsupervised over others for its ability to treat the object as a single cluster, which works well as each smart contract is unique and shares very few properties with other smart contracts.

B. Evaluation Metrics

We measure the performance of the trained ML models using the performance metrics. $Accuracy(Acc) = \frac{TP+TN}{TP+TN+FP+FN}$, $Precision(Pre) = \frac{TP}{TP+FP}$, $Recall(Rec) = \frac{TP}{TP+FN}$, and $F1 = 2 \times \frac{Pre \times Rec}{Pre+Rec}$. Here, TP(True Positive) and TN (True Negative) denote correct identifications of positive and negative instances. FP(False Positive) and FN(False Negative) represent misclassifi-

cation. For the feedback loop, we visualize the feature importance metrics to gain insights into the model's prediction scores.

C. Results

Below are the details of the ML model evaluation results and how they are interpreted further to generate the overall Rug Pull Scam prediction score.

Figure 5 illustrates the sample sentiment classification results obtained using the finBert NLP model. We employed Seaborn's Violin plot to represent these results, with positive polarity percentage values plotted along the x-axis against the content displayed on the y-axis. The underlying model for this graph is finBert, specifically pre-trained on financial datasets and further fine-tuned using cryptocurrency-related news for our research. Content exhibiting negative polarity influences the DeFi project's suspicion score. Moreover, the scores derived from machine learning models analyzing social media content account for 10% of the total evaluation score.

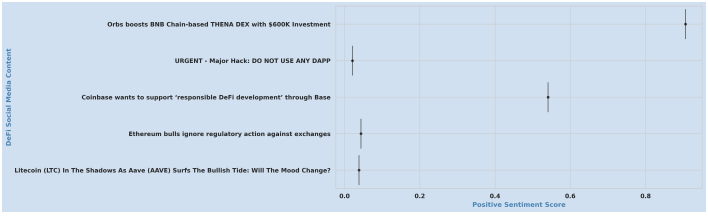

Fig. 5. FinBert Model Classification

Figure 6 highlights the results of the transaction data ML models, a supervised learning solution with the XGBoost model. The monitoring logic configured to identify the fraudulent transactions from DeFi projects transactions is protocol-specific, as are the ML models and results. The plot represents the evaluation metrics of supervised machine learning algorithms, which implies an optimized prediction score from the models. The scores from ML models of Transaction Data contribute to 30 % of the overall score.

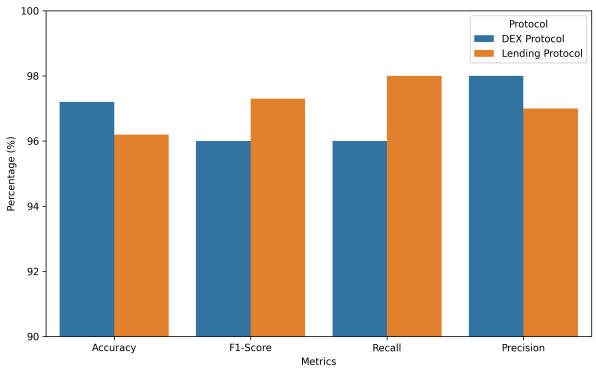

Fig. 6. Transaction Data ML model Evaluation Metrics

Figure 7 illustrates the results of a sample DeFi project (AAve); We converted the prediction score and actual score of the price data to determine the anomaly data point. Facebook's Prophet, a time series model, is used to predict the future price value of the token based on the historical trend of the fluctuations. The Figure represents the model's future predicted value of the price, the actual price, and the derived anomalies. The scores from ML models of Transaction Data contribute to 20 % of the overall score.

Figure 8 represents the clusters or the grouping of the Smart Contracts embeddings data created using the agglomerative clustering algorithm. Each Smart Contract Embedding is generated using the codeBert LLM model. The embedding of natural

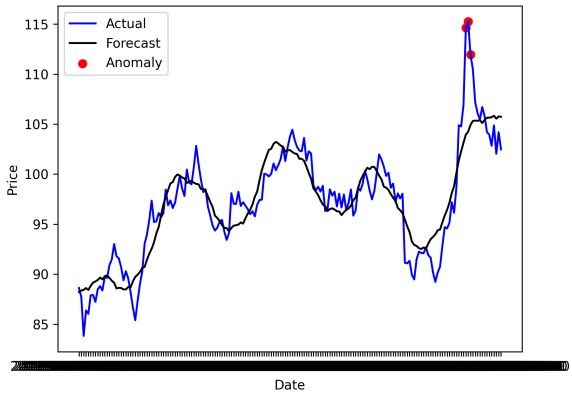

Fig. 7. Anomalies in a DeFi Token Price

language in this programming language gives us the privilege of analyzing the data in an n-dimensional vector space and deriving semantic similarity between the data points. We have used t-SNE (t-distributed Stochastic Neighbor Embedding) to reduce the dimension of Smart Contract Source code embeddings to 2-D to plot the groups or clusters determined by the distance metric of the agglomerative algorithm. The Smart Contracts that are semantically similar to existing scam Smart Contracts are marked with a high suspicion score. The scores from ML models of Transaction Data contribute to 40 % of the overall score.

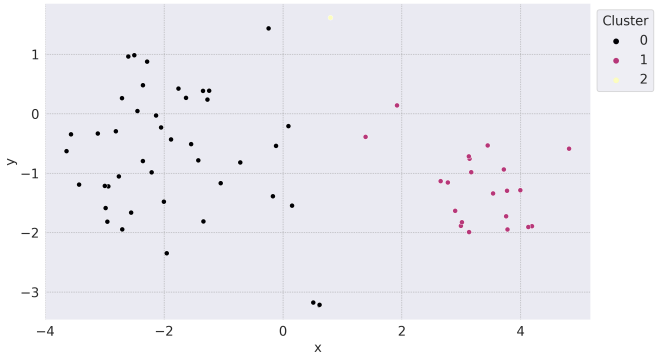

Fig. 8. Smart Contracts Clusters with codeBert Embeddings

We should note that no comparable peer approaches were found for direct comparison with our work. This may be due to the novelty of the field and our unique use of multiple machine learning pipelines to detect DeFi scams, making direct comparisons challenging.

V. Conclusion and Future Work

In this research, we tackled the critical problem of scams in DeFi projects. Our work provides a comprehensive analysis of relevant data points essential for evaluating the credibility and safety of DeFi projects. A significant aspect of our research is the integration of a feedback loop pipeline with the ML model. This structure ensures that the insights generated by the ML model are consistently reviewed and validated through human intelligence. Combining automated ML insights with human oversight is critical to prevent inaccurate analyses and keep the model updated with the latest trends and data. Such a dynamic and responsive system is pivotal in cultivating trust among investors and project owners in the DeFi space. In our future work, we plan to enhance the smart contract audit process by integrating an automated code review pipeline. This upgrade will provide developers with continuous feedback, aiding them in avoiding the deployment of vulnerable smart contracts.

REFERENCES

[1] D. A. Zetzsche, D. W. Arner, and R. P. Buckley, "Decentralized finance (defi)," *Journal of Financial Regulation*, vol. 6, pp. 172–203, 2020.

[2] C. R. Harvey, A. Ramachandran, and J. Santoro, *DeFi and the Future of Finance*. John Wiley & Sons, 2021.

[3] T. A. Xu and J. Xu, *A Short Survey on Business Models of Decentralized Finance (DeFi) Protocols*. Springer International Publishing, 2023, p. 197–206. [Online]. Available: http://dx.doi.org/10.1007/978-3-031-32415-4_13

[4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Cryptography Mailing list at https://metzdowd.com*, 03 2009.

[5] K. Shah, D. Lathiya, N. Lukhi, K. Parmar, and H. Sanghvi, "A systematic review of decentralized finance protocols," *International Journal of Intelligent Networks*, vol. 4, pp. 171–181, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666603023000179

[6] L. Gudgeon, S. Werner, D. Perez, and W. J. Knottenbelt, "Defi protocols for loanable funds: Interest rates, liquidity and market efficiency," in *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, 2020, pp. 92–112.

[7] A. Lehar and C. A. Parlour, "Decentralized exchanges," *Available at SSRN 3905316*, 2021.

[8] J. Xu, K. Paruch, S. Cousaert, and Y. Feng, "Sok: Decentralized exchanges (dex) with automated market maker (amm) protocols," *ACM Comput. Surv.*, vol. 55, no. 11, feb 2023. [Online]. Available: https://doi.org/10.1145/3570639

[9] V. Mohan, "Automated market makers and decentralized exchanges: A defi primer," *Financial Innovation*, vol. 8, no. 1, p. 20, 2022.

[10] Jakub, "How do liquidity pools work? defi explained," 2023. [Online]. Available: https://finematics.com/liquidity-pools-explained/

[11] K. Saengchote, "Where do defi stablecoins go? a closer look at what defi composability really means." *A closer look at what DeFi composability really means.(July 26, 2021)*, 2021.

[12] A. HAYES, "Stablecoins: Definition, how they work, and types," 2023. [Online]. Available: https://www.investopedia.com/terms/s/stablecoin.asp

[13] S. Cousaert, J. Xu, and T. Matsui, "Sok: Yield aggregators in defi," in *2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2022, pp. 1–14.

[14] Crypto.com, "Asset management with defi," 2023. [Online]. Available: https://assets.ctfassets.net/hfgyig42jimx/2BZyBX8zoOIRoySnHtS9pG/7376e66140e3690067b9cd8eae3c770a/Asset_Management_with_DeFi.pdf

[15] F. Schär, "Decentralized finance: On blockchain-and smart contract-based financial markets," *FRB of St. Louis Review*, 2021.

[16] Koinly, "Crypto rug pull guide," 2023. [Online]. Available: https://koinly.io/blog/crypto-rug-pulls-guide/

[17] S. Agarwal, G. Atondo-Siu, M. Ordekian, A. Hutchings, E. Mariconti, and M. Vasek, *Short Paper: DeFi Deception—Uncovering the Prevalence of Rugpulls in Cryptocurrency Projects*, 12 2023, pp. 363–372.

[18] W. Ma, C. Zhu, Y. Liu, X. Xie, and Y. Li, "A comprehensive study of governance issues in decentralized finance applications," *arXiv preprint arXiv:2311.01433*, 2023.

[19] F. Cernera, M. L. Morgia, A. Mei, and F. Sassi, "Token spammers, rug pulls, and sniper bots: An analysis of the ecosystem of tokens in ethereum and in the binance smart chain (BNB)," in *32nd USENIX Security Symposium (USENIX Security 23)*. Anaheim, CA: USENIX Association, 2023, pp. 3349–3366. [Online]. Available: https://www.usenix.org/conference/usenixsecurity23/presentation/cernera

[20] C. Carpentier-Desjardins, M. Paquet-Clouston, S. Kitzler, and B. Haslhofer, "Mapping the defi crime landscape: An evidence-based picture," 2023.

[21] 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," 2021.

[22] W. Li, J. Bu, X. Li, H. Peng, Y. Niu, and Y. Zhang, "A survey of defi security: Challenges and opportunities," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, Part B, pp. 10 378–10 404, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157822003792

[23] T. BLACKSTONE, "Euler finance attack," 2023. [Online]. Available: https://cointelegraph.com/news/euler-finance-attack-how-it-happened-and-what-can-be-learned

[24] X. Sun, S. Lin, V. Sjöberg, and J. Jie, "How to exploit a defi project," in *Financial Cryptography and Data Security. FC 2021 International Workshops: CoDecFin, DeFi, VOTING, and WTSC, Virtual Event, March 5, 2021, Revised Selected Papers*. Berlin, Heidelberg: Springer-Verlag, 2021, p. 162–167. [Online]. Available: https://doi.org/10.1007/978-3-662-63958-0_14

[25] Coingecko, "Defi market cap and volumes," 2023. [Online]. Available: https://www.coingecko.com/en/categories/decentralized-finance-defi#:~:text=DeFi%20or%20Decentralized%20Finance%20refers,in%20the%20last%2024%20hours.

[26] G. Palaiokrassas, S. Scherrers, I. Ofeidis, and L. Tassiulas, "Leveraging machine learning for multichain defi fraud detection," 2023.

[27] 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, 2022. [Online]. Available: https://www.mdpi.com/2227-7390/10/6/949

[28] P. D. Huynh, T. D. Silva, S. H. Dau, X. Li, I. Gondal, and E. Viterbo, "From programming bugs to multimillion-dollar scams: An analysis of trapdoor tokens on decentralized exchanges," *ArXiv*, vol. abs/2309.04700, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:262084504

[29] W. Li, J. Bu, X. Li, and X. Chen, "Security analysis of defi: Vulnerabilities, attacks and advances," in *2022 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2022, pp. 488–493.

[30] T. Sharma, R. Agarwal, and S. K. Shukla, "Understanding rug pulls: An in-depth behavioral analysis of fraudulent nft creators," *ACM Trans. Web*, vol. 18, no. 1, oct 2023. [Online]. Available: https://doi.org/10.1145/3623376

[31] J. Huang, N. He, K. Ma, J. Xiao, and H. Wang, "A deep dive into nft rug pulls," 2023.

[32] J. Su, X. Lin, Z. Fang, Z. Zhu, J. Chen, Z. Zheng, W. Lv, and J. Wang, "Defiwarder: Protecting defi apps from token leaking vulnerabilities," in *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2023, pp. 1664–1675.

[33] W. Zhang, Z. Zhang, Q. Shi, L. Liu, L. Wei, Y. Liu, X. Zhang, and S.-C. Cheung, "Nyx: Detecting exploitable front-running vulnerabilities in smart contracts."

[34] M. Dotan, A. Yaish, H.-C. Yin, E. Tsytkin, and A. Zohar, "The vulnerable nature of decentralized governance in defi," in *Proceedings of the 2023 Workshop on Decentralized Finance and Security*, 2023, pp. 25–31.

[35] F. Jiang, K. Chao, J. Xiao, Q. Liu, K. Gu, J. Wu, and Y. Cao, "Enhancing smart-contract security through machine learning: A survey of approaches and techniques," *Electronics*, vol. 12, no. 9, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/9/2046

[36] A. Trozze, B. Kleinberg, and T. Davies, "Detecting defi securities violations from token smart contract code," *arXiv preprint arXiv:2112.02731*, 2021.

[37] Y. Huang, T. Zhang, S. Fang, and Y. Tan, "Deep smart contract intent detection," *arXiv preprint arXiv:2211.10724*, 2022.

[38] Huggingfaces, "Huggingfaces," Hugginfaces Pretrained Models Repository, 2023. [Online]. Available: https://huggingface.co

[39] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," 2016.

[40] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan, and I. Stoica, "Ray: A distributed framework for emerging ai applications," p. 561–577, 2018.

[41] S. J. Taylor and B. Letham, "Forecasting at scale," *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2018.

[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[43] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 3, 2000, pp. 189–194 vol.3.

[44] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," *Google*, 2014.

[45] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated-learning-based anomaly detection for iot security attacks," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2545–2554, 2022.

[46] D. Araci, "Finbert: Financial sentiment analysis with pre-trained language models," 2019.

[47] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. ACM, Aug. 2016. [Online]. Available: http://dx.doi.org/10.1145/2939672.2939785

[48] Z. Feng, D. Guo, D. Tang, N. Duan, X. Feng, M. Gong, L. Shou, B. Qin, T. Liu, D. Jiang, and M. Zhou, "Codebert: A pre-trained model for programming and natural languages," 2020.

[49] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," *arXiv preprint arXiv:1109.2378*, 2011.