

An AI Multi-Model Approach to DeFi Project Trust Scoring and Security

Viraaj Mothukuri*, Reza M. Parizi*, James L. Massa*, Abbas Yazdinejad†

* Decentralized Science Lab, College of Computing and Software Engineering, Kennesaw State University, USA

vmothuku@students.kennesaw.edu, rparizi1@kennesaw.edu, james.l.massa@gmail.com

† Cyber Science Lab, School of Computer Science, University of Guelph, Ontario, Canada

ayazdine@uoguelph.ca

Abstract—Rampant scams plague decentralized finance (DeFi) projects, creating a DeFi credibility problem that limits the impact of DeFi advances in the availability and variety of financial services. This paper presents a novel solution to the DeFi credibility problem by developing an AI multi-model that generates *TrustScore* ratings for DeFi projects and clear explanations of the scores. We generate DeFi-project *TrustScore* by aggregating multiple factors that provide DeFi investors with a holistic view of DeFi project trustworthiness. To rate a DeFi project with a *TrustScore*, we combine the output of four AI pipelines that analyze smart contract code vulnerabilities, suspicious transactions, anomalous price changes to smart contracts, and social media scam sentiment. Applying four factors exponentially improves the trust-score accuracy over the single-factor approaches done historically. Two of the factors, anomalous price change, and social media sentiment, have not been used before to detect DeFi fraud. Furthermore, we enhanced the most critical factor, smart-contract code vulnerability detection, with the latest Large Language Models (LLMs). Our overall system is a multi-model composed of a *TrustScore* Explainer LLM that aggregates individual pipeline results, a fine-tuned GPT model to audit smart contract code, the Prophet forecasting tool, FinBERT tailored for financial Natural Language Processing (NLP), and XGBoost for classification. The proposed approach identifies a significant proportion of known fraudulent DeFi projects and generates an accurate and explained *TrustScore*. Thus, we address the DeFi credibility problem so that investors can make reliable decisions about DeFi projects.

Index Terms—DeFi, Decentralized Finance, Security, Scam Detection, Rug Pull, AI, LLMs, GPT.

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 programs hosted on blockchains that orchestrate transactions and data flows based on financial agreements. This automation reduces the cost of managing the traditional centralized bank or financial institution. DeFi projects aim to make financial services accessible to a broader audience, and the transparency and immutability of blockchain transactions instill a security-by-design mindset among users of these digitized and decentralized financial services.

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] is 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 *Yield 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 presents a snapshot of the DeFi ecosystem, highlighting some key components and interconnections. It illustrates various DeFi protocols and platforms, including lending services, DEXs, and stablecoin mechanisms. It also depicts the interplay between these elements, showcasing how they collectively contribute to the broader DeFi landscape.

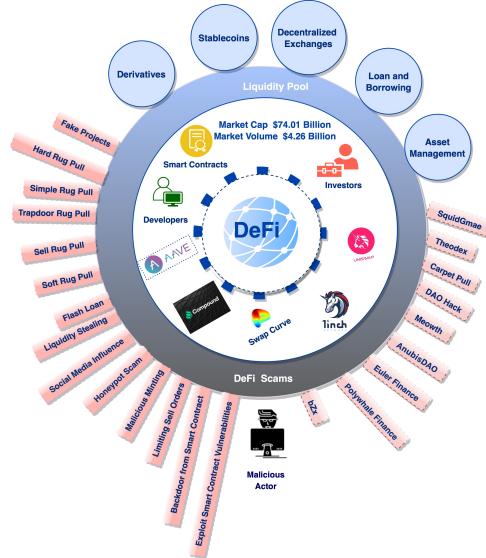


Fig. 1. DeFi Landscape

While the promise of high returns entices numerous users to DeFi, making it a lucrative marketplace for trading and investment, it is also increasingly becoming more vulnerable to modern scams and rug pulls, as indicated in various studies [16], [17], [18], [19], [20], [21], [22] and depicted in Table I. These findings emphasize the growing security concerns within DeFi protocols. Among the major categories of scams, baseline scams in DeFi projects include *Exit Scams*, where the scammers apply various techniques to disappear with the funds of the DeFi project. *Rug Pull Scams*, very prevalent in the Crypto world, are the scenarios where the project owners 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 investors' assets. 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 by the project's developers or external parties exploiting weak coding or security vulnerabilities. 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 transfer investor funds to a single address covertly.

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, including 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.

Table I provides an overview of major scams in the DeFi landscape, illustrating specific instances where the inherent vulnerabilities of DeFi protocols were exploited. Each entry in the table specifies a distinct DeFi scam incident, detailing the mechanisms of the scam, such as exit scams, rug pulls, liquidity thefts, and smart contract exploits. A few other scams are BALD, Magnate Finance, Poly Network Hack, Uranium Finance, Thodex Centralized Exchange, Meerkat Finance and Apes. These examples serve as case studies highlighting the critical need for enhanced security measures in smart contracts and more rigorous code audits within DeFi projects. By documenting the scam type, the method of execution, and the financial impact on investors, this table highlights the dual aspects of innovation and risk in the DeFi sector, reinforcing the importance of due diligence and investor awareness in this rapidly evolving financial space.

A. Motivation

Our goal is to evaluate the probability a DeFi project is either a scam or an unintentionally vulnerable project. By rating DeFi projects with a *TrustScore*, we create accountability for the DeFi project owners. By discovering and reporting DeFi project issues to the public, we help them avoid scams, and conversely, we provide recognition to trustworthy projects. By explaining the *TrustScores*, in layman's terms, we make the *TrustScores* actionable for consumers. For example, a well-explained *TrustScore* helps consumers select a DeFi project that aligns with their needs and concerns.

As of recent data, the market capitalization of DeFi cryptocurrencies is valued at approximately \$93.68 billion, accompanied by a substantial 24-hour trading volume of around \$5.76 billion. [25]. The DeFi sector, characterized by its novelty and lack of stringent regulation, has become increasingly susceptible to many

TABLE I
TOP DEFI LOSSES OVER THE YEARS

DeFi Scam Incident	Description
Squid Token Scam	With Play to earn strategy, where the token had only a buy option, and the investors are left with unworthy tokens. Scammers earned \$3.38 million from the Liquidity pool.
Fantom Foundation	Incidents involving security vulnerabilities and mismanagement within the Fantom ecosystem, leading to financial losses.
Stars Arena	Possible Scam or loss related to deceptive practices, including rug pulls or smart contract exploits.
BIGTIME Token	Incident involving the BIGTIME token, potentially a rug pull or a significant exploit resulting in investor losses.
Linea (Fake Token)	Creation and sale of a fraudulent token, misleading investors with a worthless or non-existent asset.
AnubisDAO	Rug pull or significant exploit in the AnubisDAO project, resulting in substantial financial losses for investors.
DeFi100 coin	Rug pull, smart contract exploit, or devaluation due to mismanagement or fraudulent activities within the DeFi100 coin ecosystem.
Luna Yield	Scam or significant financial loss in Luna Yield, possibly due to smart contract vulnerabilities or deceptive practices.

security threats, including sophisticated hacks, scams, and fraudulent activities aimed at siphoning digital assets. These incidents pose direct financial risks and significantly impair decentralized finance's overall financial stability and resilience. The escalating frequency and complexity of scams in DeFi emphasizes the urgent need for a comprehensive understanding of the inherent vulnerabilities and potential threat vectors. Establishing robust security measures and trust in emerging DeFi services is critical to enable 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 and facilitating informed and safer investment decisions. This, in turn, could significantly boost investor confidence and promote a safer, more robust DeFi ecosystem. By employing a comprehensive strategy that utilizes a combination of LLMs and ML models, including fine-tuned GPT-3.5 for smart contract audits, XGBoost for transaction data analysis, Meta's Prophet model for time-series analysis of token prices, and FinBERT for sentiment analysis from social media, our research aims to elevate scam detection to a more sophisticated level. This contribution seeks to enhance the trustworthiness, resilience, and long-term stability of the DeFi sector, furthering the body of knowledge in this field.

B. Contributions

- **TrustScore rating for DeFi projects:** Our research utilizes a holistic Multi-Model ML framework that aggregates data from four AI pipelines. Each AI pipeline looks at the DeFi credibility issue from a different view/data point, enabling

comprehensive, accurate monitoring of DeFi projects. This framework combines various machine learning models, including LLMs, to deliver a nuanced perspective on DeFi security and integrity, aiding in the proactive identification of fraudulent activities.

- **Innovative Use of LLM and NLP in Smart Contract Audits:** Previously, smart contract vulnerability has not been analyzed with fine-tuned personalized LLMs. We fine-tuned Open AI GPT3.5 with the domain expertise of vulnerabilities in smart contracts. We further enhance the LLM with analytics from the static smart contract audit tool. Fine-tuned Open AI GPT 3.5, our approach brings a new dimension to smart contract audits. This integration of LLM and NLP technologies significantly improves the detection of vulnerabilities and potential fraudulent patterns in smart contracts, thereby enhancing the efficacy and thoroughness of the auditing process.
- **Application of BERT for Social Media Insight Extraction:** We employ advanced NLP models, particularly BERT, to analyze sentiments from social media discussions about DeFi projects. The purpose of including social media sentiment is twofold. First, some fraudulent activity is specifically conducted through social media, e.g., market manipulation by driving investor sentiment on social media. Second, although our multi-model may not specifically address every trustworthiness factor, social media sentiment is likely to incorporate most other factors as part of the group knowledge. As soon as anyone knows anything for any reason and posts about it, our pipeline incorporates it to derive insights from similar patterns.

The structure of this paper is as follows. Section II reviews existing literature, setting the context for our work. Section III outlines our proposed approach, covering its architecture, implementation, and internal mechanisms. Section IV assesses our approach, offering a discussion of our results and their relevance. Finally, Section V concludes with a summary of our findings and suggestions for future research, pointing the way forward in enhancing DeFi security.

II. RELATED WORK

The research landscape addressing fraud and scams in DeFi [26] 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. [27] 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 [28] 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 [29] 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. [30], 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 [31], 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 [32], [33] 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 gained 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 [34], 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. [35] 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. Research work in [36] proposes a real-time attack detection system called BLOCKEYE, which attempts to identify the DeFi project's attacks through automated security analysis, transaction monitoring, and critical invariant configuration. Moreover, Dotan et al. [37] 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.

Furthermore, several recent key studies in the field of DeFi scams have made valuable contributions to understanding fraudulent patterns in DeFi projects. Our research aims to build upon and extend these findings in several ways. The research in [38] provides a helpful, systematic overview of DeFi attacks, and our study seeks to expand on this by incorporating extensive machine learning for dynamic and real-time scam detection. The valuable work by [39] explores AI-powered fraud detection across the DeFi project lifecycle, and we hope to complement their methodology by employing a diverse, multi-model ML framework to broaden and deepen the analytical scope. The research work of [40] uses of open-source tools for investigating DeFi frauds is insightful, and our study aims to further enhance this approach by leveraging advanced NLP and LLM techniques to extract additional insights from various data sources. The [41]'s innovative merging of GPT models with program analysis for detecting vulnerabilities in smart contracts is noteworthy, and we seek to extend this by including a wider array of data sources, such as social media sentiment analysis, to develop a

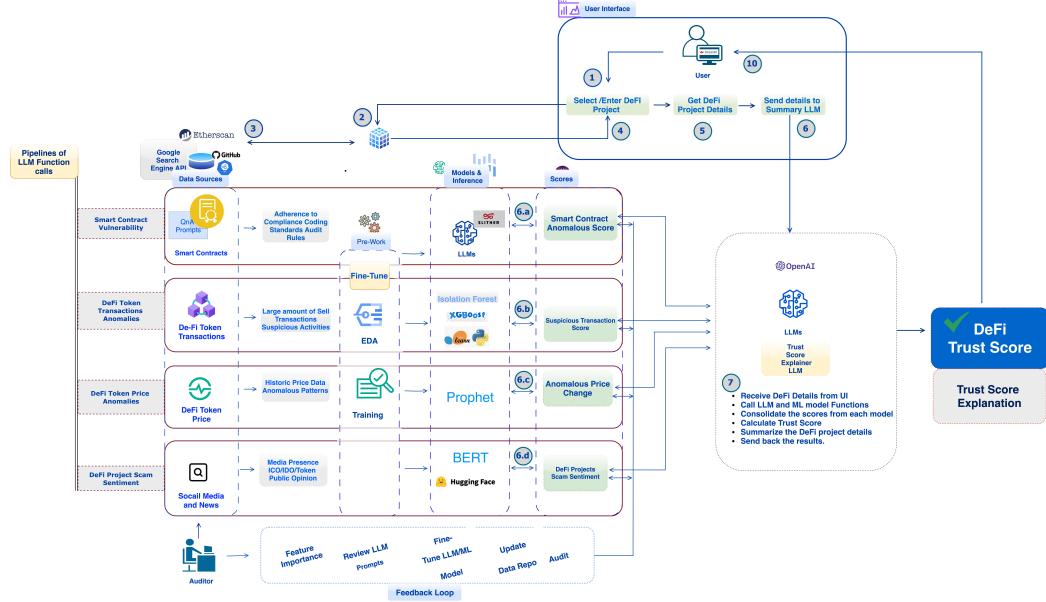


Fig. 2. Architecture

comprehensive scam detection strategy. Finally, the [42]’s creative approach in generating attack surfaces for smart contracts is commendable, and our study aims to build upon this by covering a broader spectrum of DeFi scams, including analyzing transaction data and price anomalies.

In direct comparison, our study brings together a diverse range of data sources, including detailed social media analyses and transaction records, and integrates a spectrum of sophisticated ML techniques. By employing tools like GPT3.5, XGBoost, Prophet, and BERT, we offer a more holistic and proactive solution in identifying and mitigating DeFi scams, thereby filling the gaps left by the aforementioned studies and pushing the boundaries of current DeFi scam detection methodologies. Existing research has greatly enhanced our understanding of DeFi fraud and scams, particularly in smart contract analysis and token-specific scams. However, a comprehensive and dynamic approach to scam detection is still needed. Our study addresses this by introducing a multifaceted scam detection framework that combines advanced machine learning, including LLMs for smart contract audits and NLP for sentiment analysis. This integrative approach broadens the scope of scam detection, leveraging real-time data from diverse sources to offer a more robust and adaptable solution for the DeFi sector’s evolving challenges.

III. PROPOSED APPROACH

A. Architecture

We take a holistic approach to calculating a *TrustScore* for a DeFi project. The *TrustScore* of a DeFi project is a dynamic metric that evolves throughout the project’s lifecycle. Initially, during the pre-launch phase, the *TrustScore* is determined by a comprehensive assessment of the smart contract code data. However, from the moment the project officially launches, and its smart contract code is deployed on the blockchain, the *TrustScore* calculation undergoes a significant shift. From this point forward, the score is derived based on the data associated with the project. We analyze all the major data sources that indicate DeFi project risk and derive patterns from historic scams of DeFi projects. Specifically, we analyze the smart contract code for vulnerabilities, the smart contract transactions for suspicious

transactions, anomalous price changes, and DeFi project sentiment to see if people suspect the project is a scam.

We ingest the derived features into a training pipeline for a consortium of LLMs and ML models to learn the pattern in data and gain insights. The combination of the most powerful LLMs with fine-tuning applied and ML models outperforms the pre-existing methodologies. The architecture diagram in Figure 2 illustrates our data collection process for DeFi projects that encompasses Smart Contract Source Code, Token Price Data, Token Transaction Data, and Social Media Content. Another crucial *data repo* we use contains historical records of fraudulent smart contracts and transactions, along with detailed descriptions of the associated malicious activity. We compile the data in the repo by analyzing historical patterns of fraudulent activities in the DeFi sector. Subsequently, we apply data analytics methodologies to re-validate the analysis through a combination of programmatic and manual code logic verification, assigning labels to each dataset. We then develop a series of training pipelines for ML models to detect patterns and derive features and labels using anomaly detection algorithms. Our *Smart contract audit LLM* is powered by the GPT3.5 model, which is fine-tuned for enhanced code review. Anomalies in *Transaction data* and *Market data* of the DeFi project token are reviewed using XGBoost and Prophet models. Additionally, we employ FinBERT to extract sentiment and insights from *social media* data. The social media and news data extraction of the DeFi project is performed using *Google Search Engine API*. The inference pipeline with *TrustScore Explainer LLM* powered by GPT-4 LLM [43], consolidates the results from all the models into a *TrustScore* and provides insights on the Score with a summary of vulnerabilities and potential scams on the reviewed DeFi project data. The designed *feedback loop* provides information for the *Auditor* to monitor LLMs and ML model scores and incorporates human intelligence into the training pipeline. We discuss details of each component in the below sections.

1) Data Sources, Collection, and Feature Extraction: Our methodology involves a structured data collection and feature extraction approach, as illustrated in Figure 2. We begin by analyzing smart contract data from a comprehensive list of

100+ DeFi projects [44] on Ethereum (the full list with their characteristics can be found [here](#)). The chosen projects offer a diverse representation of decentralized finance protocols, ranging from decentralized exchanges to yield optimization platforms. Their selection reflects their prominence, innovation, and significance within the DeFi ecosystem, providing rich material for analysis and comparison. This initial phase involves identifying and labeling data as suspicious or non-suspicious based on the historic fraudulent transactions on smart contracts. We harness four diverse data sources to analyze decentralized finance (DeFi) projects. We rely on tools for data collection, including the Etherscan API and Github. 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 datasets include:

Smart Contracts of DeFi Projects: We focus on smart contracts from DeFi projects (some samples shown in Table II) with diverse protocols selected for this research. Recent reports, such as those referenced in [45], and social media updates demonstrate how frequently large-scale scams exploit flaws in smart contracts of DeFi projects.

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

DeFi project scam Sentiment on Social media and Internet: 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 using Google search API. We extract the DeFi project details using search words such as project name and project description and tag the results to the project. From the search results, we filtered out the irrelevant information and considered the web page titles for sentiment analysis of the project. For the data's reliability, we prefer well-established news sources with a strong reputation for accuracy, objectivity, and rigorous fact-checking processes. By prioritizing such reputable outlets, we can ensure that the information gathered is credible and trustworthy.

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

TABLE II
SAMPLE DEFI PROJECTS (FOR FULL LIST, REFER TO THIS [LINK](#))

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

The data collected is further processed to extract the important features (Table III) that can be ingested into the training pipeline of LLMs and ML models. Below are the steps for dataset collection and feature extraction.

- Use the DeFi projects registered and audited on Ethereum.
- Scrap the publicly available smart contracts address details of Scam projects.
- Extract the smart contract source code using Etherscan API.

- Extract the transactions related to the smart contracts.
- Extract the transactions with the addresses reported as fraudulent on Social Media.
- Scrap the publicly available data of Crypto and DeFi project social media discussions and content.

2) *DeFi Transaction Data LLMs and ML models:* In this section, we discuss the details of the training pipeline built with the data sources, the features extracted and derived, and ingested as training data for AI models. The historical transaction data of fraudulent DeFi projects serves as an exemplary pattern for verifying and calculating the *TrustScore* of a DeFi project. This model serves as the tool to monitor the transaction data and report suspicious transactions with low *TrustScores*.

In the threat model for Lending and Borrowing Protocol Transactions, the key malicious actors include Flash Borrowers, who engage in short-term loans, and Collusion Attackers, who manipulate interest rates and asset prices. Similarly, the threat model mirrors those found in traditional finance (TradFi) or centralized finance (CeFi) exchanges for DEX Protocol Transactions. 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, necessitating different feature sets; Table III details the transaction features unique to the Lending and Trading aspects of DeFi projects. The malicious activities or the transaction records that could leave a project underfunded are:

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

We use a combination of classification(XGBoost) and unsupervised learning algorithms (Isolation Forest) to analyze the data and identify suspicious transactions.

3) *Smart Contract Audit LLM:* Using ML for smart contract vulnerabilities is a critical area of research [46]. The historic scams review has emphasized the importance of Smart Contracts audits, which would have prevented the occurrence of many scams [47], [48]. 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 and lack of compelling standards of compliance and risk assessments, makes the entire DeFi vulnerable to attacks and incurs losses. The smart contract's ability to perform activities such as sudden token transfer to a single address and the smart contract's ability to draw the entire token amount can signify a potential 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 a less exploit-free environment. Our LLM-powered audit outperforms ML-based audit due to its human level of understanding of code data, and the fine-tuning job enables it further to specialize in identifying the vulnerabilities in smart contract code. Furthermore, we empower the model to the next level by providing static tool slither analysis results. The static analyzers are computationally inexpensive as they look over the code without executing and try to identify the patterns of known vulnerabilities of smart contracts [49].

Our approach uses the following steps for the LLM pipeline to derive the *TrustScore* and insights on the DeFi project based on smart contracts code quality.

- Select the environment to build the LLM pipeline and set up the OpenAI GPT3.5 [50] API for the inference pipeline.
- Fine-tune the GPT3.5 model with the information on smart contract vulnerabilities.
- Pre-process the smart contract dataset (Table III).
- Run the data through a static analyzer and create results.
- Feed the smart contract feature dataset to the LLM pipeline and capture the results.
- Review and process the response and derive the *TrustScore*. Which will be used further to calculate the overall *TrustScore* of the DeFi project.

4) DeFi Token Price Anomaly Detection LLMs and ML models: The anomalous price change is a trend to look out for suspicious activity in any crypto-linked projects. Along with DeFi 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 most cases, in the DeFi 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 calculate the price change percentage as a feature for the ML model. We applied Meta's time-series analytic model Prophet [51] to detect the anomalous pattern in the price change of the Token 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.

5) Social Media and News LLM: DeFi project discussions on social media and news have established a known pattern for 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 projects that are flagged suspicious by other investors.

We applied the FinBERT model [52] to extract insights from public opinion through social media and news data on the internet and the APIs. We train the LLM model to identify the familiar pattern of the previous DeFi Scam project's social media presence. Additionally, we train an LLM model to capture the sentiment of social media on the DeFi Projects. We generate features (Table III) 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 module and calculate the weighted average for the overall *TrustScore* of the DeFi project along with detailed insights from LLMs on DeFi project scam prediction.

The details of the inference pipeline are listed in Algorithm 1. The *TrustScore Explainer LLM* (GPT-4 LLM) collects the data from data repo/the data source APIs and initiates the function call [53] to the individual modules and collects the *TrustScore* from each model. The collected information is analyzed further, and the summarized insights are sent to the user along with the *TrustScore*. This represents the overall prediction of the

possibility of a project being a scam. We derive the overall DeFi scam prediction score by performing a weighted average of the scores from each LLM and ML model. We assign the weightage of 0.4 for the smart contract audit model, 0.3 for the DeFi project transaction model, 0.2 for the DeFi price change anomalies model, and 0.1 for the social media sentiment model. The weightage is assigned based on the impact of the data on the DeFi project vulnerability score. The smart contract audit model has a weightage of 40% as the smart contract code is the showstopper component of the DeFi project, and the probability of attacks or scams is high with the smart contract as the source. In the algorithm, Lines 20-25 represent the *SummaryLLM* function, which serves as the central component of the inference pipeline. This function is responsible for calling the four main functions: *analyze_contract_LLM* (Lines 5-9), *Audit_SmartContractCode* (Lines 10-13), *TokenPriceData_Anomalies* (Lines 14-15), *TokenTransactionData_Anomalies* (Lines 16-17), and *NewsAndSocialMedia_Anomalies* (Lines 18-19). The *SummaryLLM* function returns a consolidated summary, which includes insights from each of the four functions, along with the calculated overall *TrustScore*.

C. Feedback loop & Data Repo

We gather insights from human intelligence, the Auditor who regularly reviews scores from ML models and LLMs and insights about DeFi projects. This process enables us to reassess the importance of features to keep the LLMs prompts up-to-date. Monitor the model performance degradation and schedule a fine-tuning job with the latest information.

IV. EVALUATION AND RESULTS

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

A. Experimental setup

We established a development environment utilizing Google Colab Pro paired with a T4 GPU, featuring 160 GB of hard disk space, 50 GB of system RAM, and 15 GB of GPU RAM. Additionally, we utilized a MacBook Pro equipped with a GPU, boasting a 500 GB hard disk and 32 GB of RAM, along with OpenAI and Hugging face APIs. Data extraction and preprocessing were conducted using Python 3.10.12, while training and inference pipelines were built using the PyTorch Deep Learning Framework. Visualization was achieved through a combination of Streamlit, Matplotlib, Seaborn, and Plotly Python packages. We design the evaluation of our approach by integrating the data source collection, the trained model's inference pipeline, and a user interface for the overall evaluation of the DeFi projects. We execute batch scripts to get individual results from each model for Model-specific evaluation for comparative purposes. The project artifacts, encompassing datasets, LLMs and ML model code, and the user interface are available on the project's [GitHub page](#).

Model Details: Below is the consortium of ML and LLM models trained to gain insights from the data sources and the features covered in Section III-A1. This includes creating a series of ML model pipelines that connect with the data sources for continuous learning and an inference pipeline for retrieving the scores. Additionally, we augmented the model's capabilities with insights from a static analysis tool (Slither [54]).

• AI Assistant: GPT3.5, GPT

We utilized the fine-tuned GPT-3.5 model [50] for auditing smart contract code, aiming to identify vulnerabilities and assess the trustworthiness of the DeFi projects. In our fine-tuning process, we employed the prompts (a sample of them

Algorithm 1: Inference Pipeline

Input: DeFi Project token address
Output: *TrustScore* of DeFi Project with detailed insights from LLM Model

```

1 wcode = 0.4      /* Weightage for smart contract code audit */
2 wprice = 0.3      /* Weightage for transaction data anomalies */
3 wtrans = 0.2      /* Weightage for token price anomalies */
4 wnews = 0.1      /* Weightage for news and social media */
5 Function
    analyze_contract_LLM(DeFiSourceCode):
6     while .sol in DeFiSourceCode do
        /* Assess smart contract for vulnerabilities and fraud indicators */
7         slitherAnalysis_Results =
            reviewWithSlither(.sol)
8         Trustcode =
            analyze_contract(.sol, slitherAnalysis_Results)
9     return Trustcode
10 Function
    Audit_SmartContractCode(DeFiSourceCode):
11    while .sol in DeFiSourceCode do
        /* Assess smart contract for vulnerabilities and fraud indicators */
12        TrustScorecode, Vulnerability_analysis =
            analyze_contract_LLM(.sol)
13    return TrustScorecode, Vulnerability_analysis
14 Function TokenPriceData_Anomalies(mwi):
    /* Analyze token price data for anomalies over different time windows */
15    TrustScoreprice = detect_price_anomalies(mwi)
    return TrustScoreprice
16 Function
    TokenTransactionData_Anomalies(Mwi):
        /* Analyze transaction data for unusual patterns */
17    TrustScoretrans =
        detect_transaction_anomalies(Mwi) return
        TrustScoretrans
18 Function
    NewsAndSocialMedia_Anomalies(Mwi):
        /* Analyze News and Social Media data */
19    TrustScorenewsnsocialmedia =
        detect_news_and_social_media_anomalies(Mwi)
    return TrustScorenewsnsocialmedia
        /* LLM Function Calling */
20 Function SummaryLLM(Mwi):
    /* call each function and collect insights from each */
21    TrustScorecode, Vulnerability_analysis =
        FAuditSmartContract
22    TrustScoreprice = FTokenPrice(Mwi)
23    TrustScoretrans = FTransData(Mwi)
    TrustScorenewsnsocialmedia =
        FNewsAndSocialMedia(Mwi)      /* Calculate the weighted average of individual Model TrustScores */
24    DeFi_ProjectTrust_Score =
        Trustcode × wcode + Trustprice × wprice +
        Trusttrans × wtrans + Trustnewsnsocialmedia × wnews
25    return ConsolidatedSummary,
        DeFi_ProjectTrust_Score

```

is listed below) with major vulnerabilities in smart contracts for 50 epochs.

Prompts to Fine-tune GPT3.5

■ **prompt:** "Given the following Solidity smart contract code snippet, identify any potential security vulnerabilities such as reentrancy overflow/underflow or improper access control and suggest mitigation strategies.", **completion:** "The code snippet exhibits a classic reentrancy vulnerability in the function 'withdrawFunds()' where external calls are made before updating the state. This can be exploited by an attacker to drain the contract's funds. To mitigate this, the contract should adhere to the Checks-Effects-Interactions pattern, ensuring state updates occur before external calls. Additionally, implementing reentrancy guards or utilizing the 'transfer' method for ETH transfers, which caps gas and prevents reentrant calls, can further secure the contract."

- **LLM:FinBERT**

We opted for the FinBERT Large Language Model (LLM) from the Hugging Face model repository [55], specifically trained on financial data, to analyze finance-related news and discussions on social media. We then fine-tuned FinBERT specifically for DeFi project content. Our methodology involved crafting a model for tracking social media sentiment toward DeFi projects.

- **Time series: Prophet**

We utilized Meta's Prophet model [51] 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, and GRUs, led us to select the Prophet model. This choice was driven by its lesser time consumption and lower computational demands, effectively counterbalancing our NLP pipelines' resource-intensive nature.

- **Gradient Boosting: XGBoost**

For classification problems of transaction data, we used XGBoost [56] 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.

B. Evaluation Metrics

We measure the performance of the trained LLMs and ML models using performance metrics Accuracy, Precision, Recall, and F1-Score with TP (True Positive), TN (True Negative), FP (False Positive), FN (False Negative), and FP (False Positive). Where we use TP and TN to denote correct identifications of positive and negative instances, FN and FP represent misclassification. For the GPT 3.5 model API, we review the model's capability to identify the vulnerabilities in the smart contract code. The measurement values *TP*, *TN*, *FP*, *FN* represent the model prediction value against the ground truth label. *Scoring Calculation:* As detailed in Algorithm 1 and illustrated in Figure 3, we determine the *TrustScore* by computing the weighted average of scores generated by four ML model pipelines. The process initiates with data retrieval for these pipelines, encompassing fetching smart contract code, news/media, and transaction data. Subsequently, this data undergoes analysis, including scanning smart contracts for vulnerabilities, performing sentiment analysis on news/media content, and identifying suspicious transactions. Scores generated by each pipeline are then aggregated to give the final *TrustScore* between -5 to 5. The instructions on the next page are given to the *TrustScore* explainer LLM to consolidate the scores from individual pipelines and generate a score.

C. Results

Our approach takes the DeFi project details as input from the user interface (check out our GitHub for the UI), which is

TABLE III
DATA FEATURES

Feature	Description	Model
Action	Describes the type of transaction or action performed, e.g., 'Buy', 'Sell', 'Swap'.	DEX Trade Transaction
Amount (Out)	Represents the amount of Cryptocurrency or tokens being sent out of the wallet.	DEX Trade Transaction
Token (Out)	Specifies the type of token or Cryptocurrency being sent out of the wallet.	DEX Trade Transaction
Amount (In)	Indicates the amount of Cryptocurrency or tokens received in the transaction.	DEX Trade Transaction
Token (In)	Specifies the type of token or Cryptocurrency received in the transaction.	DEX Trade Transaction
Swapped Rate	Shows the exchange rate or conversion rate at which the tokens were swapped.	DEX Trade Transaction
Swapped Pair	Specifies the pair of tokens being swapped, e.g., 'BTC/ETH', 'USDT/DAI'.	DEX Trade Transaction
DEX	Stands for Decentralized Exchange and indicates the exchange platform where the transaction occurred.	DEX Trade Transaction
is_scam	A boolean value indicates whether the transaction is considered suspicious or a potential scam.	DEX Trade Transaction
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.	Smart Contract
Static Analyzer Results	Slither [54] tool analysis of smart contract.	Smart Contract
blockNumber	The number of blocks in the blockchain.	Transaction Data
timeStamp	The timestamp when the block was created.	Transaction Data
hash	The unique hash identifier for the block.	Transaction Data
from	The sender's address in a transaction.	Transaction Data
to	The recipient's address in a transaction.	Transaction Data
value	The value of credit in the transaction.	Transaction Data
gas	The amount of gas used in the transaction.	Transaction Data
gasPrice	The price of gas in the transaction.	Transaction Data
isError	Indicates if there was an error in the transaction.	Transaction Data
txreceipt_status	The status of the transaction receipt.	Transaction Data
contractAddress	The address of the smart contract.	Transaction Data
gasUsed	The gas used in the current transaction.	Transaction Data
methodId	The unique identifier for the method called in a contract.	Transaction Data
functionName	The name of the function called in a contract.	Transaction Data
is_high_value_scam	Indicates if the transaction is a high-value scam.	Transaction Data
is_frequent_small_scam	Indicates if the transaction is a frequent small scam.	Transaction Data
is_scam	Label that indicates if the transaction is a scam.	Transaction Data
Price Change percentage	Token price change from previous day.	Token Price Data
Community Engagement Metrics	Gathers data on the project's social media presence and other engagement metrics, reflecting community interest and involvement.	News and Social Media
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.	News and Social Media

passed into the pipeline of ML and LLMs models, and the final result is the summary of the *TrustScore* of the project with an explanation to support the derived *TrustScore*.

For smart contract audit LLMs *TrustScore* evaluation and analysis, we have used open-source smart contract datasets [57] with labeled information on the vulnerabilities. Below is the response of Smart Contract LLM, which demonstrates the LLM's ability to identify the vulnerabilities in the smart contract code. To evaluate the ability of our proposed work to detect new scam types, we have selected vulnerabilities such as 'Short Addresses' that are rare and currently undetectable by existing tools. By focusing on these emerging and hard-to-detect vulnerabilities, we can effectively assess the performance and effectiveness of our approach in identifying and mitigating novel scam techniques.

Smart Contract Audit LLM response

- **TrustScore: -3**
- Identified Vulnerabilities:
 - **Vulnerability Type: Short Addresses**
 - Associated Function: sendCoin(address to, uint amount)
 - Explanation: The function 'sendCoin' is vulnerable to the **short address attack**. This attack occurs when an attacker deliberately provides an address that is shorter than 20 bytes. Solidity does not check the length of the input address and will instead read the missing bytes from the next argument (in this case, 'amount'). This can lead to an incorrect transfer of tokens, causing financial loss.
 - Potential Impact: An attacker can manipulate the amount of tokens transferred by providing a shorter address. This can lead to **significant financial losses** for the contract owner or other users.
 - Mitigation: Add a check to ensure that the input address is exactly 20 bytes long. This can be done using the 'require' function in Solidity. For example: 'require(to.length == 20)'. Alternatively, use a safe transfer function from a trusted library like OpenZeppelin, which includes this check.
 - Overall, the smart contract is at **high risk** due to the presence of a **severe vulnerability** that can lead to financial losses.

Figure 4 provides the sample results of the smart contract code model review. As shown in the figure, the vulnerabilities score given by the model based on the fine-tuned knowledge of vulnerabilities. Below is a snapshot of instructions from the whole set that are given to the fine-tuned GPT3.5 model to generate the *TrustScore* for the smart contract code of a vulnerable smart

contract.

Instructions to Fine-tuned GPT3.5 to generate score

■ **"Instructions":** You are a smart contract auditor. Your task is to analyze the provided smart contract code for potential vulnerabilities and assess its overall security posture. Follow these steps:

1. Assign a *TrustScore* to the smart contract based on the presence and severity of identified vulnerabilities. Use a scale of 1 to 5, where:
 - '-1' indicates a low to medium risk with potential risks
 - '-2' indicates a low to medium risk with minor vulnerabilities that have limited impact
 - '-3' indicates a medium risk with vulnerabilities that could lead to some financial or operational impact
 - '-4' indicates a high risk with severe vulnerabilities that could result in significant losses or compromise of the contract
 - '-5' indicates a critical risk with multiple severe vulnerabilities that render the contract unsafe and require immediate attention

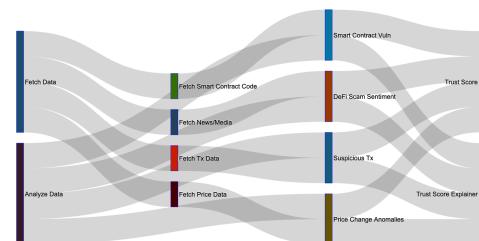


Fig. 3. *TrustScore* Flow Diagram

Figure 5 illustrates the DeFi projects scam sentiment classification obtained using the FinBERT model. The score is plotted on the y-axis for the projects listed on the x-axis. Content exhibiting negative scam sentiment influences the DeFi project's suspicion score. We have collected the news for backdated dates to capture the data for projects reported as scams. Moreover, the scores derived from machine learning models analyzing social media content account for 10% of the total evaluation score.

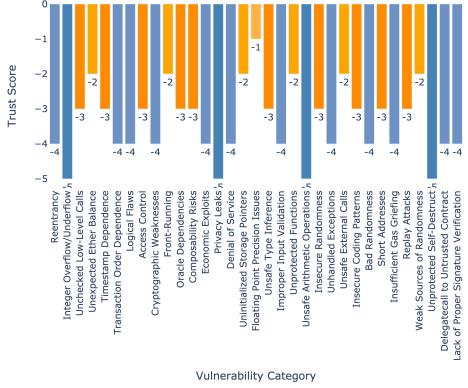


Fig. 4. LLM Smart Contract Audit *Trust Scores*

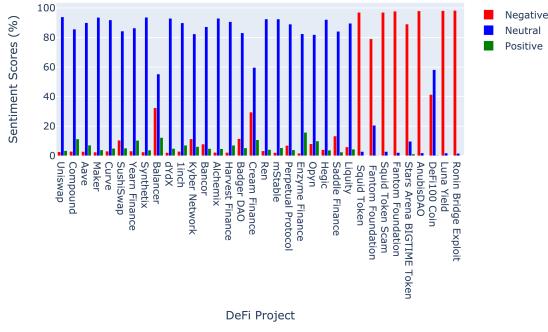


Fig. 5. DeFi Projects Scam sentiment

Figure 6 highlights sample protocol results from the transaction data ML models, a supervised learning solution with the XGBoost model. The monitoring logic configured to identify the fraudulent transactions from DeFi project transactions is protocol-specific, as are the LLMs and 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 the ML models of Transaction Data contribute to 30 % of the overall score.

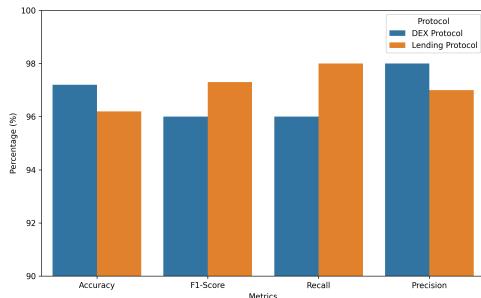


Fig. 6. Transaction Data ML models Evaluation Metrics

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. We use Meta's Prophet, a time series model, 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 change percentage to the actual price and the derived anomalies, where the x-axis is the Date of price change percentage plotted on the y-axis. The The scores from the price change ML model of Transaction Data contribute to 20 % of the overall score.



Fig. 7. Anomalies in a DeFi Token Price (AAve)

D. Comparison with existing works

As we researched the state-of-the-art to find compatible peer approaches, we were unable to find any related work with a multi-faceted approach that inspects the DeFi project with multi-model views. We compare with existing works for only the smart contract audit model as the related works on other dataset models are limited or unavailable and do not directly derive the scam category.

TABLE IV
RESULTS COMPARISON WITH EXISTING WORKS: SMART CONTRACTS LLM MODEL

Research	TP	TN	FP	FN
<i>Our Work</i>	23.75 %	30.0 %	46.25%	0.00 %
[58]	1.64 % (32)	58.12 % (1128)	38.12 % (740)	2.11 % (41)
[41]	7.12 % (40)	55.16% (310)	9.61% (54)	28.11% (158)

We compared our research findings with those of previous studies. In [58], the authors explored GPT3.5 and GPT4 LLMs, while in another study, the GPT3.5 model was utilized ([41]), serving as a basis for our evaluation comparison. Our results, outlined in Table IV, indicate that our solution surpasses existing research outcomes (The value in the closed brackets represents the number of smart contracts reported in each category). Notably, our experimental results data revealed a 0% TN rate, demonstrating that our solution correctly identified all vulnerable instances. Unlike studies such as [59] and [30], which focused solely on individual datasets (Market data and Transaction data of the token), we included the smart contract dataset in our analysis. This comprehensive approach is crucial for identifying vulnerabilities in DeFi projects, as smart contracts play a significant role in this context.

V. CONCLUSION AND FUTURE WORK

In this research, we tackled the critical problem of establishing credibility for DeFi projects by rating them with a *TrustScore* based on the scam indicators in the project. Our work takes a holistic approach to detecting scams by executing

four LLM pipelines in parallel and aggregating the results. Each pipeline looks for scams from a different viewpoint. Bringing to bear multiple viewpoints simultaneously generates a valuable *TrustScore*. As part of our future work, we plan to expand our approach to include other blockchain platforms for DeFi projects, as our current focus is primarily on Ethereum. Additionally, we aim to make the user interface accessible for general use, helping to build trust in DeFi projects before investing.

REFERENCES

- [1] D. A. Zetsche, 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.
- [4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Cryptography Mailing list*, 03 2009.
- [5] "A systematic review of decentralized finance protocols," *International Journal of Intelligent Networks*, vol. 4, pp. 171–181, 2023.
- [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. Lehrer 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.
- [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.
- [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/hfgiyg42jmxm/2BZyBX8zoOIRoySnHtS9pG/7376e66140e3690067b9cdSeac3c770a/Asset_Management_with_DeFi.pdf
- [15] F. Schäf, "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. Atonto-Siu, M. Ordekanian, A. Hutchings, E. Mariconti, and M. Vásek, *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)*. USENIX Association, 2023, pp. 3349–3366.
- [20] C. Carpenter-Desjardins, M. Paquet-Clouston, S. Kitzler, and B. Haslhofer, "Mapping the defi crime landscape: An evidence-based picture," 2023.
- [21] 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.
- [22] "crypto-hall-of-shame," Website, Accessed 2024. [Online]. Available: <https://github.com/nurkiewicz/crypto-hall-of-shame>
- [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.
- [25] Coingecko, "Defi market cap and volumes," 2023. [Online]. Available: <https://www.coingecko.com>
- [26] L. Zhou, X. Xiong, J. Ernstberger, S. Chaliasos, Z. Wang, Y. Wang, K. Qin, R. Wattenhofer, D. Song, and A. Gervais, "Sok: Decentralized finance (defi) attacks," 2023.
- [27] G. Palaiokrassas, S. Scherrers, I. Ofeidis, and L. Tassiulas, "Leveraging machine learning for multichain defi fraud detection," 2023.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] J. Huang, N. He, K. Ma, J. Xiao, and H. Wang, "A deep dive into nft rug pulls," 2023.
- [34] 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.
- [35] 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."
- [36] B. Wang, H. Liu, C. Liu, Z. Yang, Q. Ren, H. Zheng, and H. Lei, "Blockeye: Hunting for defi attacks on blockchain," in *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2021, pp. 17–20.
- [37] 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.
- [38] L. Zhou, X. Xiong, J. Ernstberger, S. Chaliasos, Z. Wang, Y. Wang, K. Qin, R. Wattenhofer, D. Song, and A. Gervais, "Sok: Decentralized finance (defi) attacks," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 2444–2461.
- [39] B. Luo, Z. Zhang, Q. Wang, A. Ke, S. Lu, and B. He, "Ai-powered fraud detection in decentralized finance: A project life cycle perspective," *arXiv preprint arXiv:2308.15992*, 2023.
- [40] A. Trozze, T. Davies, and B. Kleinberg, "Of degens and fraudsters: Using open-source investigative tools to investigate decentralized finance frauds and money laundering," 2023.
- [41] Y. Sun, D. Wu, Y. Xue, H. Liu, H. Wang, Z. Xu, X. Xie, and Y. Liu, "Gptscan: Detecting logic vulnerabilities in smart contracts by combining gpt with program analysis," 2023.
- [42] F. Özdemir Sönmez and W. J. Knottenbelt, "Contractarmor: Attack surface generator for smart contracts," *Procedia Computer Science*, vol. 231, pp. 8–15, 2024, 14th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / 13th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (EUSPIN/ICTH 2023).
- [43] O. et.al, "Gpt-4 technical report," 2024.
- [44] [Online]. Available: <https://defillama.com/>
- [45] V.-B. Pham and T.-D. Trinh, "Analysis model for decentralized lending protocols," in *Proceedings of the 11th International Symposium on Information and Communication Technology*, ser. SoICT '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 405–412.
- [46] 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.
- [47] A. Trozze, B. Kleinberg, and T. Davies, "Detecting defi securities violations from token smart contract code," *arXiv preprint arXiv:2112.02731*, 2021.
- [48] Y. Huang, T. Zhang, S. Fang, and Y. Tan, "Deep smart contract intent detection," *arXiv preprint arXiv:2211.10724*, 2022.
- [49] [Online]. Available: <https://owasp.org/www-project-smart-contract-top-10/>
- [50] T. B. B. et.all, "Language models are few-shot learners," 2020.
- [51] S. J. Taylor and B. Letham, "Forecasting at scale," *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2018.
- [52] D. Araci, "Finbert: Financial sentiment analysis with pre-trained language models," 2019.
- [53] OpenAI, "Llm function calling," 2023. [Online]. Available: <https://platform.openai.com/docs/guides/function-calling>
- [54] J. Feist, G. Grieco, and A. Groce, "Slither: A static analysis framework for smart contracts," *CoRR*, vol. abs/1908.09878, 2019.
- [55] Huggingfaces, "Huggingfaces," Huggingfaces Pretrained Models Repository, 2023.
- [56] 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.
- [57] Z. Liu, P. Qian, J. Yang, L. Liu, X. Xu, Q. He, and X. Zhang, "Rethinking smart contract fuzzing: Fuzzing with invocation ordering and important branch revisiting," *arXiv preprint arXiv:2301.03943*, 2023.
- [58] I. David, L. Zhou, K. Qin, D. Song, L. Cavallaro, and A. Gervais, "Do you still need a manual smart contract audit?" 2023.
- [59] Q. Kong, J. Chen, Y. Wang, Z. Jiang, and Z. Zheng, "Defitainter: Detecting price manipulation vulnerabilities in defi protocols," in *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis*, ser. ISSTA 2023, 2023, p. 1144–1156.