On-Chain Risk Signals: Predicting Security Threats in DeFi Projects

Bahareh Parhizkari
SnT, University of Luxembourg
Luxembourg, Luxembourg
bahareh.parhizkari@uni.lu

Antonio Ken Iannillo
SnT, University of Luxembourg
Luxembourg, Luxembourg
antonioken.iannillo@uni.lu

Ed Zulkoski
Quantstamp, Inc.
San Francisco, CA, USA
ed@quantstamp.com

Christof Ferreira Torres
INESC-ID / IST, University of Lisbon
Lisbon, Portugal
christof.torres@tecnico.ulisboa.pt

Radu State
SnT, University of Luxembourg
Luxembourg, Luxembourg
radu.state@uni.lu

***Abstract*—Blockchain has revolutionized finance through decentralization, eliminating the need for traditional intermediaries. However, security concerns remain a major barrier to adoption, as DeFi platforms increasingly face targeted attacks. In this paper, we present the first methodology for automatically assessing and quantifying the risk of fund loss in DeFi projects due to smart contract exploits. By analyzing on-chain behaviors that signal potential malicious interactions, our approach assigns a dynamic risk score to DeFi projects over time. Relying solely on on-chain data ensures resistance to data manipulation and enhances the integrity of the assessment.**

We evaluated 220 compromised and 200 unaffected DeFi projects on multiple EVM-compatible blockchains – including Ethereum, BSC, Polygon, Arbitrum, Optimism, and Fantom – and conducted a comparative risk assessment on these projects. Our findings reveal statistically significant differences in risk scores before attacks compared to a control group without attacks. We anticipated potential threats to 86% of the projects that were later attacked, one day before the incidents, with a precision of 78%.

***Index Terms*—Ethereum, EVM, Decentralized Finance, Security Risk Scoring.**

I. Introduction

DeFi, short for Decentralized Finance, represents a significant shift in the financial landscape. It encompasses projects that leverage blockchain technology to construct a new, open, and permissionless financial ecosystem [46]. The rapid growth and continual evolution of DeFi protocols over the past few years suggest that this industry is becoming an increasingly central component of the global financial system [43]. As of May 25, 2025, the total value locked (TVL) in the DeFi ecosystem is more than 150 billion USD and increasing [5].

The computational flexibility of blockchain platforms - particularly the Ethereum Virtual Machine (EVM) - has enabled developers to design innovative decentralized financial systems. These systems promise to reshape traditional finance, but their rapid expansion brings significant risks. Many investors remain hesitant to participate in DeFi due to the potential for asset loss. According to Zhou *et al.* [47], between April 2018 and April 2022, security incidents in DeFi led to losses of at least 3.24 billion USD, impacting users, liquidity providers, speculators, and protocol developers. Despite the emergence of numerous security tools for blockchain platforms and extensive efforts to mitigate attacks, malicious actors continue to target DeFi projects monthly. Such attacks do not necessarily indicate shortcomings in existing security tools themselves; but often stem from improper implementation practices, insufficient adherence to secure development principles, or the lack of integrated defensive architecture. Although several existing works focus on post-attack detection, our goal is to examine a project's systemic risk over time. This distinction is crucial since our model produces early warning signals that support strategic prioritization, such as identifying which projects merit deeper audits, closer monitoring, or adjusted insurance premiums. To achieve this objective, we introduce a systematic framework that translates these security insights into measurable, data-driven risk indicators.

Our methodology relies exclusively on blockchain data, making it resistant to external manipulation and independent of centralized reporting sources. By depending solely on decentralized information, our system ensures objective and consistent outputs—essential for a trustless environment. The computed risk score captures abnormal user behaviors, structural weaknesses in the protocol, and other on-chain signals that may precede malicious activity. We adopt the perspective that cyberattacks do not occur spontaneously; rather, they are preceded by subtle phases, including reconnaissance, probing, vulnerability discovery, and strategic execution. Our system identifies these early warning signs and assesses the likelihood that a given DeFi project may become a target of attack. Based on these capabilities, we next discuss how our framework fits into the broader DeFi security ecosystem.

To the best of our knowledge, this work presents the first fully automated risk-scoring methodology for DeFi projects that relies exclusively on on-chain behavioral and structural signals, without any off-chain input or manual data curation. Investors may use the resulting risk score to decide whether to invest in a specific DeFi project. Insurance companies and

intermediaries can rely on it to assess potential collaborations. Project owners can use the score to assess their security posture and take proactive measures to mitigate risks. As such, our method complements existing tools and provides an additional layer within the broader DeFi security stack.

Contributions: Our main contributions are as follows:

- **Development of an Automated Security Risk Assessment Methodology.** We propose a novel methodology to automatically assess DeFi security risks using only on-chain data, reducing external manipulation and ensuring consistent evaluation. By generating granular risk scores, our approach equips investors, project owners, and insurance firms with essential information for decision making, bridging the gap between decentralized finance and informed risk management.
- **Introduction of Unique Risk Metrics.** Our research introduces four novel, interpretable risk metrics derived from on-chain behavior, capturing indicators of both malicious intent and structural vulnerabilities. These chain-agnostic metrics can be dynamically computed without external input, providing stakeholders with deeper insight into project integrity and operational risk for proactive management.
- **Robust Statistical Validation of Risk Assessment.** In addition to our methodology and metrics, we conduct an extensive empirical evaluation of the effectiveness of our risk scoring system. Using a dataset of 220 security breaches across DeFi projects, we demonstrate a clear distinction in risk scores between attacked and non-attacked projects. Our analysis achieves an average recall of 86.4% and precision of 78.5%, underscoring the practical applicability of our risk assessments in real-world scenarios.

II. Background

In this section, we introduce the necessary background on Ethereum, SCs, and decentralized finance.

A. Ethereum and EVM

Ethereum [44] is a popular blockchain platform that supports secure transactions and the development and execution of SCs. The Ethereum Virtual Machine (EVM) is a decentralized virtual machine that enables the execution of SCs on the Ethereum blockchain. This robust and secure execution environment has positioned Ethereum as a leading platform for providing decentralized financial services.

Numerous blockchains leverage the EVM as their foundational technology, facilitating the creation and execution of SCs. In this paper, we consider Polygon [11], Arbitrum [2], Binance Smart Chain or BSC [4], Optimism [10], Fantom [8], and Base [3]. Ethereum accounts are uniquely identified by hexadecimal addresses. Interactions occur through transactions which act as messages sent between accounts and which may include cryptocurrency as well as data. An essential aspect of transactions is "gas", which refers to the unit of

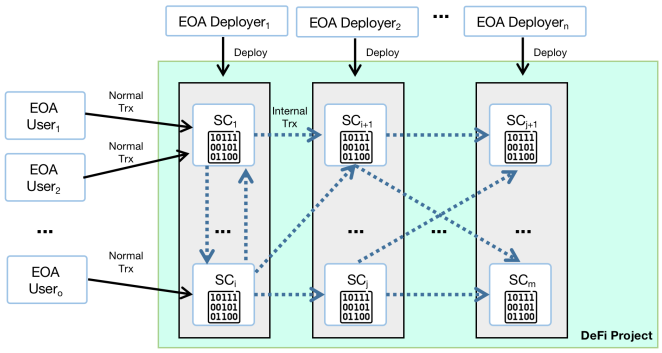

Fig. 1. Structure of a DeFi project and the interaction between deployer EOAs, user EOAs, and SCs. Solid lines denote normal transactions; dotted lines indicate internal transactions.

measurement that denotes the computational effort required to execute operations, including transactions and SC executions. Each transaction requires a certain amount of gas, and users must pay for this gas in the form of the native cryptocurrency, e.g., Ether (ETH) for Ethereum.

B. EOAs and SCs

Ethereum features two types of account: Externally Owned Accounts (EOAs) and smart contracts(SCs). EOAs, controlled by private keys, serve as the origin of all transactions on the Ethereum blockchain. They can initiate transactions that involve complex interactions with other accounts, such as mathematical computations, transferring assets, and deploying SCs. Importantly, only EOAs can initiate transactions; the account that triggers a transaction is referred to as the *transaction origin*. SCs consist of executable code stored on the blockchain that operates autonomously according to the rules of the blockchain, independent of the EOA that has implemented them. Although SC cannot independently initiate transactions, it can perform internal operations as part of a regular transaction. These internal transactions allow one SC to request services from another or interact with EOAs. This mechanism facilitates complex interactions within the Ethereum ecosystem.

C. Decentralized Finance Projects

DeFi, short for Decentralized Finance, refers to blockchain-based services that offer various financial products [15]. These services provide blockchain users with access to financial instruments, such as lending, borrowing, and crypto exchanges [21]. These services are on-chain through SCs.

A DeFi project typically comprises multiple SCs, from tokens to governance contracts, that interact via internal transactions to deliver the intended financial services. Figure 1 illustrates the communication and interactions between different actors in a DeFi project. The scope of a project encompasses several SCs, all of which are deployed by a set of EOAs serving as project deployers. The idea of using deployer addresses to collect a project's SCs is inspired by the findings of [36], which show that the victims of a single

attack can often be identified as all SCs deployed by the same deployer. End users interact with the project by initiating a normal transaction to any of its SCs. If the transaction meets the SC's constraints, it will complete the process by initiating internal transactions within other SCs of the project. Since most DeFi projects are built on public blockchains, they are transparent and open-source. This transparency benefits users, as anyone can inspect the SC's code before interacting. However, it also presents many risks [18]. With full access to the source code and deployed contracts, malicious actors can identify vulnerabilities, exploit them, and potentially steal funds. Given the decentralized nature of blockchains and the lack of centralized authority to reverse transactions after a hack, proactive security measures are essential. They help mitigate risks and prevent potential misuse of DeFi platforms.

III. Challenges

In this section, we examine the complexities of risk assessment in DeFi projects, highlighting key challenges, especially when compared to traditional financial systems.

Lack of Regulation and Standardization. One of the biggest challenges in assessing risk in DeFi projects is the lack of regulatory oversight and standardization [35], [14]. Unlike traditional financial systems, which operate under well-established regulations enforced by government bodies, DeFi works in a largely unregulated environment [38]. This lack of oversight increases risk, as there are no uniform guidelines to ensure the security and reliability of DeFi platforms. Another critical issue is clear gateways and interoperability between blockchain networks. In traditional finance, transactions and communications typically go through regulated intermediaries that provide oversight and trust. In contrast, DeFi operates in a borderless digital landscape where protocols interact directly, often without standardized interfaces. This lack of defined gateways can lead to unexpected protocol interactions, further increasing risk exposure.

Complexity of DeFi projects. DeFi projects rely on SCs, often comprising multiple interdependent components [43]. The intricate nature of these SCs poses significant challenges in risk assessment. In traditional finance, risk is typically assessed using established financial instruments and methodologies. However, in the context of DeFi, the unique characteristics of SCs complicate the auditing and verification process. Manual code reviews are labor-intensive, and automated tools, while useful, may fail to identify all potential vulnerabilities.

Open-Source Nature and Transparency. While blockchains' open-source nature is often praised as a benefit, they also introduce significant challenges in risk assessment. Although transparency allows users to audit SCs and identify potential vulnerabilities, it also allows malicious actors to easily access this information [39]. Cybercriminals can scrutinize the code to pinpoint weaknesses, making it easier for them to exploit vulnerabilities. Additionally, the abundance of publicly accessible code increases the likelihood that other projects may inadvertently replicate flawed design choices, further compounding security risks across the ecosystem.

IV. Methodology

This section outlines the proposed methodology to assess and quantify the security risks associated with DeFi projects. These metrics are derived from attacker behaviors and potential vulnerabilities, representing the likelihood of risk. Our approach is built to prevent exploits by attackers who operate entirely on-chain, and interact with deployed SCs. We focus on capturing this behavior through interpretable metrics, rather than modeling off-chain threats such as phishing, private key leaks, or insider fraud. The attackers typically carry out their operations in the following sequence: First, they begin by scanning SCs to discover potential vulnerabilities. Then, they interact with the target project using newly created, anonymous EOAs to probe possible weaknesses. Finally, they proceed to develop and deploy a malicious payload and test it on the project, attempting to penetrate different layers of the system and gradually piece together the components of their attack strategy. These operations carried out by the attackers often leave observable traces in public blockchain data.

A. Pipeline Overview

As depicted in Figure 2, the risk calculation methodology consists of calculating the risk score based on four different risk metrics. The risk assessment begins with three key input parameters:

- **Project Name:** The identifier of the DeFi project being assessed.
- **Date:** The date of the risk assessment serves as a reference point for the analysis.
- **Chain Name:** The blockchain on which the project is deployed (e.g. Ethereum, Polygon).

The pipeline has four distinct stages: data extraction, risk metrics computation, normalization of these metrics, and their subsequent aggregation. The output is a risk likelihood, a numerical value between 0 and 1 indicating the likelihood that the specified project is the victim of an attack on the given date. Providing a threshold as an additional input maps the resulting risk to a risk label, high-likelihood, or low-likelihood.

B. Data Extraction

Our risk assessment system begins by extracting data directly from the blockchain. First, it defines a five-day time window that ends on a specified input date. Within this window, data relevant to the DeFi project is extracted directly from the blockchain using two primary channels. The first channel consists of RPC endpoints on various blockchains, while the second channel involves internet-based block explorers that process on-chain transactions and provide them via API. For the RPC endpoints, we use QuickNode [12], Alchemy [1], and Erigon [7]. A comprehensive list of block explorers used in the data extraction process includes https://etherscan.io/ (Ethereum), https://bscscan.com/ (BSC), https://polygonscan.com/ (Polygon), https://arbiscan.io/ (Arbitrum),

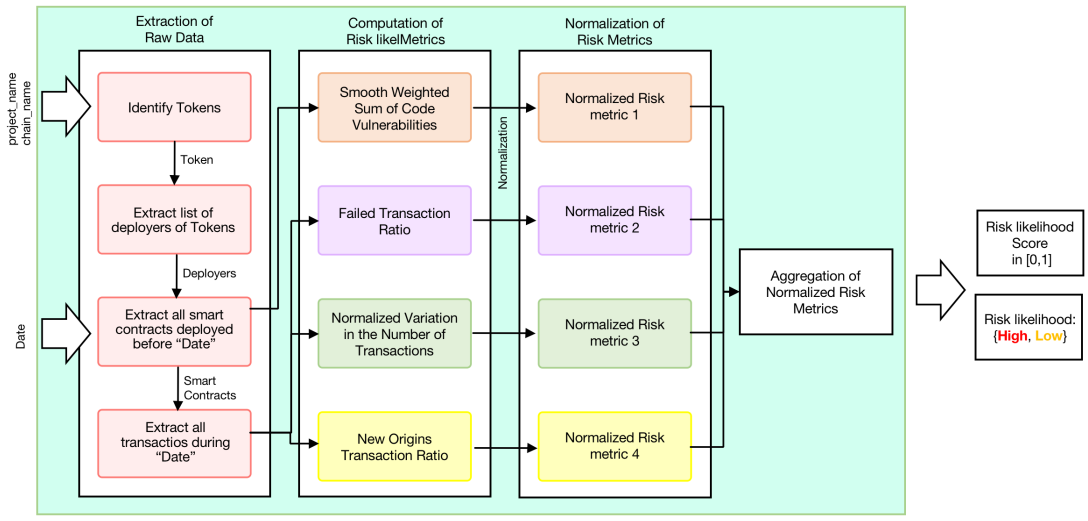

Fig. 2. The Complete DeFi Security Risk Assessment Methodology Pipeline

https://ftmscan.com/ (Fantom), https://optimistic.etherscan.io/ (Optimism), and https://basescan.org/ (Base). For each day within the time window, the system collects and analyzes two types of data:

- **Bytecode.** The bytecode of all SCs deployed before the end of that day.
- **Transactions.** Historical data, including normal and internal transactions involving the project's SCs.

Choosing 24-hour time intervals aligns with attacker behavior involving preparation phases over multiple hours or days, not minutes. Short-term intervals lack adequate data and may introduce noise, but daily granularity yields smoother, more interpretable signals.

To extract the relevant data for each DeFi project, the system begins by identifying the EOAs responsible for deploying the project's tokens, particularly governance tokens, using information from blockchain explorers. For each identified deployer, it collects all SCs they deployed up to the specified input date. The process then gathers both normal and internal transactions involving any of these SCs, focusing on the interactions where the SCs appear as either the sender or the receiver. This is done within a predefined five-day window ending on the input date, allowing for the analysis of transactional activity leading up to potential attacks.

C. Computation of Risk Metrics

After extracting raw data, our methodology involves computing risk metrics. To develop these metrics, we performed an in-depth analysis of various on-chain data points, including the volume of normal and internal transactions, the characteristics and deployment timelines of SCs, their age, bytecode complexity, and the types and features of tokens.

We avoided metrics that might introduce bias toward specific types of projects. For example, we deliberately excluded metrics such as Total Value Locked (TVL) or balance, as these

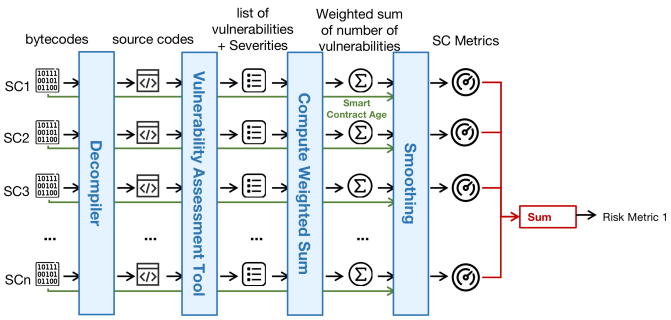

Fig. 3. The pipeline represents the whole process of extracting vulnerability risk score for a single DeFi project.

metrics indicate the impact of the attack rather than the likelihood of its occurrence. To ensure cross-chain applicability, we construct our metrics from generic on-chain signals available across EVM-compatible ecosystems. These include behavioral patterns (e.g., transaction failure rates) and code-level indicators (e.g., vulnerability density). By avoiding project-specific or financial metrics, we enable robust generalization across different blockchain environments.

Based on this comprehensive analysis, we identified four novel metrics that effectively capture and represent security risks in DeFi projects, focusing on the behavioral patterns of vulnerable projects and indicators of adversarial activity. The following subsections introduce and explain these metrics in detail.

*1) **Smooth Weighted Sum of Code Vulnerabilities**:* This risk metric indicates the impact of vulnerabilities discovered in the source code of SCs included in the project code. The risk is directly proportional to the number and severity of vulnerabilities. We used Slither [25], a static vulnerability assessment tool to detect vulnerabilities and their severity. This tool discovers potential vulnerabilities by examining the code and searching

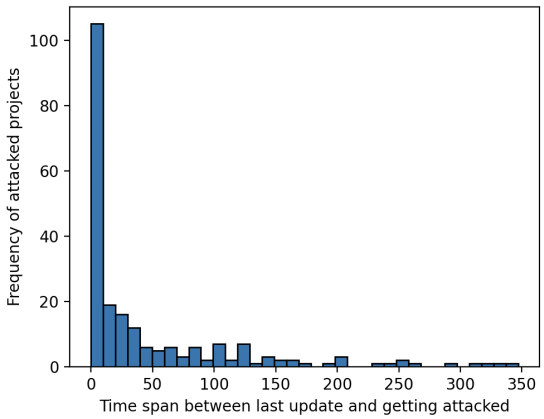

Fig. 4. Timespan between a project's last update and its attack; each bar represents a 10-day interval.

Algorithm 1 Computation of Smooth Weighted Sum of Code Vulnerabilities

1: **procedure** COMPUTEMETRIC1($SCs, Date$)
2: ▷ SCs from Raw Data Extraction
3: $weight \leftarrow \{\text{“}high\text{”}: 7,\ \text{“}medium\text{”}: 4,\ \text{“}low\text{”}: 0.1\}$
4: ▷ Weights of vulnerabilities' severities.
5: $metric \leftarrow 0$
6: **for** each sc in SCs **do**
7: $dd \leftarrow$ GETDEPLOYMENTDATE(sc)
8: $sc\_age \leftarrow Date - dd$
9: $bytecode \leftarrow$ GETBYTECODE(sc)
10: $source\_code \leftarrow$ DECOMPILE($bytecode$)
11: $vulnerabilities \leftarrow$ SCAN($source\_code$)
12:
13: $sc\_metric \leftarrow 0$
14: **for** each v in $vulnerabilities$ **do**
15: $severity \leftarrow v.severity$
16: **end for**
17: $sc\_risk \leftarrow sc\_metric/((sc\_age - 5)^2 + 1)$
18: $risk\_metric\_1 \mathrel{+}= sc\_risk$
19: **end for**
20: **return** $risk\_metric\_1$
21: **end procedure**

for specific vulnerability patterns. Although Slither is used in our implementation, our methodology is scanner-agnostic. The risk score module is adjusted to work with the output of any vulnerability scanner and can seamlessly integrate more advanced tools in future deployments.

As depicted in Figure 3, the calculation of this risk metric begins with the bytecodes of the SCs of the project obtained in the raw data extraction phase. We use Panoramix decompiler [24] to retrieve the source code from these bytecodes, since it is difficult to apply static analysis tools on bytecode than on the source code [31]. However, if an SC has verified source code available on any known trusted block explorers, we prioritize using that verified code over the decompiled version. Afterward, we analyze the extracted SCs using Slither. This process generates a list of vulnerabilities for each contract, complete with detailed descriptions and severity levels for each vulnerability. Severity levels are classified as high, medium, or low. To obtain a single metric for each contract, we compute a weighted sum of the number of vulnerabilities.

Algorithm 1 presents the process for computing this risk metric. The algorithm takes as input the SCs extracted in the previous phase and the date for which we compute the metric. It first initializes the weights for the vulnerabilities at different severities (line 3). We map the weight of each severity to its corresponding minimum risk score on the CVSS qualitative severity rating scale [27]. Then, for each SC, it determines the age of the SC (lines 7-8) and scans for its vulnerabilities (lines 10-11). A weighted sum of these vulnerabilities is calculated (lines 13-18), incorporating the age of the contract. To smooth volatility and prolong the impact of vulnerable contracts, the algorithm uses

$$sc\_risk = \frac{sc\_metric}{(sc\_age - 5)^2 + 1} \tag{1}$$

in this calculation. This formula incorporates three key adjustments. First, as shown in Figure 4, approximately half of the attacks occur within the first ten days after the creation of new SCs. To align with this trend, we introduce a temporal decay function centered on day 5 by subtracting 5 from the contract age. This ensures that the risk score peaks five days after deployment and remains high throughout the initial 10-day high-risk window. We apply a temporal decay function, consistent with prior work modeling time-dependent risk attenuation in security assessment [32]. Second, squaring the term $(sc\_age - 5)$ introduces a smooth decay, allowing the impact of a vulnerable SC to persist over a longer period. Third, adding 1 to the denominator is for preventing division-by-zero errors when $sc\_age = 5$, ensuring numerical stability.

This metric can be evaluated using any state-of-the-art static vulnerability analysis tool. Since assessing SC vulnerabilities is still an open problem, we designed this methodology to leverage existing tools for metric evaluation and mitigate the impact of reported vulnerabilities that are more likely to be false positives.

*2) **Failed Transaction Ratio**:* This risk metric refers to the ratio of failed incoming transactions to the total submitted incoming transactions within a project. Note that transaction fees are not refunded for failed transactions. Therefore, legitimate users naturally seek to avoid transaction failures. Accordingly, a high rejection rate signals the need for a thorough investigation to uncover root causes.

A high rate of failed transactions might be a sign of malicious activities, such as attack attempts. Hackers, upon discovering potential vulnerabilities, need to inspect their exploitability and write malicious payloads to interact with the vulnerable system and exploit its weaknesses. Compromising a complex DeFi project is commonly a multi-step process involving numerous interactions and function calls to both the vulnerable SC and other related contracts within the project.

To successfully exploit the system, a significant number of transactions must be executed and tested. Consequently, an increase in the rejection rate could signal malicious activity targeting the project. This requires executing numerous transactions, and so an increase in the rate of failed transactions could signal that attackers are trying to hack the system.

Even if a high rejection rate does not directly indicate malicious activity, it still points to underlying flaws in the system and how it handles user interactions. These weaknesses might arise from inadequate constraint validation in SCs, creating potential entry points for exploitation by hackers. Therefore, a high failed transaction rate, regardless of its explicit root cause, should always be considered a red flag warranting immediate attention.

Given the two points mentioned above, we have defined a metric for measuring the rate of failed transactions. This metric is calculated by dividing the number of failed transactions in a project within the defined time frame by the total number of transactions in that project. To enhance the accuracy of this metric, we have excluded transactions that failed due to "out of gas" errors, which occur when the gas provided is insufficient to complete the transaction. These failures are commonly a result of the user's unintentional mistakes in estimating gas requirements and are not indicative of underlying system issues. Therefore, including such transactions in the metric could distort the metric, leading to misleading results. By focusing on failures caused by logical issues, we can obtain a more accurate and informative risk metric.

Algorithm 2 describes how we calculate this metric. We get the transactions associated with the project within the defined time frame from the raw data extraction phase. Then, the status of each transaction is inspected. If a transaction's status is not failed, we skip it(lines 5-7). Otherwise, we check the error message of the failed transaction. If the error does not indicate an "out of gas" error (lines 8-10), we ignore it. Finally, the ratio of failed transactions to total transactions is calculated.

Algorithm 2 Computation of Failed Transaction Ratio

```
1:  procedure ComputeMetric2(TXs)        ▷ Transactions from Raw Data Extraction
2:      txs_count ← GetLength(TXs)
3:      failed_txs_count ← 0
4:      for each tx in TXs do
5:          if tx.status is not "failed" then
6:              continue                 ▷ Skip to the next iteration
7:          end if
8:          if tx.error is not "out of gas" then
9:              failed_txs_count += 1
10:         end if
11:     end for
12:     risk_metric_2 ← failed_txs_count/txs_count
13:     return risk_metric_2
14: end procedure
```

*3) **Normalized Variation in the Number of Transactions**:* This metric assumes that significant variations in user transaction rates over consecutive days are signs of abnormal activity and could indicate potential threats. Such variations may be a result of malicious activities and an attacker's interactions with the DeFi service, or they could be attributed to changes in the system or related systems, which could potentially attract attackers' attention to analyze the SCs of the project.

Algorithm 3 computes this metric. This algorithm gets the total number of normal incoming transactions to the project on days N and $N-1$(lines 3-6). Then, it uses:

$$risk\_metric_3 = \frac{|l - l\_p|}{l\_p + 1} \tag{2}$$

to compute a normalized variation in the number of transactions. The numerator represents the absolute difference between the numbers of transactions on days N and $N-1$. The denominator represents the normalization factor. The "+1" ensures that the denominator is never zero, which avoids division by zero errors.

*4) **New Origins Transaction Ratio**:* To maintain anonymity and evade forensic analysis, attackers often use newly created EOAs for their malicious operations. These accounts are typically involved not only in the attack itself but also during earlier testing phases. Using new EOAs helps them to obscure the link between their actions and true identities. According to Study [37], attackers commonly use newly created EOAs as their primary interface for malicious activities.

Algorithm 3 Computation of the Normalized Variation in the Number of Transactions

```
1: procedure ComputeMetric3(Project, Date, TXs)
2:                          ▷ Transactions from Raw Data Extraction
3:     prevDate ← Date − 1
4:     TXs_p ← ExtractTXs(Project, p_Date)
5:     l ← GetLength(TXs)
6:     l_p ← GetLength(TXs_p)
7:     risk_metric_3 ← abs(l − l_p)/(l_p + 1)
8:     return risk_metric_3
9: end procedure
```

Algorithm 4 Computation of New Origins Transaction Ratio

```
1:  procedure ComputeMetric4(Date, TXs)
2:                          ▷ Transactions from Raw Data Extraction
3:      new_origin_cnt ← 0
4:      for each tx in TXs do
5:          if Date−CreationDate(FirstTx(tx.from)) < 20 then
6:              new_origin_cnt += 1
7:          end if
8:      end for
9:      risk_metric_4 ← new_origin_cnt/tx_cnt
10:     return risk_metric_4
11: end procedure
```

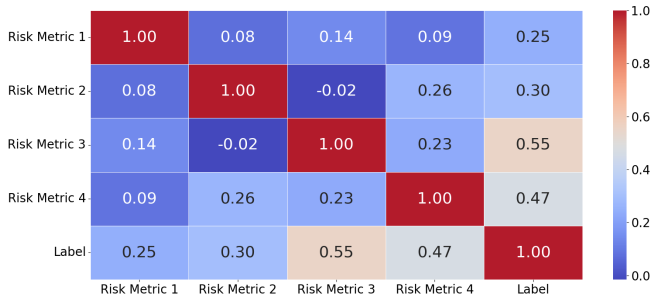

Fig. 5. Correlation between risk metrics and label of expected outcome.

As explained in Section II-B, only EOAs can initiate transactions on EVM-compatible blockchains and EOAs are the sole origin of transactions within EVM-compatible blockchains. The creation of a new EOA starts by generating a new public-private key pair and initiating the first transaction with the newly created private key on the blockchain. The age of an EOA is determined by the time elapsed since its first initiated transaction. We classify an EOA as "new" if its age is less than 20 days, as this threshold safely covers almost all known attack timelines. In the dataset of 181 real-world DeFi incidents analyzed by Zhou et. al.[47], 180 cases occurred within 20 days of the deployment of the malicious contract, with only a single case exceeding this threshold; the maximum observed interval was less than 25 days. We present a detailed description of the calculation process for this measure in Algorithm 4. To evaluate this metric, we analyzed all transactions directed toward the project, identifying their origin EOAs and determining whether they fell within our definition of "new" (lines 5-9). We then calculated the proportion of transactions initiated by new EOAs and compared it to the total transaction volume (line 10).

D. Normalization of Risk Metrics

After extracting the metrics, the next step is normalization and standardization of them. Our methodology first considers applying smoothing techniques to reduce the impact of fluctuations and noise over time. We apply the Moving Averages smoothing technique [17]. This method involves calculating the average of metrics over a defined time window. This approach can minimize the impact of short-term spikes, allowing the impact of high-risk behaviors to last longer. For this purpose, the proposed methodology uses the five-day time windows mentioned in Subsection IV-B.

Next, we scale the average to fit within the range of [0, 1] using a Min-Max scaler [34]. For metrics 2 and 4, which are derived from transaction ratios, the Min-Max scaler works well. However, metrics 1 and 3 lack upper bounds, which can cause the Min-Max scaler to compress all values toward 0. To effectively address outliers, we apply winsorization to the data [23]. Unlike simple truncation, winsorization limits extreme values while preserving the data distribution, an important property since large spikes in DeFi activity may reflect genuine behavior rather than noise. In this approach, we perform a 90% winsorization before applying Min-Max scaling to ensure that extreme outliers do not influence the normalization process. In metric 1, the original data range from 0 to 4474.17. After winsorization, the adjusted range is reduced to 0-10.56. Similarly, in metric 3, the original data span from 0 to 774.73, while the range of winsorized data is from 0 to 1.0.

E. Aggregation of Normalized Risk Metrics

Finally, we need a unified score to assess the risk likelihood of the project on the specified date. The risk likelihood is evaluated using the following formula.

$$risk\_likelihood = 1 - \prod_{i=1}^{4}(1 - risk\_metric_i) \quad (3)$$

This formula inputs normalized risk metrics and computes the risk likelihood. It is derived from evaluating the impact of independent risk metrics in CVSS v4.0 [27] and calculates the probability of at least one high metric when considering multiple independent metrics. The formula ensures that the final risk metric is constrained between the lower and upper limits of 0 and 1. Given the inherent transparency and accessibility of DeFi projects, the risk impact structure differs from that of traditional systems. The risk metrics presented in this paper are autonomous and independent. A high value in any of these metrics indicates a high overall risk of the system. This formula provides a realistic and nuanced assessment of potential harmful behaviors considering the synergistic effects of the proposed risk metrics. The formula ensures that a high value for any metric can increase the likelihood of risk without being diminished by other metrics. By considering the aggregated impact of the proposed metrics, the formula can recognize high-risk scenarios that might be overlooked by individual metric analysis.

F. Analysis of Risk Metrics' Sensitivity

Before assessing the final risk criterion, we first evaluate the likelihood and impact of each defined metric in identifying projects at risk. To accomplish this, we use a heatmap that visualizes both inter-metric correlations and their relationship with the label indicating whether a project was targeted by an attack. As shown in figure 5, metrics 3 and 4 demonstrate a notably stronger correlation —over 40%— with the desired outcome, in contrast to metrics 1 and 2. However, despite their lower correlation, metrics 1 and 2 remain essential due to their greater independence from the other metrics, supporting a more balanced and comprehensive risk assessment. The final four metrics were selected from a larger set developed in this investigation, based on their strong correlation with the likelihood of a project falling victim to an attack, while ensuring diversity in predictive factors.

V. Validation and Evaluation

We conducted a comprehensive validation to assess the effectiveness of our proposed methodology in estimating the

likelihood of security risks in DeFi projects. During the validation phase, we determined the optimal threshold for risk classification. The validation process involved comparing the risk likelihoods generated by our metrics for both attacked and non-attacked projects.

For each attacked project, we gathered historical data from a five-day window before the attack and comparable data from five randomly selected consecutive days for non-attacked projects. For each attacked project within a specific quarter, a comparable non-attacked project from the same chain and the same quarter was selected (baseline). This ensures that the projects are matched based on categories and evaluated under similar market conditions, with a consistent five-day window for comparison. Then, we implemented our methodology and computed the risk likelihood for each window, resulting in two datasets of risk likelihoods for the attacks and the baseline.

A. Data Collection and Experimental Design

We identified 220 known security breaches in 207 projects in 7 different EVM-compatible blockchains (Ethereum, Polygon, Arbitrum, BSC, Optimistic, Fantom, and Base) covering various categories, *e.g.,* lending, yield aggregation, and decentralized exchanges (DEX). We collected information from different sources, such as DefiYield rekt db [20], rekt news [13], defiLlama's hack monitoring panel [9], and ChainSec's hack list [6]. They contain different attack types, *e.g.,* code vulnerabilities, flash loans, and reentrancy, all exploiting on-chain vulnerabilities. These categories are detailed in TABLE I. The distribution of projects and attacks across the different blockchains is shown in TABLE II. We also identified 200 projects that have never been known as attacked. All of these projects are recorded as known DeFi projects in [5]. To ensure a fair comparison, these non-attacked projects were selected to match the attacked ones in terms of analysis date, project type, and TVL distributions.

B. Statistical Analysis

To assess the significance of differences in risk likelihoods between the two datasets, we formulate our null and alternative hypotheses:

- **Null Hypothesis (H0).** There is no significant difference in the risk likelihood between attacked and non-attacked DeFi projects.
- **Alternative Hypothesis (H1).** There is a significant difference in the risk likelihood between attacked and non-attacked DeFi projects.

The descriptive statistics indicated a pronounced discrepancy in the risk likelihoods. The average risk likelihood for attacked projects was approximately 0.633 ($SD = 0.287$), while for non-attacked projects, the mean was 0.256 ($SD = 0.171$). Additionally, the median risk likelihood exhibited a similar pattern, with attacked projects scoring a median of 0.643 compared to 0.237 for non-attacked projects. To further explore the distribution of these scores, we executed the Shapiro-Wilk test, revealing significant deviations from normality for both groups ($p < 0.001$). Given this non-normal distribution, we applied the Mann-Whitney U Test, which confirmed a statistically significant difference ($U = 38976.500$, $p < 0.001$) with an effect size of 0.854, indicating a large effect.

TABLE I
NUMBER OF ATTACKS BY CATEGORIES OF ATTACKED PROJECTS AND TYPES OF ATTACKS. **The "Others" attacked projects' category includes Launchpad, DAO, Hedge Fund, AMM, Decentralized Mortgage, Betting Platform, Liquid Restaking, Gambling, Liquidity Protocol, Staking, Portfolio Management, and Token Vesting.*

Project Category	Code Vulnerability	Flashloan	Insider	Logic flaw	Oracle	Price Manipulation	Reentrancy	Total
Yield Aggregator	19	17	1	1	3	2	5	48
DEX	20	13	0	0	2	0	4	39
Token	12	17	0	0	0	1	0	30
Lending	12	6	0	0	8	1	1	28
Stablecoin	3	4	0	3	0	0	1	11
Bridge	8	1	0	1	0	0	0	10
NFT	8	1	0	0	0	1	0	10
Gaming	2	3	0	0	0	0	1	6
Others*	20	10	0	4	0	1	3	38
Total	**104**	**72**	**1**	**9**	**13**	**6**	**15**	**220**

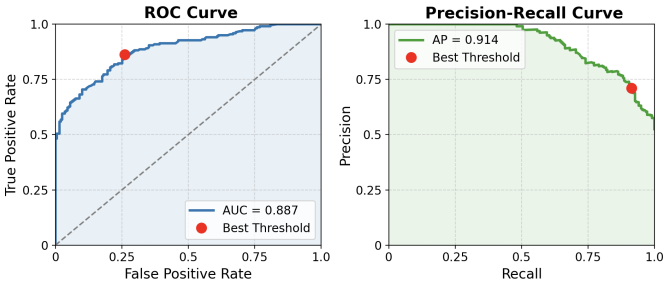

Fig. 6. ROC curve (left) and Precision-Recall curve (right) at various threshold settings. The red point marks the selected threshold in each plot.

C. Classification Performance Metrics

We treated the likelihood of risk as a binary classification problem to evaluate the effectiveness of our proposed risk-scoring methodology. The Receiver-operating characteristic curve (ROC curve) and the precision-recall curve in Fig. 6 visualize the trade-offs available between true positive rates and false positive rates across a spectrum of threshold values. The ROC Area under the curve (AUC) score of 0.887 signifies that the model distinguishes between attacked and non-attacked projects well. A value close to 1 indicates that our risk scoring model is effective in its predictive capacity. We selected the threshold that maximized the F1 score on the evaluation dataset, resulting in a value of 0.364. The F1 score, defined as the harmonic mean of precision and recall, provides a balanced measure that accounts for both false positives and false negatives and reflects the model's optimal performance trade-off to distinguish high-risk projects. We computed various performance metrics summarized in TABLE III based on this threshold.

TABLE II
BREAKDOWN OF ATTACKED PROJECT CATEGORIES AND ATTACK TYPES BY CHAIN

	Ethereum	Polygon	Arbitrum	BSC	Optimism	Fantom	Base
Projects	95	11	11	77	6	6	1
Attacks	101	12	11	82	6	7	1

TABLE III
CLASSIFICATION PERFORMANCE METRICS CALCULATED USING THE THRESHOLD AT THE THIRD QUARTILE.

Metric	Value	Metric	Value
True Positive (TP)	0.864	Precision	0.785
False Negative (FN)	0.136	Recall	0.864
False Positive (FP)	0.261	F1 Score	0.822
True Negative (TN)	0.739	ROC AUC	0.887

The results show that our risk scoring system achieves a recall of 0.864, indicating that approximately 86% of the actual attacked projects were correctly identified as high-likelihood. The false negative rate of 14% highlights areas for potential improvement, suggesting that some attacked projects were not flagged above the threshold, which could lead to a false sense of security for stakeholders. With an F1 Score of 0.822, we achieve a balanced measure that considers both precision and recall, indicating satisfactory model performance.

D. Analysis of Risk likelihood Threshold

The selected threshold of 0.364 guarantees that our model emphasizes high precision at the expense of some recall. By setting this threshold, we assert that the identified high-likelihood projects will likely represent credible threats while minimizing the number of false positives. To validate the stability of our threshold selection, we computed the F1-optimal threshold over progressively larger, chronologically sorted subsets of the dataset. Figure 7 illustrates the progressive evaluation of thresholds in various subsets of data. The plot shows that the selected threshold quickly converges around 0.364, demonstrating robustness to the dataset size. Similarly, Figure 8 shows that the corresponding F1 score also converges, indicating consistent model performance. Together, these results confirm that the classification boundary and predictive quality of the model remain stable as the dataset grows. Identifying an optimal threshold for specific user needs (*i.e.,* prioritizing false negatives or false positives) can guide stakeholders in making well-informed decisions in the DeFi risk landscape.

E. Analysis of High-likelihood Labels by Attack and Project Categories

We further analyzed the distribution of high-likelihood labels among attack and project categories. The results are summarized in TABLE IV, showing the percentage of attacks labeled as high-likelihood across various categories of attacked projects and types of attacks.

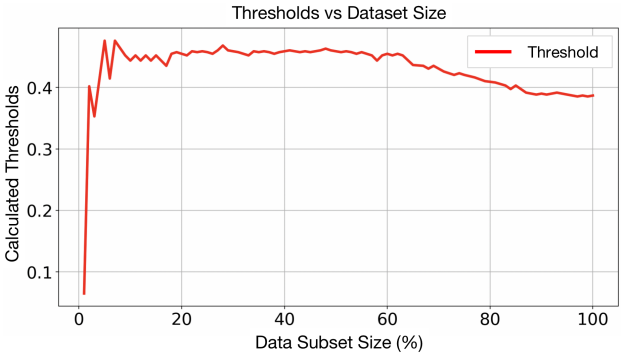

Fig. 7. Progressive evaluation of thresholds on data subsets sorted by date. The plot shows threshold changes as data size grows.

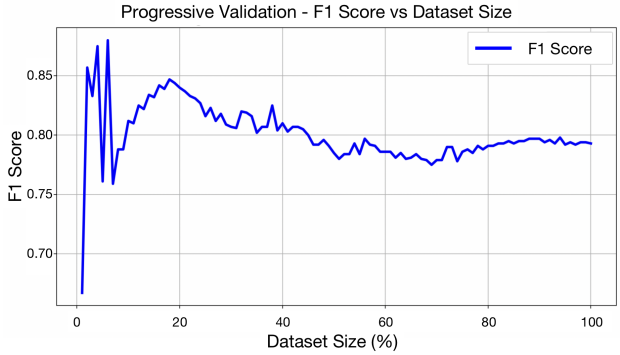

Fig. 8. Progressive evaluation of F1 scores across data subsets sorted by date. This figure evaluates thresholds shown in figure 7.

The data shows that certain project categories exhibit a higher prevalence of high-likelihood label assignments. Notably, Lendings and NFTs demonstrated substantial percentages of attacks classified as high-likelihood, with 96.4% and 100.0%, respectively. This suggests that, despite their critical roles in the DeFi ecosystem, these projects are easier to be labeled as high-likelihood when an attack is imminent. When examining the types of attacks, Code Vulnerabilities and Price Manipulation breaches consistently resulted in high-likelihood designation across most project categories.

F. Evaluation Results

In this section, we evaluate the proposed risk assessment methodology on 17 projects that got attacked between November 2024 and March 2025 and compare it with other 15 projects that never got attacked, in the same period of time. Our goal is to determine whether the risks of these projects could have been assessed and predicted before their occurrence. All of these attacks took place after those introduced in section V-A, making them a suitable benchmark for validating

TABLE IV
PERCENTAGE OF HIGH-LIKELIHOOD LABELED ATTACKS BY PROJECT CATEGORIES AND ATTACK TYPES. *The "Others" attacked projects' category includes Launchpad, DAO, Hedge Fund, AMM, Decentralized Mortgage, Betting Platform, Liquid Restaking, Gambling, Liquidity Protocol, Staking, Portfolio Management, and Token Vesting.*

Project Category	Code Vulnerability	Flashloan	Insider	Logic flaw	Oracle	Price Manipulation	Reentrancy	Total
Yield Aggregator	94.7	70.6	100.0	100.0	100.0	100.0	100.0	87.5
DEX	85.0	92.3	NaN	100.0	50.0	NaN	50.0	82.1
Token	75.0	94.1	NaN	NaN	NaN	00.0	NaN	83.3
Lending	100.0	83.3	NaN	NaN	100.0	100.0	100.0	96.4
Stablecoin	100.0	100.0	NaN	66.7	NaN	NaN	100.0	90.9
Bridge	71.43	NaN	NaN	100.0	NaN	NaN	NaN	70.0
NFT	100.0	100.0	NaN	100.0	NaN	NaN	NaN	100.0
Gaming	50.0	100.0	NaN	NaN	NaN	NaN	100.0	83.3
Others*	85.0	90.0	NaN	75.0	NaN	100.0	66.7	84.2
Total	**87.5**	**86.1**	**100.0**	**77.8**	**92.3**	**83.3**	**80.0**	**86.4**

TABLE V
RESULTS OF THE EVALUATION AND VALIDATION OF THE RISK SCORING METHODOLOGY ON 17 ATTACKED PROJECTS AND 15 NON-ATTACKED PROJECTS BETWEEN 2024/11/01 TO 2025/03/30.

	Attacked		Non-Attacked	
High Risk	12	(70.59%)	1	(6.67%)
Low Risk	5	(29.41%)	14	(93.33%)
Total	17	(100.00%)	15	(100.00%)

our methodology. The results of this evaluation are shown in Table V, indicating that 12 of the 17 analyzed projects were identified as high-risk before the attack, while 5 were classified as low-risk. Meanwhile, among 15 non-attacked projects, only one was labeled as risky, while the other 14 projects were identified as not risky.

Figure 9 shows the average risk likelihood for these 17 attacked projects over 30 days preceding the attack, compared to non-attacked projects during the same period. For attacked projects, the average risk score increases as the attack date approaches. Examining the changes in average risk over time, we find that from one month to 15 days before the attack, the risk levels of attacked and non-attacked projects were generally the same. However, as the attack date approached, the average risk for attacked projects increased and the difference became more noticeable, exceeding the threshold three days before the attack.

VI. DISCUSSION

This research opens a new path to advance the exploration of DeFi security. The outcomes of our work can be valuable not only for current DeFi projects and users but also serve as a cornerstone for future studies focused on evaluating risks in DeFi. In this section, we provide a detailed overview of the practical applications of this research for the DeFi ecosystem and discuss its current limitations.

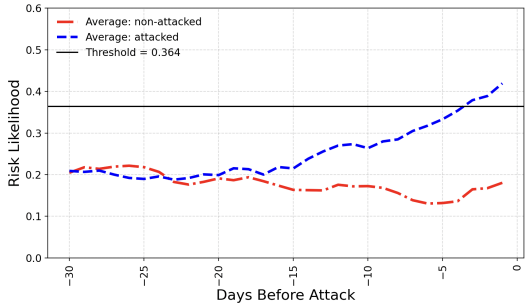

Fig. 9. Average risk similarity over 30 days before attacks vs. non-attacked projects, based on the evaluation dataset.

A. Practical Applications

We introduce a new class of proactive security mechanisms in DeFi, based on predictive modeling of systemic vulnerability rather than transaction-level detection. The findings of this research can benefit DeFi projects, DeFi insurance providers, as well as the users of these projects. The proposed method complements existing tools, adding a distinct layer to the broader DeFi security ecosystem.

B. Limitations

In this paper, we focus only on the likelihood of risk, excluding attack impact, which in DeFi may involve financial loss, service disruption, or reputational damage. During the course of our investigation, we attempted to quantify the impact of attacks but were unsuccessful for two main reasons: 1. DeFi projects experience significant short-term daily fluctuations in metrics such as market cap or TVL, making it difficult to distinguish attack-related changes from ordinary fluctuations. In some projects, these metrics even surpass their pre-attack levels within a short time after the incident. 2. More than half of the attacked projects lack accurate historical data on TVL and market cap. For example, of the 220 projects in our dataset, only 95 had DefiLlama records before being attacked. Calculating impact depends on knowing the value of a project at the time of the incident, and the value of a project is impossible without information on the assets it holds.

Another limitation of our system is that it calculates risk based on 24-hour time intervals, from 00:00 UTC to 23:59 UTC. This limits its ability to assess risk for projects that are launched or updated with exploitable vulnerabilities and then attacked on the same day. Although such cases are relatively rare, they account for a portion of the missed cases in our results. This limitation results from the high volume of transactions in some DeFi projects, making real-time risk calculation infeasible. The issue mainly impacts chains with low transaction fees and, consequently, high transaction frequencies. However, we showed that the risk score noticeably increases as the attack date approaches, and the average risk score exceeds the threshold 3 days before the attack.

Additionally, the model is limited by the scarcity of comprehensive data on attacked DeFi projects. Although many

incidents include details such as the attacker's address, victim, and amount of stolen assets, the type of attack is often unclear, and the available reports sometimes conflict. We manually analyzed all reports on each attack and we did not incorporate attacks in which there was ambiguity regarding the root cause of the attack, as well as those stemming from off-chain sources, such as phishing, private key leakage, and rug pulls. However, while the size of the dataset is limited, its diversity (chains, attack types, and project categories) makes it representative of real-world DeFi risks.

VII. Related Work

Most existing DeFi risk assessment frameworks, such as Gauntlet [30] and Skynet [19] rely on off-chain data, manual reviews, or curated security signals. Although these systems offer valuable insights, they are fundamentally limited by their dependence on centralized data sources and subjective inputs. Weingärtner et al. [41] performed a comprehensive analysis that delineates the principal risks in DeFi projects, distinguishing between systematic and unsystematic risks and introducing *risk wheel*, a tool for navigating these complexities. While these systems offer valuable insights, they depend on centralized data sources, manual interpretation, or subjective scoring. In contrast, our work presents the first fully automated risk-scoring framework that operates exclusively on-chain, using behavioral and structural signals derived from EVM-compatible blockchains. This allows our system to provide tamper-resistant real-time insights without relying on external data or manual intervention.

Another approach to risk assessment in DeFi involves identifying and analyzing potential vulnerabilities within DeFi protocols. For instance, BLOCKEYE [40] is a real-time attack detection system designed to identify potentially vulnerable DeFi projects through an automatic security analysis process. This system performs symbolic reasoning to analyze data flows and assess whether critical service states, such as asset prices, can be manipulated externally. Lanturn [16] uses adaptive learning to assess the economic security of SCs, focusing on threats like frontrunning and value extraction. In contrast, our approach targets software-based attacks and project-level risk by analyzing interaction patterns over time. Other tools such as Foray [42], Ægis [26], and MadMax [31] use static analysis to uncover bugs such as price manipulation, out-of-gas, reentrancy, and access control. However, these approaches operate at the SC level and are typically reactive – identifying specific vulnerabilities post-deployment – rather than providing a predictive risk score. Our system complements these tools, modeling risk at the project level by aggregating behavioral and structural patterns across the blockchain.

Outside of blockchain, current risk-scoring systems methodologies emphasize a systematic approach to identifying, assessing, and mitigating cybersecurity risks, integrating advanced technologies and frameworks tailored to specific industries and organizational needs. One of the key advances in risk-scoring systems is the integration of quantitative risk assessment techniques [33], [22], [28] These methodologies are crucial for developing robust risk-scoring systems that can adapt to the unique challenges faced by different sectors. Systems like the Common Vulnerability Scoring System (CVSS) [27] leverage metrics like complexity, function count, and coupling to evaluate code risk. Additionally, frameworks such as the NIST Cybersecurity Framework [29] provide a structured approach to managing cybersecurity risks across various sectors. The framework facilitates the identification of risks and the implementation of appropriate controls, strengthening organizational resilience against cyber threats. Furthermore, Younis and Malaiya provide a comparative analysis of CVSS and other rating systems, underscoring the importance of quantifiable metrics in assessing software vulnerabilities [45]. These approaches rely on centralized control and trusted infrastructure, making them unsuitable for decentralized systems such as DeFi. Our work adapts metric-based risk assessment to the unique constraints of blockchain systems.

Despite the progress made by existing tools and frameworks for assessing vulnerabilities in DeFi projects, most current metrics emphasize singular aspects of project risk, often neglecting the multifaceted and dynamic nature of DeFi engagements. While tools like BLOCKEYE detect known vulnerability patterns or attack conditions, they do not model project-level risk or forecast future compromise. Our method is complementary and offers early warning signals based on behavioral trends. Using only on-chain data, our system provides a tamper-proof risk metric and considers the complex interactions and behaviors that precede attacks. It also offers a more holistic view of a project's security by incorporating indicators of code weaknesses and user behavior analytics. To our knowledge, no prior work offers a fully automated, on-chain-only framework for dynamic, behavior-based risk scoring of DeFi projects. Our system introduces this paradigm, enabling proactive, data-driven prioritization for audits, insurance, and security interventions across decentralized platforms.

VIII. Conclusion

This study introduces a novel methodology for automatically assessing and quantifying the security risks associated with DeFi projects. The key finding reveals that our risk assessment model can effectively differentiate between attacked and non-attacked projects, achieving an average recall of approximately 86.4% and a precision of 78.5%. This capability has significant implications, offering investors, project owners, and insurance companies actionable insights to make informed decisions and manage risks in the fast-evolving DeFi landscape.

However, this study has limitations. Relying exclusively on on-chain data to evaluate risk likelihood while minimizing external manipulation may overlook off-chain factors that contribute to security vulnerabilities, such as phishing or social engineering tactics. Furthermore, the performance of the model may change with new types of attacks in the future, suggesting that further evaluation is necessary under diverse conditions. Future research should expand upon this work by integrating off-chain data into the risk assessment

framework, enhancing the model's robustness. Furthermore, ongoing refinement of risk metrics, particularly in response to emerging threats, will be essential to maintaining the relevance and utility of the risk assessment methodology over time.

REFERENCES

[1] Alchemy. [Accessed: Apr. 14, 2025]. URL: https://www.alchemy.com/.

[2] Arbitrum. [Accessed: Apr. 14, 2025]. URL: https://arbitrum.io/.

[3] Base. [Accessed: Apr. 14, 2025]. URL: https://base.org/.

[4] Binance smart chain (bsc). [Accessed: Apr. 14, 2025]. URL: https://www.bnbchain.org/en.

[5] Defillama. [Accessed: May. 25, 2025]. URL: https://defillama.com/.

[6] Documented timeline of defi exploits. [Accessed: May. 25, 2025]. URL: https://chainsec.io/defi-hacks/.

[7] Erigon. [Accessed: Apr. 14, 2025]. URL: https://erigon.tech/.

[8] Fantom. [Accessed: Apr. 14, 2025]. URL: https://fantom.foundation/.

[9] Hacks panel in defillama. [Accessed: May. 25, 2025]. URL: https://defillama.com/hacks/.

[10] Optimism. [Accessed: Apr. 14, 2025]. URL: https://www.optimism.io/.

[11] Polygon. [Accessed: Apr. 14, 2025]. URL: https://polygon.technology/.

[12] Quicknode. [Accessed: Apr. 14, 2025]. URL: https://quicknode.com/.

[13] Rekt news. [Accessed: May. 25, 2025]. URL: https://rekt.news.

[14] Olawale Adisa, Bamidele Segun Ilugbusi, Ogugua Chimezie Obi, Kehinde Feranmi Awonuga, Odunayo Adewunmi Adelekan, Onyeka Franca Asuzu, and Ndubuisi Leonard Ndubuisi. Decentralized finance (defi) in the us economy: A review: Assessing the rise, challenges, and implications of blockchain-driven financial systems. *World Journal of Advanced Research and Reviews*, 21(1):2313–2328, 2024.

[15] Raphael Auer, Bernhard Haslhofer, Stefan Kitzler, Pietro Saggese, and Friedhelm Victor. The technology of decentralized finance (defi). *Digital Finance*, 6(1):55–95, 2024.

[16] Kushal Babel, Mojan Javaheripi, Yan Ji, Mahimna Kelkar, Farinaz Koushanfar, and Ari Juels. Lanturn: Measuring economic security of smart contracts through adaptive learning. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1212–1226, 2023.

[17] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[18] Nic Carter and Linda Jeng. Defi protocol risks: The paradox of defi. *Regtech, suptech and beyond: innovation and technology in financial services" riskbooks–forthcoming Q*, 3, 2021.

[19] CertiK. Certik skynet. [Accessed: Apr. 14, 2025]. URL: https://www.certik.com/products/skynet.

[20] DefiYield. Top crypto hacks. [Accessed: May. 25, 2025]. URL: https://defiyield.app/rekt-database.

[21] Saulo Dos Santos, Japjeet Singh, Ruppa K Thulasiram, Shahin Kamali, Louis Sirico, and Lisa Loud. A new era of blockchain-powered decentralized finance (defi)-a review. In *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1286–1292. IEEE, 2022.

[22] Ahmed Abdelwahab Elmarady and Kamel Rahouma. Studying cybersecurity in civil aviation, including developing and applying aviation cybersecurity risk assessment. *IEEE access*, 9:143997–144016, 2021.

[23] Feature Engine. Winsorizer. [Accessed: Apr. 14, 2025]. URL: https://feature-engine.trainindata.com/en/1.8.x/user_guide/outliers/Winsorizer.html.

[24] eveem org. Panoramix. [Accessed: Apr. 14, 2025]. URL: https://github.com/eveem-org/panoramix/.

[25] Josselin Feist, Gustavo Grieco, and Alex Groce. Slither: a static analysis framework for smart contracts. In *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, pages 8–15. IEEE, 2019.

[26] Christof Ferreira Torres, Mathis Baden, Robert Norvill, Beltran Borja Fiz Pontiveros, Hugo Jonker, and Sjouke Mauw. Ægis: Shielding vulnerable smart contracts against attacks. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, pages 584–597, 2020.

[27] FIRST.ORG. Common vulnerability scoring system v4.0: Specification document, 2023. [Accessed: Apr. 14, 2025]. URL: https://www.first.org/cvss/v4-0/specification-document.

[28] András Földvári, Francesco Brancati, and András Pataricza. Preliminary risk and mitigation assessment in cyber-physical systems. In *2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 267–274. IEEE, 2023.

[29] Cybersecurity Framework. Getting started with the nist, 2021. [Accessed: Apr. 14, 2025]. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=932713.

[30] Gauntlet. Gauntlet risk dashboards. [Accessed: Apr. 14, 2025]. URL: https://dashboards.gauntlet.xyz/.

[31] Neville Grech, Michael Kong, Anton Jurisevic, Lexi Brent, Bernhard Scholz, and Yannis Smaragdakis. Madmax: Surviving out-of-gas conditions in ethereum smart contracts. *Proceedings of the ACM on Programming Languages*, 2(OOPSLA):1–27, 2018.

[32] Frank L Greitzer, Rex A Kliner, and Samantha Chan. Temporal effects of contributing factors in insider risk assessment: Insider threat indicator decay characteristics. 2022.

[33] Basil Hamdan. Simulating cybersecurity risk using advanced quantitative risk assessment techniques: A teaching case study. In *Journal of The Colloquium for Information Systems Security Education*, volume 10, pages 5–5, 2023.

[34] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[35] Caroline Malcolm. Defi regulation: Practical next steps to make the industry safer. [Accessed: Apr. 05, 2025]. URL: https://chainalysis.com/blog/defi-regulation-practical-next-steps-to-make-the-industry-safer/.

[36] Bahareh Parhizkari, Antonio Ken Iannillo, Christof Ferreira Torres, Sebastian Banescu, Joseph Xu, and Radu State. Timely identification of victim addresses in defi attacks. In *European Symposium on Research in Computer Security*, pages 394–410. Springer, 2023.

[37] Bahareh Parhizkari, Antonio Ken Iannillo, Christof Ferreira Torres, Joseph Xu, Sebastian Banescu, et al. Beyond the public mempool: Catching defi attacks before they happen with real-time smart contract analysis. In *20th EAI International Conference on Security and Privacy in Communication Networks*, 2024.

[38] Kaihua Qin, Liyi Zhou, Yaroslav Afonin, Ludovico Lazzaretti, and Arthur Gervais. Cefi vs. defi–comparing centralized to decentralized finance. *arXiv preprint arXiv:2106.08157*, 2021.

[39] Julie-Anne Tarr. Distributed ledger technology, blockchain and insurance: Opportunities, risks and challenges. *Insurance Law Journal*, 29(3):254–268, 2018.

[40] Bin Wang, Han Liu, Chao Liu, Zhiqiang Yang, Qian Ren, Huixuan Zheng, and Hong Lei. Blockeye: Hunting for defi attacks on blockchain. In *2021 IEEE/ACM 43rd international conference on software engineering: companion proceedings (ICSE-companion)*, pages 17–20. IEEE, 2021.

[41] Tim Weingärtner, Fabian Fasser, Pedro Reis Sá da Costa, and Walter Farkas. Deciphering defi: A comprehensive analysis and visualization of risks in decentralized finance. *Journal of risk and financial management*, 16(10):454, 2023.

[42] Hongbo Wen, Hanzhi Liu, Jiaxin Song, Yanju Chen, Wenbo Guo, and Yu Feng. Foray: Towards effective attack synthesis against deep logical vulnerabilities in defi protocols. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 1001–1015, 2024.

[43] Sam Werner, Daniel Perez, Lewis Gudgeon, Ariah Klages-Mundt, Dominik Harz, and William Knottenbelt. Sok: Decentralized finance (defi). In *Proceedings of the 4th ACM Conference on Advances in Financial Technologies*, pages 30–46, 2022.

[44] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32, 2014.

[45] Awad A Younis and Yashwant K Malaiya. Comparing and evaluating cvss base metrics and microsoft rating system. In *2015 IEEE International Conference on Software Quality, Reliability and Security*, pages 252–261. IEEE, 2015.

[46] Dirk A Zetzsche, Douglas W Arner, and Ross P Buckley. Decentralized finance. *Journal of Financial Regulation*, 6(2):172–203, 2020.

[47] Liyi Zhou, Xihan Xiong, Jens Ernstberger, Stefanos Chaliasos, Zhipeng Wang, Ye Wang, Kaihua Qin, Roger Wattenhofer, Dawn Song, and Arthur Gervais. Sok: Decentralized finance (defi) attacks. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2444–2461. IEEE, 2023.