Reap the Harvest on Blockchain: A Survey of Yield Farming Protocols

Jiahua Xu and Yebo Feng

Abstract—Yield farming represents an immensely popular asset management activity in decentralized finance (DeFi). It involves supplying, borrowing, or staking crypto assets to earn an income in forms of transaction fees, interest, or participation rewards at different DeFi marketplaces. In this systematic survey, we present yield farming protocols as an aggregation-layer constituent of the wider DeFi ecosystem that interact with primitive-layer protocols such as decentralized exchanges (DEXs) and loanable funds (PLFs)/protocol for loanable funds (PLF). We examine the yield farming mechanism by first studying the operations encoded in the yield farming smart contracts, and then performing stylized, parameterized simulations on various yield farming strategies. We conduct a thorough literature review on related work, and establish a framework for yield farming protocols that takes into account pool structure, accepted token types, and implemented strategies. Using our framework, we characterize major yield aggregators in the market including Yearn Finance, Beefy, and Badger DAO. Moreover, we discuss anecdotal attacks against yield aggregators and generalize a number of risks associated with yield farming.

Index Terms—Decentralized finance (DeFi), yield farming, yield aggregator, simulation, blockchain.

I. INTRODUCTION

YIELD farming protocols are deemed as the decentralized asset managers on blockchain. After having absorbed crypto assets from users—including both retail and institutional investors, yield farming protocols algorithmically deploy those funds into one or more revenue generating services such as lending and market making. Yield farming protocols have become immensely popular as they seem to create a win-win situation: users can earn return on their idle funds through an automated process; yield farming protocols can charge a management fee; other DeFi services can gain more liquidity. As such, on top of the inherently designed benefit that users get for providing liquidity in different kinds of pools (e.g., interest in the case of lending protocols, or fees in the case of providing liquidity in automated market maker (AMM) pools), additional governance tokens are rewarded to users to further encourage their participation in the issuing platform during the early stage of adoption. The basic yield farming idea was born: the search for opportunities in the DeFi ecosystem to generate returns on otherwise dormant crypto assets.

A reaction to the creation of a multitude of platforms returning interests, fees and token rewards, yield aggregators—represented by Yearn Finance, Beefy, and Badger DAO (Table I)—dedicated to farming yield through DeFi primitives emerged. At the beginning 2021, the total value locked (TVL) of DeFi yield aggregators was still shy of 1 billion USD; by May 2021, however, this value grew exponentially to 8 billion USD (illustrated in Figure 1).

In this paper, we present a systemic examination of yield farming protocols. We first inspect yield farming protocols from the perspective of DeFi architecture and posit them as an aggregation-level component that interact with lower-level primitives in DeFi (see Section II). We then synthesize an action-state framework of yield farming operations, and extract yield farming protocols’ features such as pool structure and accepted token types as well as their variations (see Section III). With our established model framework, we characterize top yield

TABLE I

| Yield aggregators       | Governance token | TVL (in USD) | MCap (in USD) | Time established | Tokenholders |
|-------------------------|------------------|-------------|--------------|-----------------|--------------|
| Yearn Finance           | YFI              | 452.07      | 354.96       | 07/2020         | 48.684       |
| Beefy                   | BFP              | 302.45      | 39.00        | 09/2020         | 23.737       |
| Badger DAO              | BADGER           | 187.22      | 47.82        | 11/2018         | 30.777       |
| Idle Finance            | IDLE             | 94.17       | 1.36         | 11/2020         | 3.728        |
| Yield Yak               | YAK              | 72.13       | 5.56         | 09/2021         | 2.208        |
| Aave/Filk               | AAVE             | 64.99       | 26.52        | 02/2021         | 45.074       |
| Pancake/Worm            | PANC             | 59.54       | 86.00        | 06/2020         | 61            |
| Rent Capital            | RENT             | 47.03       | 64.27        | 07/2020         | 4.438        |
| Vesper                  | VESR             | 42.82       | 6.35         | 02/2021         | 5.690        |
| Spook Protocol          | SPook            | 39.42       | 5.99         | 12/2021         | 532           |
| Harvest Finance         | HARVEST          | 32.74       | 37.39        | 09/2020         | 13.847       |
| Algo Capital            | ALGO             | 30.05       | 2.96         | 12/2020         | 8.078        |
| Repetitive Farms        | RF                | 18.21       | 9.25         | 07/2021         | 2.993        |
| Public Finance          | PUB              | 14.20       | 1.49         | 09/2020         | 5.151        |
| Ocean Finance           | OCEAN            | 4.41        | 1.41         | 03/2021         | 2.941        |
| Trilium/Defi             | TQ                | 4.08        | 1.74         | 11/2021         | 852          |
| Solotax                 | SOLIX            | 3.98        | 0.19         | 02/2022         | 9.369        |
| Robo-Vault              | RV               | 3.91        | 0.07         | 07/2021         | 2.119        |

Data fetched on 14/08/2022 from https://defillama.com/ - Yield Aggregators.

Manuscript received 13 September 2022; revised 13 November 2022 and 14 November 2022; accepted 14 November 2022. Date of publication 16 November 2022; date of current version 7 March 2023. The associate editor coordinating the review of this article and approving it for publication was A. Veneris. (Corresponding author: Yebo Feng.)

Jiahua Xu is with the Computer Science Department, University College London, WC1E 6EA London, U.K. (e-mail: jiahua.xu@ucl.ac.uk).
Yebo Feng is with the Computer and Information Science Department, University of Oregon, Eugene, OR 97403 USA (e-mail: yef0@uoregon.edu).
This article has supplementary downloadable material available at https://doi.org/10.1109/TNSM.2022.3222815, provided by the authors.
Digital Object Identifier 10.1109/TNSM.2022.3222815.
II. BACKGROUND IN DEFI

Yield farming protocols are a constituent part of the wider DeFi ecosystem, and operate heavily dependent on other ecosystem components. In this section, we present those related components to understand where yield farming protocols reside within the DeFi ecosystem (illustrated in Figure 2).

A. DeFi Overview

Built on top of decentralized blockchain networks, DeFi [128] systems allow various financial products and services, including lending and asset trading, to be available to the general public. Compared with traditional financial systems, DeFi democratizes finance by replacing legacy, centralized institutions with algorithm-backed protocols, thereby improving the accessibility, inclusion, and transparency of financial services [99], [120].

B. Chain Layer

The distributed ledger technology (DLT) layer forms the infrastructural basis for decentralized applications (dApps). Like all other dApps, DeFi protocols consist of one or more smart contracts deployed on blockchain. To this end, the DeFi chain layer typically requires compatibility with smart contracts. As the oldest and the most widely adopted DLT that supports smart contracts, the Ethereum blockchain is also home to the majority of DeFi protocols [74]. The blockchain implements Ethereum Virtual Machine (EVM) to ensure that state transitions follow the same rules regardless of node they are performed on. The energy consumption and scalability issues associated with blockchains (e.g., EthereumPoW) that are based on the legacy proof of work (PoW) [104] prompted the emergence of the new Ethereum 2.0 and other EVM-compatible chain layer solutions such as Polygon [106], BNB Chain [65], Fantom [78], and Avalanche [60]. Those solutions incorporate alternative consensus mechanisms like proof of stake (PoS) and exhibit an improved throughput capacity [101]. PoS chains in particular not only provides the architectural foundation for the DeFi ecosystem, but can also be a source of yield: to encourage users’ participation in the consensus of the distributed network, many of these PoS chains—including Ethereum 2.0 [77], Solana [116] and Polkadot [105]—reward users’ staking activities.

C. DeFi Primitive Layer

Serving as the fundamental building blocks of the application layer of the DeFi ecosystem, DeFi primitives include AMM-based DEXs, PLFs and stablecoins. DeFi yield mainly comes from AMM-based DEXs and PLFs.

1) AMM-Based DEX: Different from order book-based exchanges where a trade has both buy and sell sides, AMM-based exchanges—often simply referred to as AMM—leverage an algorithm termed “conservation function” to determine the swap rate between two assets given the swap tokens and size [130]. As illustrated in Figure 3(a), traders using an AMM-based DEX swap their tokens against the exchange protocol’s liquidity pool, which contains tokens deposited by liquidity provider (LPs). Against their funds contributed, LPs receive “LP tokens” as a form of “I owe you” (IOU), which allow liquidity withdrawal and entitle LP for their share of swap fee income. At the time of writing this paper, most prominent AMM-based DEX include Uniswap [122], Curve [71] and Balancer [61].

2) PLFs: A PLF (illustrated in Figure 3(b)) typically applies a pre-coded interest rate model that dynamically adjusts the borrow and supply rates [100]. Both rates are commonly programmed to positively correlate with the utilization ratio of the funds, defined as the total amount borrowed as a fraction of the total amount supplied for each specific token asset. PLFs on blockchain are mostly collateral-based rather than credit based. This means that a borrow position can only be created when a sufficient amount of deposit is in
place acting as collateral. The collateral might become liquidated if market movements or interest accrual cause the borrow position to become insufficiently collateralized. From the accounting perspective, interest accrual is achieved through “interest-bearing tokens” which, while sitting in their holder’s wallet, increase in value with the passage of time. Analogous to AMM’s LP tokens, interest-bearing tokens also serve as a form of IOU, which are emitted to lenders according to funds supplied and must be surrendered upon funds withdrawal. Aave [55] and Compound [67] can be counted as the two most popular PLF at the time of writing this paper.

3) Stablecoins: Stablecoins are token contracts deployed on blockchain representing cryptocurrencies that offer price stability relative to a certain reference asset [98], namely, a “peg”. The peg can be another cryptocurrency, legal tender, commodities, or a combination of the above. At the time of writing this paper, the biggest stablecoins, USDT, USDC and DAI are all pegged to the US Dollar. Stablecoins can be custodial or non-custodial, asset-backed or algorithmically programmed.

D. Aggregation Layer

DeFi protocols on the aggregation layer interact with the chain layer or the DeFi primitive layer on end users’ behalf [115]. Depending on whether their target users are requesting or providing services, aggregation layer protocols can be classified as demand-side aggregators and supply-side aggregators. The latter is the category the yield farming protocols belong to.

1) Demand-Side Aggregators: Channeling similar services offered by DeFi primitives, demand-side aggregators seek to present users with the most competitive offer so that they do not have to manually perform the comparison themselves. DEX aggregators Paraswap [37] and 1inch [1] algorithmically search for the optimal swap route through multiple primitive DEXs to generate the best exchange rate for users.

2) Supply-Side Aggregators: All supply-side aggregators to a certain extent perform some form of yield farming. Some protocols farm yields directly from the chain layer. For instance, staking platforms like Lido [93] and Ankr [57] act as a one-stop shop for users to benefit from staking rewards from various PoS chains; yield aggregators like Yearn Finance [134], Beefy [12] and Badger DAO [10] collect users’ funds, redeposit them to DeFi primitives such as DEXs and PLFs or other aggregators to generate returns that will be re-distributed back to the users (presented in Figure 3(c)).

III. YIELD FARMING PRELIMINARIES

In this section, we dive deep into the workings of yield farming protocols, understand how they generate yield for the users as well as revenue for the protocols themselves.

A. Types of Yield Farming Protocols

There is no universal definition for yield farming protocols. Some equate yield farming protocols to generic yield-generating protocols, in which sense, DeFi primitives such as AMMs and PLFs would also be counted as they offer yield to LPs and lenders, respectively. More commonly, however, yield farming protocols refer to protocols on the aggregation layer (see Section II-D) that pool funds to generate return by interacting with DeFi primitives. This is the type of yield farming protocols that we focus on in this paper.

Besides yield aggregators which are the most widely recognized type of yield farming protocols, some other protocols are more implicit in their farming activities by branding themselves as, e.g., stablecoin or lottery protocols. Those protocols mainly differ in the form of IOU tokens they mint to end users upon new deposit.

1) Yield Aggregators: Represented by Yearn, Beefy and Badger, the most classic and commonly known yield farming protocols are yield aggregators. In return for deposit into a yield farming pool, pool tokens that represent a fraction of
the pool wealth are issued. Typically, the value of a pool token varies according to the total pool wealth (see Section III-B2h).

2) Yield-Bearing Stablecoins: A yield-bearing stablecoin protocol works similar to a savings account with a bank. Instead of minting pool tokens, the protocol issues stablecoins to users as a form of certificate of deposit. The yield-generating nature of the protocol is reflected in the increase in the quantity of the stablecoins to their holders, as opposed to the value of the stablecoin token; the value is designed to remain stable to the peg. OUSD issuer Origin Dollar and USDi issuer Bank of Chain [62] are two examples of this type of protocols.

3) Lottery Protocols: A lottery protocol collect users’ funds and issue them each a lottery ticket token in return. The protocol then performs yield farming under the hood. Instead of distributing yield proportionate to users’ deposit, the protocol every once in a while randomly selects one or more winners who can pocket the yield of all participants. PoolTogether [107] is one of the most popular protocols of this type while writing this paper.

4) NFT Farming: Recently, with the popularity of non-fungible token (NFTs), groups began to explore involving NFTs in yield farming. The main goal of NFT farming is to create liquidity and utility for NFTS, especially in gaming space, thereby earning yields for token owners [64]. Axie Infinity [52], ZooKeeper [136], Pulsar Farm [54], and MOBOX [53] are typical platforms that provide NFT farming services.

B. Yield Farming Operations

As illustrated in Figure 3(c), the entire yield farming process comprises actions from both the user and the protocol sides. We discuss common action types associated with yield farming protocols. The exact name and implementation of actions may deviate from one protocol to another.

1) User Actions: The actions that yield farming users, a.k.a “farmers”, need to take are often trivial and straightforward.

   a) Deposit: Protocol users simply select their favored yield farming pool and deposit their funds by transferring token assets to the pool smart contract. In return, users receive pool tokens as a form of IOU which should increase in value with the passage of time due to the yield farmed by the protocol [100], [130].
   b) Redeem: Unless there is a timelock, users can redeem their deposited funds plus any yield generated anytime by surrendering their pool tokens.

2) Protocol Actions: The more sophisticated operations are assumed by the algorithm of the protocol where the actual yield farming is performed automatically under the hood.

   a) Mint: The protocol mints pool tokens to the user proportionate to the amount of funds deposited, representing their share of the liquidity within the yield farming pool.
   b) Burn: When a user requests to withdraw funds from a yield farming protocol, pool tokens need to be surrendered by the user and consequently burned by the protocol.
   c) Invest: Depending on the DeFi primitive that the yield farming pool interacts with, the yield farming protocol can invest funds collected from users either into an AMM as a liquidity provider to collect swap fees (see Section III-B2c), or into a PLF as a lender to earn supply interests (see Section III-B2a). A yield farming pool may also invest in another yield farming protocol, often for the benefit of receiving reward tokens.
   d) Withdraw: When end users request to redeem their funds from a yield farming pool, the pool contract needs to withdraw the corresponding amount of liquidity from the protocol(s) that it has invested in.
   e) Swap: “Raw yield” does not always come in the form of the originally deposited assets. Therefore, the yield farming protocol may perform a swap, usually on an AMM, to convert yield tokens into the same tokens as originally deposited, which are sometimes reinvested to achieve the compounding effect.
   f) Borrow: Yield farming protocols may use all or part of the funds deposited by users as collateral to borrow from a PLF. This may need to be performed due to various reasons: (i) to arbitrarily inflate the borrow position to be qualified for more participation reward (see Section III-B2b), (ii) to borrow out assets that can be invested to generate higher yield than the deposited assets.
   g) Repay: Yield farming protocols that take the “borrow” action may need to partially or fully repay their loans to reduce or close its borrow position if: (i) the borrow position is on the verge of becoming liquidated, (ii) the collateral must be withdrawn so that it can be invested elsewhere or returned to end users.
   h) Rebase: A yield farming pool mints or burns pool tokens depends on the quantity of the asset deposited or withdrawn as well as the exchange rate between the pool token and the asset. As yield farming progresses, the farming pool usually accumulates wealth and the exchange rate changes. Due to diversified investment in various protocols, some yield farming pools may possess an array of assets different from the one deposited by end users. Yield farming protocols connect to price oracles to fetch the price of each of these assets, and subsequently calculate the total value held by the pool. The exchange rate can thus be updated through dividing the latest pool value denominated by the asset deposited by end users with the circulating quantity of the pool tokens. This process of updating the pool token price is termed “rebase”.

C. Forms of Yield Farming Pools

Different yield farming protocols vary in terms of their pool structure and token types acceptable by each pool (Table II).

1) Pool Structure: A yield farming pool may accept deposits in single or multiple assets.

   a) Single asset: Most yield farming protocols have single-asset pools. While those pools only accept one particular token asset, they may still hold various assets due to different sorts of yield farmed. Typically, those other assets are automatically swapped for the one acceptable as deposits, and reinvested to generate compounded yield (see Section III-B2).

   b) Multiple assets: A yield farming pool may also accept multiple token assets. Usually assets acceptable by the same pool share a peg. For example, at the time of writing this paper, Badger DAO’s ibBTC/crvsBTC pool
accept ibBTC, renBTC, WBTC and ibbtc/sbtcCRV-f, all pegged to BTC.

2) Accepted Token Types: Yield farming protocols accept various types of tokens, ranging from stablecoins to LP tokens.

a) Stablecoins: In the recent low-interest environment in the traditional finance (TradFi) space, yield farming solutions that boast to offer a high single-digit to a double-digit annual percentage yield (APY) for USD-pegged stablecoins have been of particular interest. Most yield farming protocols offer stablecoin farming; in fact, among the top 20 yield aggregators, only Badger DAO has no stablecoin pool thus far.

b) LP tokens: Many yield farming pools also accept LP tokens. As discussed in Section II-C1, LP tokens themselves already entitle their tokenholders to swap fee income. Nevertheless, having LP tokens managed by a yield farming pool provides the additional benefit of automatically converting and reinvesting participation reward (see Section III-D3) distributed by the respective AMM.

c) Others: Other asset types may also be eligible for yield farming. For example, Yearn Finance accepts ETH, the native currency on the Ethereum blockchain, as well as UNI and YFI, which are protocol governance tokens of Uniswap and Yearn Finance itself, respectively.

D. Sources of Yield

1) Supply Interest: The most straightforward type of yield originates from lending. As the demand for loans in crypto assets grows, the borrowing interest rate increases, leading to higher yields for lenders. Particularly in a bullish market, speculators are keen to borrow funds despite a high interest rate, in expectation of an appreciation in the assets of their leveraged long position. A borrower wishing to increase their exposure to ETH, for example, may use ETH as collateral to borrow USDC, then repetitively exchanging USDC for ETH to deposit it as collateral to borrow more USDC, forming a “leveraging spiral” [131]. Compound [67] and Aave [55], two major DeFi lending protocols while writing this paper (see Section II-C2), have witnessed the borrow APY of USDC rising from 2.3% in May 2020 to as high as 10% in April 2021.\(^1\) This specific kind of yield is incorporated in interest-bearing tokens, such as cTokens from Compound or aTokens from Aave.

2) Swap Fee Income: Some tokens entitle users to part of the revenue that is going through the protocol. These can be governance tokens or other kinds of tokens. One example is the liquidity provider tokens in AMM-based DEXs [130]. By supplying liquidity into an AMM pool, users receive the fees that are paid by traders within that pool. The higher the volume in that pool, the more fees that are generated, and the more a liquidity provider profits from this. In Uniswap [122], a 0.3% fee is charged for every trade within a pool and goes fully to LPs.

3) Participation Reward: Another yield source comes from liquidity mining programs, where early participants receive native tokens representing protocol ownership. This incentivizes people to contribute funds into the protocol, and enhances decentralization as the protocol ownership is distributed to users. The native tokens often have a governance functionality attached to them which is deemed valuable, as the token holders have a say in the future strategic direction of the project. Native tokens sometimes also entitle holders to a share of the protocol revenue. Further, the values these tokens possess itself especially in a speculation context can be the benefits of owning a protocol.

This brings up a second kind of revenue-sharing token, where users have to actively stake their tokens to receive a share of the revenue. For example, SUSHI holders that stake their SUSHI will get xSUSHI in return, which represents the proportional share of a pool that captures 0.05% of all trades on Sushiswap [117]. Vesper Finance’s governance token, VSP, can also be deposited in a pool, in return for vVSP, a token that represents the user’s proportional share of a pool that captures part of the revenue generated throughout the whole Vesper platform [124].

\(^{1}\)https://app.defiscore.io/assets/usdc
TABLE III

OVERVIEW OF ATTACKS IN AGGREGATORS AND POTENTIAL SOLUTIONS

| Attacks          | Yield aggregator attacked | Summary                                                                 | Solutions                                                                 | Estimated lost | References | Time       | Major chain |
|------------------|----------------------------|-------------------------------------------------------------------------|---------------------------------------------------------------------------|----------------|------------|------------|-------------|
| Flash loan attack| ApeRockey                  | Using the fact that the AntiCake vault was only deployed 10 hours and was low in TVL, attacker conducted price manipulation and drained the vault. | Project team updated the protocol and at least two audits will be conducted before its V2 launch. | 1.26 m USD    | [18], [19], [20] | 09/03/2021 | SNB         |
|                  | Poolsity                   | Within the interface to create a new pool, attacker transferred USDT into the contract and called removal of liquidity. Caused for value of USDT token to crash by more than 95%. | A realization of the floating rate of interests and the remuneration code changes. | 3 m USD       | [35], [36], [37], [38], [39] | 09/03/2021 | SNB         |
| Harvest Finance  |                            | Attacker swapped USDC to USDT to swap the price of USDT, depleting USDT from vault and swap back USDT to USDC to gain profit as USDT price fell. This action is repeated to drain the vault. | Team updated the following: deposit and withdraw funds within a single transaction is not allowed to avoid flash loans, and withdraws of tokens are made into multiple transactions to minimize damage. | 3.5 m USD     | [28], [29], [30] | 28/10/2020 | Ethereum    |

Rug pull

A. Security Risks

We identify four major types of attacks associated with yield farming. Table III presents anecdotal events of these attacks and their potential solutions.

1) Flash Loan Attack: Flash loan attacks [110] abuse the mechanism of flash loan protocols in which an attacker borrows a great deal of funds that do not require collateral. The attacker then manipulates the price of an asset in a very short period and quickly resell it to earn profits. Such a procedure can be repeated for multiple time by the attacker, thereby causing considerable losses to investors and yield aggregators.

Major yield aggregators have witnessed multiple waves of flash loan attacks, losing millions of dollars each. For example, ApeRocket suffered two flash loan attacks that costed investors 1.26 million USD on July 2021 [18]; in October 2020, a farmer leveraged flash loans to reap 33.8 million USD from the USDT and USDC pools [28]; Pancake Bunny Finance lost around 690,000 BUNNY tokens due to removal of liquidity and price manipulation by flash loan attacks [35].

To defend against flash loan attacks, developers must consolidate and improve flash loan protocols, making them difficult to be exploited by attackers. An effective approach is to setup floating interest rates, thereby increasing the cost of launching flash loan attacks [35]. Developers can also choose to enhance the audits [7] or forbid depositing and withdrawing funds within a single transaction [51].

2) Rug Pull: A rug pull refers to the abandonment of a project by the project administrator after collecting investor’s funds, leaving investors with valueless assets [118], [130]. One way of conducting this type of scam is to lure yield farming protocols into buying assets with no value and then swap this asset for ETH or another type of asset with value. For example, Arbix Finance, a typical rug pull, drained around 10 m USD in users’ assets directly from the vaults without any advanced attack techniques in sight [8].

To prevent from being rugged, investors should exercise caution and always confirm a project’s credibility before investing in it [95], [129]. Besides, continuously tracking the audit information of invested projects enables investors to quickly identify risks and take appropriate measures, thereby reducing losses [43].

3) Reentrancy Attack: Even though the composability factor of DeFi is what makes yield farming possible in the first place by allowing for complex, interconnected financial
protocols, it does bring along the danger of smart contract risk as more and more money legos are plugged into a strategy. While two smart contracts may be secure in isolation, the combination of them may not. By composing multiple smart contracts together, the attack surface might be greater than the sum of its parts [126], [127].

Reentrancy attack is one of the most destructive attacks that appear when multiple smart contracts operate with each other. More specifically, the reentrancy attack occurs when a smart contract makes an external call to another smart contract. Then the another contract makes recursive calls back to the original function, intentionally or unintentionally withdraw funds. When the original contract fails to update its state before sending funds, the attacker can exploit this vulnerability to continuously drain the contract’s funds.

Major yield aggregators have witnessed a large amount of reentrancy attacks in the past several years. In April 2021, the ForceDAO DeFi aggregator was exploited by a group of attackers, who utilized reentrancy attacks to steal 367 thousands USD worth of tokens before the ForceDAO team took effective actions to prevent further attacks [19]; in September 2021, DAO Maker, a decentralized finance platform on Ethereum, was hacked for almost 4 million USD due to insecure smart contracts [16]; a reentrancy attack on the Grim Finance project within the Fantom Blockchain also successfully drained over 30 million USD worth of tokens in 2021 [5].

To defend protocols against reentrancy attacks, researchers and developers have proposed a variety of frameworks and methods [86]. For example, Rodler et al. [112] propose a backward compatible approach based on run-time monitoring and validation to protect smart contracts on Ethereum; Das et al. [73] propose a reentrancy-aware language called Nomos, which enforces reentrancy security using resource-aware session types; Cecchetti et al. [66] first formalize the reentrancy interface on general distributed systems and then leverage information flow control to automatically fix defective smart contracts. However, with the increasing complexity and variety of DeFi protocols, reentrancy attacks will also become increasingly difficult to detect and counter. From users’ side, protection can be sought from DeFi-native insurance protocols such as Nexus Mutual that cover smart contract risks [68].

4) Key Exploit: Due to poor access control of some DeFi systems, yield aggregators can be attacked by exploiting various keys (e.g., API key, wallet key) to tamper with the smart contracts or drain funds. For example, the Bent Finance utilized non-multisig wallets to deploy their project’s smart contracts. Anyone who knows the appropriate private key can perform updates to the contracts, allowing attackers to inject malicious code and create the backdoor. In December 2021, an attacker leveraged this feature to drain 1.75 million USD worth of tokens from the pool [13], [14], [20].

To avoid attacks based on key exploiting, DeFi contracts should always be deployed upon multisig wallets to eliminate single points of failure [20]. DeFi platforms should also properly protect the private keys used to access and control correlative smart contracts. The developments of DeFi system API, application, and user interface should follow software security practices, ensuring that the access control and function calls are solidly implemented.

5) Other Attacks: As yield farming protocols are built upon multiple complex systems with a variety of software and hardware components interacting with each other, both technical and economic weaknesses give rise to attractive exploit opportunities for malicious hackers. Besides the aforementioned attacks, there are many other attacks targeting the blockchain infrastructure, user interface, or even network communications, thereby disturbing the proper operation of yield farming protocols. For example, malicious miners can prioritize transactions in their favor by inspecting maximal value (MEVs), thereby causing damages to the smart contracts running on the upper layers [109], [135]; attackers can break the network connections between the users and the blockchain system through border gateway protocol (BGP) hijacking [58]; malicious traders can leverage front-running attacks to drain funds from pools [76].

These attacks are out of scope for this paper, but it is important for users and developers to be aware of that yield farming security is a systemic problem. Only by ensuring the security of every component in the system can the security of yield seekers’ funds be ensured.

B. Economic Risks

Besides security concerns, there exist various economic risks associated with yield farming. In Section ??, we demonstrate that investment strategies with the potential to generate remarkably high yield also bear high risks. While our simulation only illustrate return courses in a deterministic fashion, through various simulated scenarios one can easily extrapolate that the ever-changing market conditions—including volatile price movements and trading activities—lead to return instability, and sometimes even losses. Below, we discuss several types of economic risks associated with yield farming.

1) Yield Dilution Risk: Yield farming pools providing double or even triple digit APY can be deceiving in their return generating capability. Often enough, those pools are thin in liquidity and lack scalability, unable to accommodate a large amount of deposit while sustaining a similar level of APY. Users who invest a significant amount of funds into such a pool may find the pool APY dropping significantly immediately afterwards. For users with funds already in the pool, they may experience a decrease in return on their investment due to dilution from newly added funds. For those who do not constantly monitor their investment performance, this may mean leaving their funds in a diluted, low-APY pool, while missing the opportunity of reallocating their funds to more profitable strategies.

2) Conversion Risk: As discussed in Section III-C, yield farming pools typically specify tokens that they can accept. Therefore, to participate in yield farming, users may have to first convert partially or all of their funds into acceptable assets for yield farming. This engenders conversion risk: a user might have been better off holding their original funds, than converting them to “eligible” assets. This is because the “eligible” assets might depreciate against the original assets
prior to the conversion to such an extent that even the yield
generated cannot make up for the depreciation loss. This risk
is most prominent with liquidity provision strategies, mani-

tested by the so-called “impermanent loss” (see [130]). By
design, the value LP tokens (e.g., USDT–ETH–LP token) from
an AMM-based DEX falls against the original portfolio (e.g., a
combination of USDT and ETH). Sometimes, even the swap fee
income and the participation reward are insufficient to cover
the conversion loss.

3) Exchange Risk: Related to conversion risk, exchange
risk is associated with the uncertainty surrounding the ex-
change rate between the assets held by the yield farm-
ing pool and the denominating currency (usually USD). As
demonstrated in Section III-A4, yield farming strategies ben-

efit from—and, in cases such as leveraged borrow, solely rely
on—the high value of participation reward. This makes yield
farming highly speculative as token prices are unpredictable.
An overly low price of yielded tokens such as reward tokens
might result in a loss for end users.

4) Counterparty Risk: This risk is associated with farming
strategies that incorporate lending, where loans might not be
repaid. While the simple lending strategy (see Section III-B2a)
is a relatively low-risk one, losses may still occur under
extreme market conditions, e.g., when the price of the asset
lent out relative to the collateral suddenly increases to such
a significant extent that the loan becomes undercollater-
lized (see lending protocol MakerDAO’s Black Thursday
Incident [91], [100]). In such cases, borrowers may choose
not to repay their loans since their collateral is not worth the
effort anymore, resulting in a default. Kao et al. [90] simu-
late a wide range of market volatility to stress-test lending
protocols such as Compound, and find that only rarely can
undercollateralization occur.

5) Liquidation Risk: Liquidation risk is associated with
farming strategies such as leveraged borrow (see Section III-B2b)
that incorporate taking a overcollateralized
loan on a PLF. Due to price movements and interest accrual,
a loan position may become insufficiently collateralized,
triggering liquidation of the deposited assets backing the
loan. At liquidation, the value of the collateral liquidated by
design exceeds the loan payable reduced, resulting a loss
on the side of the borrower. Thus, yield farming protocols
that implement borrow but unable to handle liquidation risk
properly may cause users to lose their funds.

V. RELATED WORK

In this section, we introduce literature that is related to yield
farming in some shape and form. As it is still a fairly new
area, there is a paucity of existing works related to our paper.
Table IV summarizes the most related and representative ones.

In general, our paper is different from existing papers in the
following aspects:

• our paper focuses on the subject of yield farming,
while other papers either investigate a more specific
DeFi topic (e.g., yield generating mechanism [125],
yield chasing [114]), examine a different but related
DeFi applications (e.g., AMM-based DEX [130], lend-
ing [63]), or have a broader coverage (e.g., DeFi transformation [108]);

• to investigate yield farming comprehensively, our paper
utilizes multiple methodologies, including literature sur-
vey, modeling, empirical analysis, and taxonomization.
In contrast, other papers use only one or a few of these
methodologies to study their subjects;

• our paper examines a series of aspects of yield farming,
including related DeFi protocols, yield generation, yield
farming strategies, financial risks, and security issues,
while most of the related papers only cover some of these
topics.

We discuss related literature in more details from different
perspectives in the following subsections.

A. Yield Farming

A few papers focus on studying and comparing yield
farming strategies or yield aggregators [59]. For instance,
Walton [125] provides a break down of yield generating
mechanism, covering four different farming strategies and
a few other related topics (e.g., benefits and risks); Kanis
Saengchote [114] studies DeFi composability, which covers
yield-chasing behaviors and some introductions about major
yield aggregators; another case study about Compound [113]
also comes with explanations about yield aggregators and yield
farming incentives; Popescu [108] discuss the transition from
the traditional finance to DeFi and include some descriptions
about yield farming.

However, none of these works have comprehensively inves-
tigated yield farming from the perspective of literature survey,
modeling, empirical analysis, and taxonomization like our
paper.

B. DeFi Platforms

As a type of DeFi application, yield farming is built upon
DeFi platforms. Combing the design of general DeFi platforms
lays the groundwork for yield farming designs. Moin et al. [98]
and Pernice et al. [102] systematically study the general
designs of DeFi platforms by decomposing the structure into
diverse elements (i.e., peg assets, collateral amount, price
and governance mechanism). They also investigate the mer-

its and demerits of DeFi platforms to spot future directions.
Nonetheless, yield farming is not the main focus of these
works.

C. Related DeFi Protocols

There are various papers studying the DeFi protocols (e.g.,
flash loan, lending, trading) that can be leveraged by yield
farming. For example, as fundamental protocols of yield farm-
ing, the mechanisms, properties, and risks of DeFi lending pro-

tocols are extensively investigated in several publications [63],
[90], [100], [119]; some papers [81], [100] provide analysis
and discussions about protocols for Loanable Funds, intro-
ducing the interest rate determination and liquidity issues;
Han et al. [82] zoom into the launch event of the yield farming
protocols for Uniswap liquidity provision and further establish
the causal impact of this on Binance investor trading activities; within the scope of the analysis of financial attack vectors that involve a flash loan, Qin et al. [110] study the existing flash loan-based attacks and propose optimizations that significantly improve the ROI of these attacks; Gudgeon et al. [80] explore how design weaknesses in DeFi protocols can trigger a decentralized financial crisis.

Although these papers can cover almost all the DeFi protocols used by yield farming, our paper presents this topic more systematically by putting together all the relevant protocols, components, and problems worth exploring.

VI. CONCLUSION

In this survey paper, we examine yield farming protocols from multiple perspectives. We first highlight yield farming’s dependence on lower-level DeFi primitives in the context of the broader DeFi ecosystem and propose a general framework for yield farming protocols. We then explain code-level actions and associated yield farming operations. We decompose various aspects of yield farming protocols such as protocol form, pool structure, accepted token types, and enumerate their variations. Later, we stylize three frequently used strategies and simulate yield farming performance under a set of assumptions. We also compare four major yield aggregators by summarizing their strategies and revenue models. Finally, we discuss security and economic risks of yield farming protocols, together with related work.

While yield farming has been exploding since 2020, an important question remains if current yields will be sustainable in the long term. Higher rewards also imply higher risks, and associated DeFi attacks prove that the safest and most robust yield provider will win the race. Besides security enhancement, new industry developments should consider building one-stop-shop solutions, in pursuit of aggregating more than just yield and facilitating the on-boarding of new DeFi users.

ACKNOWLEDGMENT

The authors would like to extend their sincere thanks to Simon Cousaert and Toshiko Matsui for the base of this paper [69]. The authors also would like to thank Yitian Wang and Honglin Fu for their research assistance.

This material is based upon work partially supported by Ripple under the University Blockchain Research Initiative (UBRI) [79]. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Ripple.
