

What Doesn't Kill You Makes You Stronger? Evidence from Vampire Attack on Decentralized Exchange

Xi Zhao
Xi'an Jiaotong University
zhaoxi1@mail.xjtu.edu.cn

Jian Li
Xi'an Jiaotong University
lijian_som_phd@stu.xjtu.edu.cn

Xiang (Shawn) Wan
Santa Clara University
xwan@scu.edu

Xinyu Zang
University of Florida
xinyu.zang@warrington.ufl.edu

Hsing Cheng
University of Florida
kenny.cheng@warrington.ufl.edu

Abstract

Our study examines the impact of vampire attack, a unique platform entry strategy in the blockchain ecosystem. The cryptocurrency market has grown dramatically over the past decades since the emergence of Bitcoin protocol in 2008. Decentralized exchanges (DEXs), allowing blockchain users to directly buy and sell cryptos on the chain, started to blossom recently with trading volume reaching 80% on-chain market share at its peak. As the DEX becomes immensely popular, so does the vampire attack, where the entrant attacker completely clones the incumbent's platform and "sucks" its liquidity by providing tokenized rewards. We implement a quasi-experimental design to uncover the deposit-side and exchange-side impact of the first and most famous vampire attack launched by Sushiswap (attacker) against Uniswap (incumbent). Surprisingly, the vampire attack leads to no significant downturn on the deposit-side, and even promotes the exchange-side performance. We plan to further uncover the underlying reasons that contribute to these intriguing results.

Keywords: Vampire Attack, Decentralized Exchange (DEX), Tokenized Incentive, Platform Entry Strategy, Matrix Completion

1. Introduction

Since the emergence of the Bitcoin protocol in 2008, the cryptocurrency (crypto) market has grown dramatically over recent decades. By October 2022, over 9,000 cryptocurrencies existed in the blockchain ecosystem, which promoted the emergence of platforms for fiat-to-crypto and crypto-to-crypto exchanges (Amiram et al., 2022). Traditionally, blockchain users trade cryptos on centralized exchanges (CEXs) such as

Binance, OKex, and Huobi. Playing the role of an intermediary that matches order makers and takers to execute crypto trades (Reif, 2021), CEXs face the challenges of asset security, transaction limitations, privacy concerns, and redundant costs because traders have to transfer the custody of their crypto assets to the CEXs for subsequent trades (Benedetti & Nikbakht, 2021; Schär, 2021). Decentralized exchanges (DEXs) have begun to flourish to overcome these challenges (Capponi & Jia, 2021; Han et al., 2021; Xia et al., 2021). DEX is an application for blockchain users to directly buy and sell cryptos on the chain without involving the custody of traders' crypto assets. Over the past few years, the trading volume of DEXs has grown dramatically, reaching an 80% on-chain market share at its peak in 2021. Popular DEX platforms include Uniswap, Balancer, Curve.fi, Bancor, and so on.

In contrast to the CEX platform, where users interact with order books, asset exchanges on the DEX platform (specifically, the automated market maker type) rely on automatically executed smart contracts. Users can deposit or swap crypto assets in the liquidity pool of the DEX platform. Traders must pay trading fees to swap "tokens" (a generalized term to describe crypto assets) in a liquidity pool. These fees are paid as interest to liquidity providers who deposit token assets in this liquidity pool. Multiple liquidity pools exist on a DEX platform, and each liquidity pool consists of a pair of tokens, such as ETH/WBTC and ETH/DAI. Liquidity providers must reserve a pair of tokens with equal value rather than a single token type. Liquidity providers receive a pool-specific deposit certificate in token form (called "liquidity pool tokens"). Liquidity pool (LP) tokens represent providers' liquidity shares in a liquidity pool and are used to withdraw their corresponding token assets and interest reserved in the pool.

As the DEX becomes immensely popular, so does the **vampire attack** on the DEX platform, where the

entrant completely clones the incumbent's platform and "sucks" liquidity from the incumbent to its cloned platform by providing tokenized rewards to the incumbent's depositors. One of the most famous vampire attacks in Ethereum is the attack launched by Sushiswap against the largest DEX on the Ethereum blockchain – Uniswap, in 2020. Other examples of vampire attacks include the Swerve protocol's vampire attack on the Curve.fi in 2021 (Onyshchenko, 2020), and the KLEX Protocol's vampire attack against the KLAYswap in 2022 (Jackson, 2022).

Although prior studies have examined various competition strategies of the entrant platform, there is a lack of literature providing insights into the emerging "vampire attack" strategy owing to its unique features: First, due to the "open-source" feature of blockchain, the attacker can completely "clone" the incumbent platform by copying the incumbent's smart contracts and launching a new platform with identical functionalities and quality. Second, since any decentralized project on programmable blockchains can release its own token as access to on-chain services, investments and even voting rights, the attacker can launch tokenized monetary incentives to attract potential users even without large-scale external financing. Third, thanks to the tokenized deposit certificates (elaborated in Section 2.1) issued by the incumbent platform, the attacker can launch a targeted attack on the specific industry leader (i.e., the incumbent platform) by collecting corresponding tokenized deposit certificates to drain its livelihood (i.e., its cryptos deposits), hence the term "vampire" attack. Accordingly, we ask the following research question: *What is the impact of the vampire attack on the operational performance of an incumbent platform?*

To answer the above research questions, we implement a quasi-experimental design by leveraging the first and the most famous vampire attack that occurred on Ethereum: the vampire attack by Sushiswap (hereafter interchangeably termed the attacker) on Uniswap (the largest DEX on Ethereum, hereafter interchangeably termed the incumbent). During the attack period, some liquidity pools on Uniswap were selected as attacking targets by Sushiswap, whereas others were not. A comparison between these attacked pools and unattacked pools helps us identify the effect of the vampire attack at the liquidity pool level. The attacker may not randomly choose the incumbent's liquidity pools to attack, and the number of attacked liquidity pools is also limited. Thus, we employ the matrix completion (MC) estimator as our main identification strategy (Athey et al., 2021), which addresses the concern of unobservable time-varying confounders in estimation and specializes in handling the condition with a substantial number of pre-treatment

period data for one or a limited number of treated units (Liu et al., 2022; Pan & Qiu, 2022). We also use the estimation from the fixed-effects difference-in-differences (DID) method as a robustness check.

We examine the impacts of the vampire attack on both the deposit side and exchange side of the incumbent platform at the pool-day level. The deposit-side performance of a liquidity pool is captured by its liquidity (\$) balance till the end of each day, and the exchange-side performance of a liquidity pool is captured by its trading volume (\$) during each day. To acquire convincing results, we conduct such analyses on both the unmatched full sample and the matched sample obtained from propensity score matching. One would expect the vampire attack to hurt the liquidity of the attacked pool. Surprisingly, we find an *insignificant* effect of the vampire attack on the daily liquidity of the attacked liquidity pool. Even more surprisingly, we find that the vampire attack *expands* the overall trading market of the attacked liquidity pool.

Our study contributes to the DEX literature by uncovering several intriguing results on the impact of the vampire attack on the operational performance of the incumbent DEX platform. Our study also contributes to the literature stream on the competitive strategy of the entrant platform by empirically examining a new tokenized reward-based competitive strategy. Our study offers important managerial implications to blockchain users, DEX platforms, and policymakers.

The remainder of this paper is organized as follows. The following section describes the empirical context, data collection, variable and sample construction, and identification strategy used in our study. Section 3 presents results with discussions. Section 4 reports the robustness checks. Section 5 concludes this study and offers future research directions.

2. Research setting

In this section, we present the research context, data collection, variable construction, sample construction, and identification strategies.

2.1. Research context

We implement a quasi-experimental design by leveraging the first and the most famous vampire attack that occurred on Ethereum: the vampire attack by Sushiswap (hereinafter called "the attacker") on Uniswap (the largest DEX on the Ethereum blockchain). In contrast to centralized exchanges that serve as trusted intermediaries to match buyers' and sellers' trading orders, DEXs such as Uniswap allow automated cryptocurrency depositing, withdrawing, and swapping

of a pair of tokens at any time on the chain through a liquidity pool that consists of the pair of tokens.

On Uniswap, users can deposit or swap crypto assets in the liquidity pool. Figure 1 shows how a liquidity pool (Token A/Token B) works. Users must pay trading fees to swap tokens in a liquidity pool (called “traders”). These fees are paid to users who deposit token assets in this liquidity pool (called “liquidity providers”) as interest. Multiple liquidity pools exist on Uniswap, and each liquidity pool consists of a pair of tokens, such as WBTC/ETH and DAI/LUSD. Liquidity providers receive a pool-specific deposit certificate in token form (called “liquidity pool tokens”). Liquidity pool (LP) tokens are transferable on the blockchain, represent providers’ liquidity shares in a liquidity pool, and are used to withdraw their corresponding token assets and interests reserved in the pool. Note that, the crypto assets in DEXs on the Ethereum blockchain can be divided into two categories: (i) Ether, which is the native cryptocurrency to the Ethereum blockchain; (ii) Tokens, which are digital units issued by third-party projects on Ethereum. Following the industry’s practice (e.g., <https://docs.uniswap.org/contracts/v2/concepts/core-concepts/pools>), we do not distinguish these two types of crypto assets in our study and use the generalized term “token” to refer to both of them.

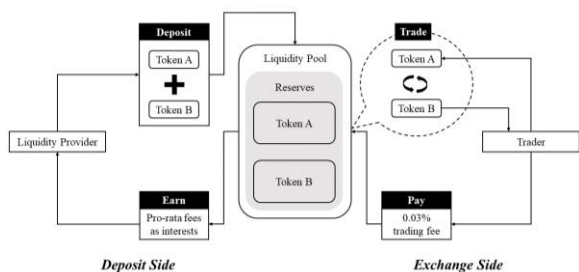


Figure 1. An operation diagram of a liquidity pool (Token A/Token B) on Uniswap

The attacker, Sushiswap, selected a number of liquidity pools on the Uniswap platform to launch the vampire attack. The vampire attack on liquidity pools (i.e., treatment) occurred on the same day and lasted for two weeks, starting on August 26, 2020, and ending on September 9, 2020. During this period, the liquidity providers of the attacked liquidity pools on Uniswap could receive tokens issued by the attacker as a reward if they provided their Uniswap liquidity pool tokens to the attacker. The attacker established its DEX at the end of the attack period and migrated the “drained” liquidity from Uniswap to its platform. The converted liquidity was subsequently used for token trading on the attacker’s own platform. In addition to providing tokenized rewards to liquidity providers on the deposit

side, the attacker also copied the Uniswap smart contract and cloned the whole platform, including both the deposit and exchange sides.

2.2. Data collection and variable construction

Our unit of analysis is at the pool-day level. The transparency of the public Ethereum blockchain allows access to both the incumbent’s operational performance and its user behavior recorded on the chain (Amiram et al., 2022; Shen et al., 2021). We compare the performance prior to and post the vampire attack to uncover the influence of the event. Specifically, we select *around one week* (i.e., September 10-15, 2020) after the end of the attack as the post-attack period to examine the desired effects because Uniswap released its own token after September 15, 2020. We select *two weeks* (i.e., August 12-25, 2020) before the vampire attack as the pre-attack period since our identification strategy (see details in Section 2.4) requires no less than ten pre-treatment observations for each unit. Figure 2 displays our research design.

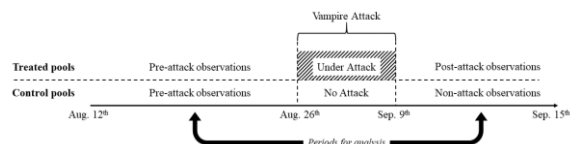


Figure 2. Event timeline and research design

We collect two sets of raw data for the incumbent’s liquidity pools: (i) liquidity pool-related characteristics, including the characteristics of the token in each liquidity pool, and (ii) interaction between Ethereum users and the liquidity pool, covering all Ethereum users who used to deposit (i.e., provide liquidity), withdraw, and swap tokens (i.e., trade) in the liquidity pools of the incumbent. Ethereum users are labeled as liquidity providers if they provide liquidity to the incumbent’s liquidity pools, and as traders if they trade/swap tokens through those pools. Then we aggregate these raw data into pool-day level variables as follows.

As for the treatment indicator variable, we create a dummy variable, *Treat*, which takes the value of one if a liquidity pool was attacked and in the post-vampire attack period. On the DEX platform, liquidity and trading volume are the most widely used operational performance metrics for the liquidity pool on the deposit and exchange sides, respectively (Han et al., 2021; Xia et al., 2021). Hence, we are interested in the impact of the vampire attack on both the deposit-side performance (i.e., liquidity) and exchange-side performance (i.e., trading volume) of liquidity pools. As for outcome variables, (i) we use the daily *liquidity* (\$) of a liquidity pool to proxy its deposit-side performance, which serves

as the deposits balance in a liquidity pool till the end of a day; (ii) we use the daily *trading volume* (\$) of a liquidity pool as a proxy for the exchange-side performance, which is the sum of the trading volume across all traders in a liquidity pool on a given day.

As for covariates, we include the observed characteristics of the liquidity pools in our analysis. Each liquidity pool comprises a pair of tokens. Note that both the attacked and unattacked pools in the final samples (elaborated in Section 2.3) for analysis include the Ethereum blockchain’s native cryptocurrency – Ether (ETH). Thus, we only collect information on the other token, not Ether, in each liquidity pool. We collect the following observed characteristics for each liquidity pool: (i) age of a liquidity pool, which counts the number of days since the liquidity pool was created; (ii) listed days of a token, which counts the number of days since a token in a liquidity pool was listed; (iii) daily open price (close price) of a token, which is the price at the beginning (end) of a day; (iv) daily top price (bottom price) of a token, which is the highest (lowest) price of the token in a day; (v) centralized exchange trading volume of a token, which is the trading volume of a token in centralized exchanges in a day; (vi) market cap of a token, which is the total market value of a token’s circulating supply in a day. Table 1 presents summary statistics of the outcome variables and the covariates.

Table 1. Summary statistics

Variables	Mean	S.D.
Outcome variables		
<i>Liquidity</i>	331,526	1,521,152
<i>Volume</i>	458,724	3,918,119
Covariates		
<i>Pool Age</i>	60.93	32.90
<i>Listed Days</i>	523.00	431.6
<i>Open Price</i>	270.80	3,479.94
<i>Close Price</i>	272.80	3,478.41
<i>Top Price</i>	316.70	4,225.71
<i>Bottom Price</i>	230.80	2,835.12
<i>CEX Trading Volume</i>	1.21×10^8	2.10×10^9
<i>Market Cap</i>	9.32×10^7	6.80×10^8

2.3. Sample construction

To ensure the stability and cleanliness of the estimation, we first select appropriate liquidity pools by setting preliminary standards like “no less than ten pre-treatment observations” (required by our identification approach in Section 2.4). A liquidity pool consists of a pair of tokens. To ensure the comparability between the treated and control pools, we should ensure the compatibility of tokens in each pool. However, not all tokens are created equally. Especially, as the native cryptocurrency used in the Ethereum platform, liquidity pools with Ether may have relatively higher popularity.

In addition, all eligible treated pools contain Ether, and around 60 percent of liquidity pools on Uniswap include Ethereum. Therefore, to ensure the comparability between the treated and control pools, we drop the unattacked pools (control pools) without Ether. The resulting sample consists of 391 pools, of which 11 are attacked and 380 are unattacked.

The attacker, Sushiswap, might not randomly choose the liquidity pools on Uniswap to launch a vampire attack. For example, an attacker may tend to attack pools with higher liquidity and trading volumes to attract more providers and traders from Uniswap. In our dataset, we find that, on average, the attacked pools have higher liquidity and trading volumes than the unattacked pools. To alleviate such a concern, we leverage the propensity score matching (PSM) method to construct a matched sample. Specifically, we include the following liquidity pool-related covariates in the pre-attack period for matching: *pool age* and the non-Ether token’s *listed days*, *open price*, *close price*, *top price*, *bottom price*, *centralized exchange trading volume*, and *market cap*. We adopt the nearest-neighbor function to match each pool in the treatment group with ten pools in the control group with replacement. The matching process results in a sample of 7 attacked pools and 47 unattacked pools. We further check the balance of the matched sample and find qualitatively similar characteristics between the treated and control pools, indicating comparability (see Table 2). We employ both unmatched and matched samples in our analysis.

Table 2. Balance checks

	Before Matching (N _a =11, N _{na} =380)		
	Mean (attacked)	Mean (unattacked)	Mean Diff.
<i>Pool age</i>	92.46	58.66	33.80***
<i>Listed days</i>	607.00	519.19	87.81
<i>Open price</i>	649.34	166.10	483.24
<i>Close price</i>	682.08	170.47	511.61
<i>Top price</i>	738.95	208.47	530.48
<i>Bottom price</i>	585.66	140.18	445.48
<i>CEX Trading Volume</i>	3.46e+09	5.50e+06	3.45e+09***
<i>Market Cap</i>	1.63e+09	3.87e+07	1.59e+09***
	After Matching (N _a =7, N _{na} =47)		
	Mean (attacked)	Mean (unattacked)	Mean Diff.
<i>Pool age</i>	95.71	95.19	0.52
<i>Listed days</i>	410.57	326.77	83.80
<i>Open price</i>	26.99	14.38	12.61
<i>Close price</i>	27.20	14.57	12.63
<i>Top price</i>	28.93	15.52	13.41
<i>Bottom price</i>	25.66	13.53	12.13
<i>CEX Trading Volume</i>	4.55e+07	2.26e+07	2.29e+07
<i>Market Cap</i>	2.67e+08	1.38e+08	1.29e+08

Notes. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

2.4. Identification strategy

Drawing on the computer science and statistics literature on matrix completion and the econometric literature on factor models and interactive fixed effects (Xu, 2017), Athey et al. (2021) develop the *Matrix Completion* (MC) method that treats the causal inference problem as a task of completing a matrix with missing elements of treated counterfactuals and show that the MC estimator outperforms the unconfoundedness approach, synthetic control approach, and fixed-effects difference-in-differences (DID) in several cases (Bronnenberg et al., 2020).

Considering the advantage of the MC estimator in handling the condition with a substantial amount of pre-treatment period data for one or a limited number of treated units, we adopt the MC estimator as our primary identification strategy. The MC estimator can alleviate the concern of unobserved time-varying confounders (Liu et al., 2022; Pan & Qiu, 2022). This method regards the causal inference problem as a task of completing a matrix with missing elements of treated counterfactuals and does not rely on the assumption of no unobservable time-varying confounders like conventional estimation methods (Liu et al., 2022; Pan & Qiu, 2022). In general, there are three steps in the estimation process: (i) treat the observations of treated counterfactuals as missing values in a matrix and build the predictive model with only untreated observations; (ii) impute the counterfactual outcomes as a prediction problem; (iii) obtain the estimated effect by taking an average of the differences between actual treated outcomes and predicted untreated outcomes for the treated units (Pan & Qiu, 2022). The details of the algorithm can be found in Athey et al., (2021) and Liu et al., (2022).

As the fixed-effects DID method is widely used to identify causality in time-series cross-sectional data, we use it as an alternative identification strategy and present the results in robustness checks. We additionally use the fixed-effects estimator (FEct) to demonstrate the robustness of our findings.

3. Results

We employ the matrix completion (MC) estimator as our main identification strategy to investigate the impact of the vampire attack on the performance of attacked liquidity pools (Athey et al., 2021).

3.1. The deposit-side impact of the vampire attack

The vampire attack initiated by the attacker Sushiswap cloned the incumbent Uniswap's platform

and aimed at draining liquidity from the incumbent. Naturally, the vampire attack would affect the liquidity of the attacked liquidity pools on the deposit side and the trading volume on the exchange side.

We first explore the impact of the vampire attack on the deposit-side performance of the attacked liquidity pools, measured by the *liquidity* (\$) of a pool till the end of a given day. We take the log transformation for the outcome variable and covariates owing to their skewed distributions. The MC estimator results of both matched and unmatched samples are presented in Table 3.

Table 3. Estimation of the impact of the vampire attack on the deposit side

	(1) Unmatched	(2) Matched
<i>Treat</i>	-0.2776 (0.2680)	-0.1061 (0.4503)
<i>Pool Age</i>	-0.0501 (0.1981)	16.9318 (6.6199)
<i>Listed Days</i>	-0.1798 (0.1591)	-0.6564** (0.2842)
<i>Open Price</i>	0.0442 (0.2510)	-0.2124 (0.2672)
<i>Close Price</i>	1.0171*** (0.2491)	0.9288** (0.3512)
<i>Top Price</i>	-0.0786 (0.1568)	-0.4735 (0.3414)
<i>Bottom Price</i>	0.0419 (0.3576)	0.5132 (0.3652)
<i>CEX Trading Volume</i>	0.0329 (0.0230)	0.0295 (0.0275)
<i>Market Cap</i>	-0.0083 (0.0170)	0.0051 (0.0259)
Observations	7,778	1,079
Pool Fixed Effects	Yes	Yes
Day Fixed Effects	Yes	Yes

Notes. Standard errors in the MC estimation are produced by parametric bootstraps of 2000 times. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

On the deposit side, one would expect the vampire attack to result in a significant decrease in the liquidity of the incumbent Uniswap's attacked liquidity pools because of the uniqueness of the vampire attack: (i) the attacker completely copies the open-source code of the incumbent's smart contracts and thus can provide perfect substitution of services with the same quality (*platform cloning*); (ii) the attacker provides its tokens to the participants as a reward (*tokenized incentives*);

(iii) only those who have deposit certificates from Uniswap can receive the token rewards offered by the attacker (*targeted attack*). Surprisingly, from consistent results shown in Columns (1) to (2) of Table 3, we find an *insignificant* effect of the vampire attack on the total liquidity of the attacked liquidity pools compared with those of the unattacked pools.

We plot the actual and counterfactual outcomes over time estimated through the MC estimator based on the matched sample in Figure 3. The black solid line represents the average outcome of attacked pools, and the blue dashed line represents the average counterfactual outcome of synthetic pools without attack. The white (shaded) area represents the pre-attack (post-attack) period. The vertical axis labels the log transformation of *liquidity* (\$) used in our estimations above, and the horizontal axis marks the days. Figure 3 shows a good fit of the outcomes in the pre-attack period, and the liquidity of attacked pools in the post-attack period is indistinguishable from that in synthetic untreated pools.

Interestingly, we find a surge in the treatment effects on liquidity in the first few days. One possible explanation may be that, at the end of the vampire attack, the attacker migrated the liquidity of the incumbent to its own platform. Some liquidity providers may withdraw their liquidity from the attacker's platform to the established incumbent in the first few days due to momentary panic since the migration day of the attacker was not announced in advance. Then, the liquidity providers' behavior may become rational and stable. We plan to collect user-level data to check such a conjecture.

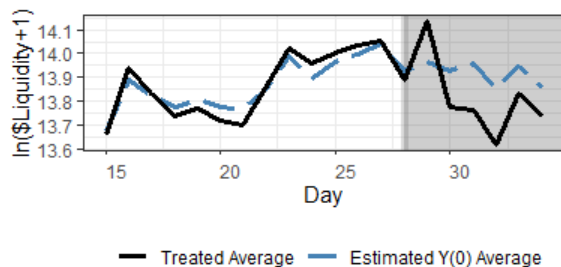


Figure 3. Actual vs. counterfactual outcomes on the deposit side over time

3.2. The exchange-side impact of the vampire attack

We conduct an analysis on the exchange side similar to that on the deposit side. We explore the impact of the vampire attack on the exchange-side performance of Uniswap's liquidity pools, measured by the total *trading volume* (\$) of a liquidity pool each day. We take

the log transformation for the outcome variable and covariates owing to their skewed distributions. The MC estimator results of both matched and unmatched samples are presented in Table 4.

On the exchange side, it is expected that the vampire attack hurts the exchange-side market since the attacker also can provide identical swapping services with identical quality (*platform cloning*). Naturally, this should divert traders away from the attacked liquidity pool of Uniswap, leading to a shrinking trading market for the incumbent. However, contrary to expectations, the vampire attack *expands* the overall trading market of the attacked liquidity pool as evidenced by Columns (1) to (2) of Table 4. For example, we find a positive and significant coefficient of 0.6103 in column (1), indicating the trading volume of the attacked pools is $(e^{0.6103}-1) \times 100\%=84.10\%$ higher than the attacked pools.

Table 4. Estimation of the impact of the vampire attack on the exchange side

	(1) Unmatched	(2) Matched
<i>Treat</i>	0.6103* (0.2590)	0.8385* (0.4272)
<i>Pool Age</i>	0.0494 (0.2687)	14.1067 (10.2195)
<i>Listed Days</i>	-0.6792*** (0.1843)	-0.3231 (0.4802)
<i>Open Price</i>	-0.3089 (0.5476)	-0.4628 (0.9752)
<i>Close Price</i>	0.1162 (0.5735)	-0.5249 (0.8506)
<i>Top Price</i>	2.0599*** (0.6670)	4.9442*** (1.3681)
<i>Bottom Price</i>	-0.6467 (0.5379)	-2.4906*** (0.8559)
<i>CEX Trading Volume</i>	0.3355*** (0.0441)	0.2783** (0.1422)
<i>Market Cap</i>	-0.0011 (0.0200)	0.0189 (0.0418)
Observations	7,778	1,079
Pool Fixed Effects	Yes	Yes
Day Fixed Effects	Yes	Yes

Notes. Standard errors in the MC estimation are produced by parametric bootstraps of 2000 times. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

We also plot the actual and counterfactual outcomes over time of the exchange side in Figure 4.

The vertical axis labels the log transformation of *trading volume* (\$), and other parameters stay the same. Figure 4 shows a good fit of the outcomes in the pre-attack period, and the liquidity of attacked pools in the post-attack period is much higher than that of the synthetic untreated pools.

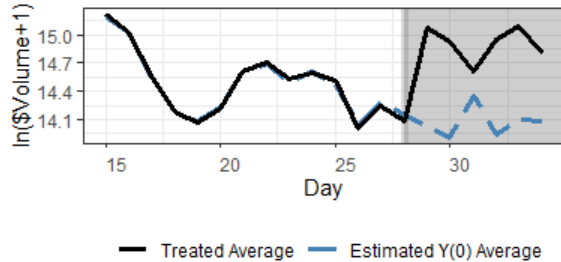


Figure 4. Actual vs. counterfactual outcomes on the exchange side over time

3.3. Discussions

We find a differential impact of vampire attack on the deposit side and exchange side. One plausible explanation is that potential users (including providers and traders) might be unaware of the attacked liquidity pools of the incumbent Uniswap (called “unaware users”) or be aware of it but have not provided liquidity or traded yet (called “aware users”) (Wang et al., 2018). After being exposed to a vampire attack, the unaware users become aware of the attacked pools (the informational effect) and the aware users subsequently provide liquidity or trade on the attacked pools (the persuasion effect). Although the expectable competition effect (He et al., 2020) from the vampire attack hurts the attacked liquidity pool, the informational and persuasion effects benefit the attacked liquidity pool. On the deposit side, the competition effect may be strong enough to dampen the informational and persuasion effects, resulting in an overall insignificant effect.

However, on the exchange side, the competition effect may be suppressed by the informational and persuasion effects to result in an overall positive effect. The competition effect on the deposit side may be much stronger than that on the exchange side because users on the deposit side (i.e., liquidity providers) can get token rewards from the attacker if they switch to the attacker. Differently, the users on the exchange side (i.e., traders) cannot since the vampire attack is targeted at the deposit side only. The different magnitude of the competition effects resulting from tokenized incentives on one side versus their absence on the other side lead to the overall insignificant effect on the deposit side but positive and significant effect on the exchange side.

4. Robustness checks

We additionally use the conventional fixed-effects DID estimation and another counterfactual estimator – fixed-effects estimator (FEct) to demonstrate the robustness of our findings.

Specifically, we implement the DID method using a two-way fixed-effects model with specification (1):

$$Y_{it} = \gamma_i + \delta_t + \beta Treat_{it} + \sum_k \tau_k X_{it} + \varepsilon_{it}, \quad (1)$$

where the dependent variable Y_{it} includes the outcome variable of interest for pool i on day t . γ_i and δ_t are the fixed effects for liquidity pool i and day t , respectively. $Treat_{it}$ is a dummy variable equal to one if an attacked liquidity pool i has suffered a vampire attack on day t . X_{it} is the time-varying liquidity pool-related variable that might affect the dependent variable.

Similar to our previous analysis, we examine the impact of the vampire attack on the deposit-side and exchange-side performance of the attacked liquidity pools. To ensure that DID is appropriate for our data, we check the parallel assumption by a standard event-study method (Cao et al., 2021). On both the deposit and exchange sides, we find parallel trends hold. Thus, we are able to use the DID as our alternative method.

The DID estimation results of both matched and unmatched samples are presented in Table 5. We find that the vampire attack has an insignificant effect on the deposit side performance of the liquidity pool but a positive and significant effect on the exchange side. These results are consistent with our estimations by the MC estimator.

Table 5. DID estimation of the impact of the vampire attack on the deposit and exchange sides

	Deposit side	
	(1) Unmatched	(2) Matched
<i>Treat</i>	-0.2733 (0.2711)	-0.1468 (0.4283)
	Exchange side	
	(3) Unmatched	(4) Matched
<i>Treat</i>	0.6059* (0.2590)	0.8671* (0.3889)
Observations	7,778	1,079
Covariates	Yes	Yes
Pool Fixed Effects	Yes	Yes
Day Fixed Effects	Yes	Yes

Notes. Standard errors in parentheses are clustered at the pool level. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

The FEct estimator shares a similar procedure as the conventional two-way fixed effects DID model. However, the conventional DID model is not robust for condition with a significant treatment heterogeneity (de, Chaisemartin & D’Haultfœuille, 2020) while the FEct estimator is still robust when faced with heterogeneous treatment effects (Liu et al., 2022). The FEct estimation results of both matched and unmatched samples are presented in Table 6. All results are consistent with our estimations by the MC and the conventional DID estimation.

Table 6. FEct estimation of the impact of the vampire attack on the deposit and exchange sides

	Deposit side	
	(1)	(2)
	Unmatched	Matched
<i>Treat</i>	-0.0512 (0.2958)	0.0259 (0.4722)
	Exchange side	
	(3)	(4)
	Unmatched	Matched
<i>Treat</i>	1.2279*** (0.2241)	1.1924*** (0.3297)
Observations	7,778	1,079
Covariates	Yes	Yes
Pool Fixed Effects	Yes	Yes
Day Fixed Effects	Yes	Yes

Notes. Standard errors in parentheses are clustered at the pool level and are produced by jackknife method. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

5. Conclusions

Our study examines a unique platform entry strategy in the blockchain ecosystem – vampire attack. As decentralized exchanges (DEXs) grow dramatically in the crypto market, vampire attacks on DEXs become increasingly popular. A vampire attack, made possible by the unique features of blockchain technology, is a competition strategy launched by an entrant platform that clones the targeted incumbent platform by copying its smart contracts and drains the incumbent’s liquidity away with token rewards to the incumbent’s liquidity providers. The nature of vampire attacks and the associated frightening image with the term “vampire” has fueled extensive misleading coverage in news and media reports. Unfortunately, there is a lack of literature examining the impact of the vampire attack on the operational performance of the targeted platform most likely due to the fact that vampire attacks are a rather nascent competition strategy.

In this study, we implement a quasi-experiment design to examine the impact of the first and the most famous vampire attack launched by Sushiswap against Uniswap. Specifically, we examine both the deposit-side and exchange-side impact of the vampire attack on the operational performance of liquidity pools on the incumbent platform. On the deposit side, surprisingly, we find that the vampire attack triggers no significant decline in the total liquidity of attacked liquidity pools. On the exchange side, even more surprisingly, the vampire attack significantly *increases* the trading volume of liquidity pools. The results preliminarily show that targeted tokenized incentives play a key role in forming the negative impact of vampire attacks.

Our study contributes to both the literature on DEX and the competitive strategy of entrant platforms. As for managerial implications, our study helps the users, DEX platforms, and policymakers understand the true impact of the vampire attack, which refutes some misleading claims of the adverse impact on the incumbent in extensive news and media reports (Bradley et al., 2020).

We plan to further uncover the underlying reasons that contribute to these intriguing results. Our study has several limitations and provides directions for future research. First, our study focuses on vampire attacks in the context of decentralized exchanges. Future studies can examine vampire attacks in different contexts, such as nonfungible tokens (NFT). Second, our study focuses on the short-term effect of the vampire attack because the incumbent platform, Uniswap, released its token one week later. Future studies can examine the vampire attack’s long-term effect when the data is available. Future studies can also examine the defensive strategies of the incumbent platform when facing the threat of a vampire attack. Third, our study is conducted at the pool level. Future studies can examine the impact of vampire attacks at the individual level.

6. Acknowledgements

Xi Zhao is a Tang Scholar. This work is supported by the National Natural Science Foundation of China (Grant No. 72231007), The Key Research and Development Projects of Shaanxi Province (Grant No. 2020ZDLGY09-08).

References

- Amiram, D., Jørgensen, B. N., & Rabetti, D. (2022). Coins for bombs: The predictive ability of on-chain transfers for terrorist attacks. *Journal of Accounting Research*, 60(2), 427–466. <https://doi.org/10.1111/1475-679X.12430>
- Athey, S., Bayati, M., Doudchenko, N., Imbens, G., & Khosravi, K. (2021). Matrix completion methods for

- causal panel data models. *Journal of the American Statistical Association*, 116(536), 1716–1730. <https://doi.org/10.1080/01621459.2021.1891924>
- Benedetti, H., & Nikbakht, E. (2021). Returns and network growth of digital tokens after cross-listings. *Journal of Corporate Finance*, 66, 101853. <https://doi.org/10.1016/j.jcorpfin.2020.101853>
- Bradley, K., Omkar, G., & Sebastian, S. (2020). *First Mover: DeFi “Vampire” SushiSwap Sucks \$800M from Uniswap; BitMEX Basis Lags*. Coindesk. <https://www.coindesk.com/markets/2020/09/10/first-mover-defi-vampire-sushiswap-sucks-800m-from-uniswap-bitmex-basis-lags/>
- Bronnenberg, B. J., Dubé, J. P., & Sanders, R. E. (2020). Consumer misinformation and the brand premium: A private label blind taste test. *Marketing Science*, 39(2), 382–406. <https://doi.org/10.1287/mksc.2019.1189>
- Cao, G., Jin, G. Z., Weng, X., & Zhou, L.-A. (2021). Market-expanding or market-stealing? Competition with network effects in bike-sharing. *RAND Journal of Economics*, 52(4), 778–814.
- Capponi, A., & Jia, R. (2021). The adoption of blockchain-based decentralized exchanges. *Available at SSRN*, <https://ssrn.com/abstract=3805095>.
- de, Chaisemartin, C., & D’Haultfœuille, X. (2020). Two-way fixed effects estimators with heterogeneous treatment effects. *American Economic Review*, 110(9), 2964–2996. <https://doi.org/10.1257/aer.20181169>
- Han, J., Huang, S., & Zhong, Z. (2021). Trust in DeFi: An empirical study of the decentralized exchange. *Available at SSRN*, <https://ssrn.com/abstract=3896461>.
- He, S., Peng, J., Li, J., & Xu, L. (2020). Impact of platform owner’s entry on third-party stores. *Information Systems Research*, 31(4), 1467–1484. <https://doi.org/10.1287/isre.2020.0957>
- Jackson, R. (2022). *KLEX Protocol Launches “Vampire Attack” against KLAYswap to Attract Liquidity*. Nasdaq. <https://www.nasdaq.com/articles/klex-protocol-launches-vampire-attack-against-klayswap-to-attract-liquidity>
- Liu, L., Wang, Y., & Xu, Y. (2022). A practical guide to counterfactual estimators for causal inference with time-series cross-sectional data. *American Journal of Political Science*, *Forthcoming*. <https://doi.org/10.1111/ajps.12723>
- Onyshchenko, S. (2020). *How to Prevent Liquidity Vampire Attacks in DeFi? Blaize*. <https://blaize.tech/article-type/web3-security/how-to-prevent-liquidity-vampire-attacks-in-defi/>
- Pan, Y., & Qiu, L. (2022). How ride-sharing is shaping public transit system: A counterfactual estimator approach. *Production and Operations Management*, 31(3), 906–927. <https://doi.org/10.1111/poms.13582>
- Reif, N. (2021). *What are centralized cryptocurrency exchanges?* Investopedia. <https://www.investopedia.com/tech/what-are-centralized-cryptocurrency-exchanges/>
- Schär, F. (2021). Decentralized finance: On blockchain- and smart contract-based financial markets. *Available at SSRN*, <https://ssrn.com/abstract=3843844>.
- Shen, B., Dong, C., & Minner, S. (2021). Combating copycats in the supply chain with permissioned blockchain technology. *Production and Operations Management*, 31(1), 138–154. <https://doi.org/10.1111/poms.13456>
- Wang, Q., Li, B., & Singh, P. V. (2018). Copycats vs. original mobile apps: A machine learning copycat-detection method and empirical analysis. *Information Systems Research*, 29(2), 273–291. <https://doi.org/10.1287/isre.2017.0735>
- Xia, P., Wang, H., Gao, B., Su, W., Yu, Z., Luo, X., Zhang, C., Xiao, X., & Xu, G. (2021). Trade or trick? Detecting and characterizing scam tokens on Uniswap decentralized exchange. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 5(3), 1–26. <https://doi.org/10.1145/3491051>
- Xu, Y. (2017). Generalized synthetic control method: Causal inference with interactive fixed effects models. *Political Analysis*, 25(1), 57–76. <https://doi.org/10.1017/pan.2016.2>