Skolkovo Institute of Science and Technology

MASTER'S THESIS

Economic Data Mining and Analysis of MakerDAO DeFi Project

Master's Educational Program: Data Science

Student: ______________ Sudarut Kasemsuk
signature

Research Advisor: ______________ Yury Yanovich
signature
PhD, Senior Research Scientist

Moscow 2023

Skoltech

Skolkovo Institute of Science and Technology

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Сбор и анализ экономических данных проекта MakerDAO DeFi

Магистерская образовательная программа: Науки о данных

Студент: ______________ Сударат Касемсук

подпись

Научный руководитель: ______________ Янович Юрий Александрович к.ф.-м.н., старший научный сотрудник

подпись

Москва 2023

Economic Data Mining and Analysis of MakerDAO DeFi Project

Sudarut Kasemsuk

Submitted to the Skolkovo Institute of Science and Technology on October 11, 2023

ABSTRACT

This research study is a comprehensive approach to improving the analytical capabilities of the MakerDAO platform, a decentralized protocol built on the Ethereum blockchain. The primary goal of this study is to eliminate the discrepancy between raw Ethereum smart contract data and structured data formats that can be used for detailed statistical analysis. The initial component of the task entails the creation of utility functions. These functions are intended to convert the intricate and frequently complex data produced by Ethereum smart contracts into a structured and standardized approach. Various statistical analyses become more feasible as a result, allowing insights into crucial platform parameters like collateral ratio and loss given default. Researchers, analysts, and participants can acquire a better understanding of the platform's health and risk profile as a result of this procedure. This study also reaches into an issue of interest rate determination. Interest rates in financial systems are traditionally set using a variety of parameters, but the MakerDAO platform is different because the interest rate has been changing over the time. The study intends to apply optimization approaches to develop an individual interest rate for each user based on their unique profile and activity on the platform. This not only enhances the fairness of interest rate setting, but also helps to optimize the overall financial efficiency of the ecosystem. Another important aspect of this research is the development and application of a probability of default model. This model, designed specifically for single collateral assets within the MakerDAO architecture, aims to calculate the probability of a user defaulting on their collateral-backed loan.

Keywords: blockchain, decentralize finance, dataset, smart contract, data mining, Brownian motion

Research advisor:
Name: Yury Yanovich
Degree, title: PhD, Senior Research Scientist

Contents

Chapter 1
Introduction

1.1 Overview

The financial industry's business sectors play an important role in establishing the modern economic environment, serving as critical components of the larger financial system. These industries include an extensive variety of service providers who all contribute to the intricate web of financial operations. The financial industry has evolved through time into a complicated institution comprised of several segments, each providing unique services adapted to the needs of individuals, enterprises, and economies at large. These financial institutions' growth and development have been driven not only by domestic regulatory agencies, but also by international standards such as the Basel framework(BF), the Basel framework is the full set of standards of the Basel Committee on Banking Supervision, which is the primary global standard setter for the prudential regulation of banks [6]. As of mid-2023, 28 jurisdictions covering a half of the humanity population use the BF. The key quantities of a loan are an interest rate, a loss given default, and a probability of the default. But the real bank data to compute the parameters is a part of bank secrecy and may contain sensitive personal information [40], hence it is not openly available.

Confidentiality and privacy of sensitive data are critical in the banking and financial industries. While the Basel Committee on Banking Supervision (BCBS) [6] and other regulatory agencies set norms and criteria for banks to follow in order to assure their stability and risk management, they also acknowledge the need to safeguard sensitive financial and customer data. Banks and financial institutions handle sensitive information such as customer financial data, unique trading tactics, risk management procedures, and more. This information is extremely sensitive and must be kept secure to avoid unauthorized access, fraud, and potential disruptions to financial systems. Central banks and financial supervisory agencies, for example, have mechanisms in place to collect essential data from banks in order to assess their financial health and compliance with regulatory criteria. However, in order to avoid market disruptions and to protect the privacy of both individuals and organizations, these agencies treat the data with strict confidentiality. Financial data collected by regulatory organizations is often aggregated and anonymized before being used for analysis and decision. This helps to protect individual banks' specific information and sensitive customer data. This aggregate data is used for monitoring in many aspects to improve the financial system. Overall, the delicate balance of accountability, disclosure, and data privacy highlights the difficulty of regulating the financial industry while protecting sensitive information.

Decentralized finance (DeFi)–peer-to-peer financial services on public blockchains [9, 10]–not only bring new web3-based services but also provide analogs of traditional financial instruments [34, 29]. So, Maker protocol resolves crypto-backed loans [25]. The set of smart contracts implement Maker protocol on the Ethereum blockchain, and a decentralized autonomous organization (DAO) named MakerDAO governs the project, including economic parameters assignment. Maker's smart contracts are deployed on Ethereum blockchain, so one can view all the related transactions. The financial information hiding, like transaction amount, is technically possible [7], but makes the protocol significantly more complicated and slower, increases transaction fees and lowers transparency for participants, hence makes the project less reliable. Maker does not encrypt transaction amounts, and the only hidden things are real-world entities behind the protocol users:

one can see user identifiers but not names.

Although of transaction transparency of blockchain-based projects, they lack regulation and standards [33, 15]. The goal of the current research is to consider a lending in DeFi project Maker from a traditional finance point of view. The outcome is two-fold: we provide a real landing portfolio dataset and equip it with the standard banking numerical parameters.

1.2 Related Work

Banks use a variety of tools to maintain reasonable risk levels and increase efficiency, including regulator-required frameworks like the Basel framework [6] and machine learning models [27, 21]. Loan portfolio data has also attracted researchers' attention, with some studies accessing proprietary data that is not publicly available. For example, Hayden et al. [17] and Rossi et al. [32] examine the impact of loan portfolio diversification on risk and capital efficiency based on German and Australian large banks, respectively. They have an access to more than thousand individual bank portfolios over seven years. Serengil et al. [36] compare various machine learning techniques to predict non-performing loans using a portfolio dataset provided by a bank for four years, consisting of 181 thousand borrowers and hundreds of features. Annisa and Rusdah [4] apply random forest to classify non-performing loans for Indonesia's bank loan dataset with 3300 borrowers and 12 features.

Several classic finance datasets are publicly available, such as Home Credit Default Risk on Kaggle, which challenges participants to predict the probability of default among 307 thousand debts using 239 features [3]. The UC Irvine Machine Learning repository contains several credit datasets, with the Taiwan credit card defaults dataset being the largest, containing 30 thousand debts and 24 features [19]. The peer-to-peer lending platform Lending Club provides a dataset containing 887 thousand debts collected from 2007 until 2015 with 79 features to predict the probability of the default [11].

In addition to traditional finance, there have been several studies on decentralized finance (DeFi) and its potential impact on traditional finance. For instance, Zetzsche et al. [42] analyze the challenges and opportunities of DeFi in the financial industry, highlighting the need for regulatory frameworks and standards to ensure its stability and security. Similarly, Schar [35] investigates the risks and benefits of DeFi from the perspective of financial intermediation and proposes a framework for analyzing DeFi projects.

While data scientists have limited access to loan data from traditional banking due to trade secrecy and privacy concerns, DeFi loan data is openly available on public blockchains. Several studies have been conducted on DeFi lending platforms specifically, such as Maker, AAVE, and Compound [25, 22, 2, 23], including their data collection, economic parameters estimation, and risk management. For example, the paper [14] analyzes data from the decentralized Ethereum protocol called Compound, using a relational database and providing statistical details to facilitate further analysis. Kjaer et al. [22] assess the stability of the DAI stablecoin of the MakerDAO project over the course of its first year of full protocol, including the cryptocurrency crisis in March 2020. Azoulay et al. [5] discuss the issue of high collateral requirements for blockchain-based loans using cryptocurrencies as collateral due to their high volatility and propose a solution to make loans more accessible by offering lower collateral requirements while keeping risk for lenders bound.

However, there has been limited research on providing a real lending portfolio dataset for Maker and equipping it with standard banking numerical parameters. Such a dataset could be useful for both academic research and practical applications, such as risk management and portfolio optimization in Maker lending.

1.2.1 Maker Protocol for Borrower

The Maker Protocol [25] operates using the native DAI token, which has a one-to-one soft peg to the United States dollar and is an ERC-20 token [39]. The protocol allows for collateral-secured DAI debts, with loan terms such as financial parameters and a DAO mechanism to change them included in its smart contract. Financial parameters, such as the lending interest rate f (the multiplier applied to the loan balance over time) and the liquidation ratio r (the minimum allowed ratio of the locked collateral value to the debt value), are examples of these loan terms. Users can deposit Ethereum or other tokens into their instance of a specific smart contract Vault and use them as collateral to mint DAI debt, see Figure 1.1.

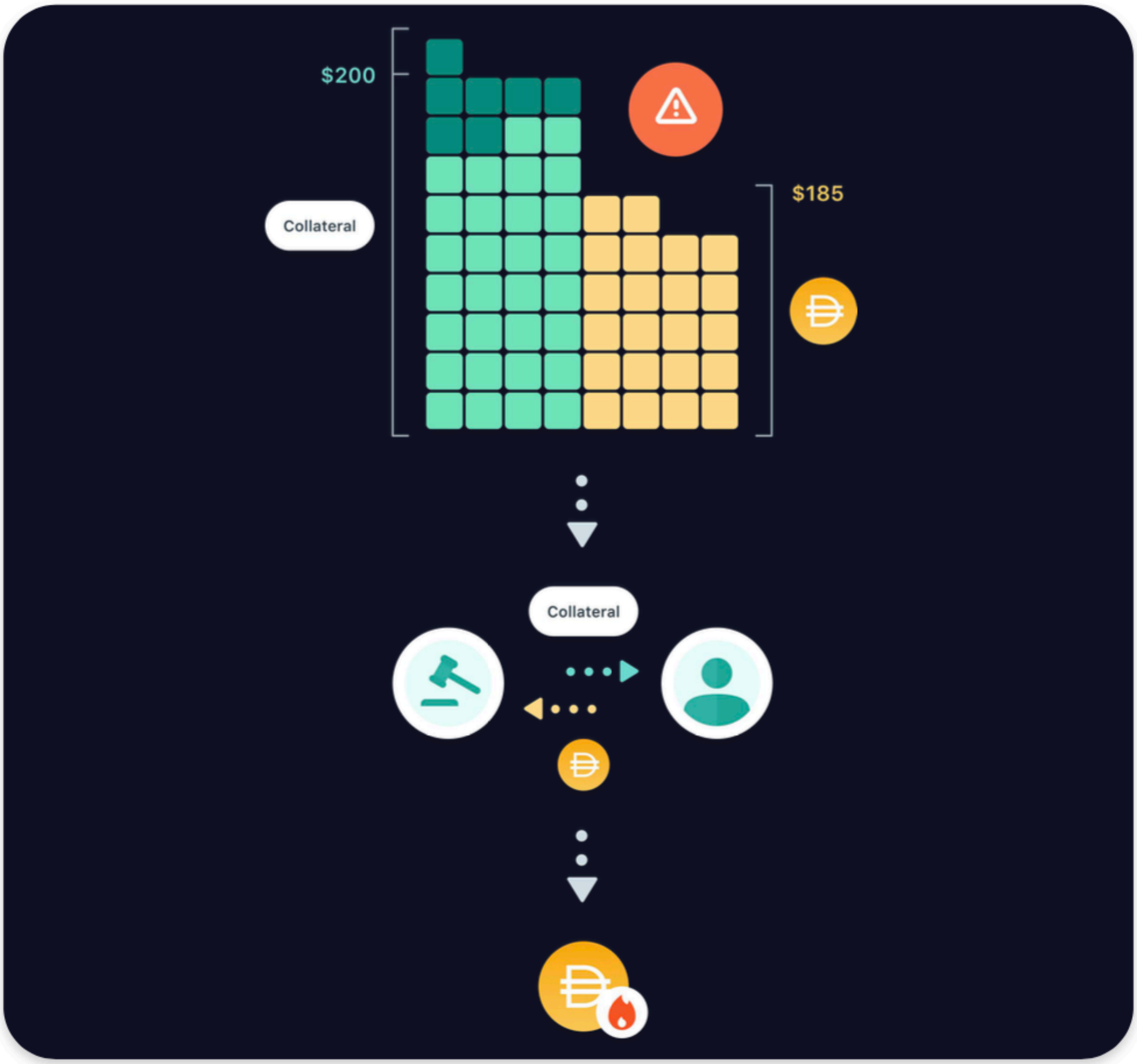

Figure 1.1: The liquidation process on MakerDAO platform [24]

Let's consider a borrower's workflow in the Maker Protocol. The borrower starts by creating a Vault and depositing supporting collateral. The Maker Protocol's Oracle then evaluates the collateral, providing a real-time price feed for each asset. Based on the current market value of the collateral and the chosen borrowing program, the protocol calculates the maximum amount of DAI that can be borrowed. For example, ETH-A, ETH-B, and ETH-C are different borrowing programs with Ethereum cryptocurrency (ETH) as collateral, each with its own set of parameters and risk profiles. ETH-A is the original and most commonly used collateral type, while ETH-B and ETH-C were introduced to offer additional options for users with different risk tolerances or preferences.

Once the borrower has minted DAI, they can use it for any purpose. The borrower is responsible for repaying the loan with interest, which is calculated based on the lending interest rate and duration of the loan. They can fully or partially repay the loan at any time, borrow more up to the maximum permitted collateral program and size of the collateral amount, or increase or decrease the amount of collateral.

If the value of the collateral falls below a certain threshold, the Vault is at risk of being liquidated. In this case, the MakerDAO system will automatically initiate a liquidation process, which involves selling off a portion of the collateral to cover the outstanding debt. The borrower can not interact with the Vault under the liquidation process. The liquidation process is designed to be fast and efficient, with the goal of minimizing losses for both the user and the MakerDAO system. When a Vault is liquidated, the collateral is sold through an auction, allowing users to bid on the collateral using DAI. The auction is competitive, with bidders offering progressively lower prices until the collateral is sold.

If the auction is successful and the collateral is sold for a price that covers the outstanding debt, the remaining DAI is returned to the user. If the auction is unsuccessful and the collateral is not sold for a sufficient price, the MakerDAO system may take a loss on the liquidation. The resulting penalty for the borrower is flexible but usually ranges from 10% to 33%.

All actions involving the Vault and system parameters are recorded as plaintext Ethereum blockchain transactions. However, these transactions may be challenging for the general audience to understand due to Maker's use of technical terms such as ilk, frob, and art. To address this issue, we aim to present the loan portfolio dataset from the Maker project in a more accessible format.

1.3 Aim

Considering the MakerDAO platform as a significant data source[24], we use Web3.0's sophisticated features to handle and analyze the huge amount of information contained inside this dynamic loan environment. The MakerDAO framework, a complex orchestration of components, serves as the foundation for our data investigation. Our strategy begins with the punctual collection of raw data from the MakerDAO network. This raw data, which contains a wealth of insights, is then neatly turned into an intelligible and interpretable format. This metamorphosis is the result of a harmonious marriage between MakerDAO's ABI contract, which serves as an interface to its smart contracts, and Python's Web3 module, a versatile tool that allows the extraction of hexacode data. We have access to a number of essential indicators that drive the MakerDAO ecosystem by smoothly integrating these tools. We examine the complexity of Maker Vaults, determining the makeup of assets within, while concurrently investigating the Dai debt that powers this novel lending mechanism. Through this rigorous investigation, we also uncover the complex equilibrium of collateral balances, shedding light on the platform's core security measures. Nevertheless, our analytical path does not end with assets and liabilities. We continue the process of capturing the constantly changing DAI/ETH exchange rate, an important dynamic that represents a wider market's impact on the MakerDAO ecosystem. However, as we dig into the complex terrain of financial data, the scope of our study increases beyond the visible sphere. We uncover a critical aspect of risk management using the Loss Given failure (LGD), an indicator of possible losses in the case of borrower failure. In the meantime, the Effective Interest Rate, a complex indicator that captures the actual price of borrowing, emerges as an important benchmark in our analytical toolbox. In addition to these risk-based indicators, the Probability of Default (PD) provides information into the likelihood of borrowers failing on their obligations, improving our comprehension of the platform's dynamics. In summary, our methodology is to work with MakerDAO's sophisticated design, Web3.0's transformative potential, and Python's information manipulation prowess. It allows us to extract, evaluate, and dissect a wide range that together offer a comprehensive picture of the MakerDAO system. This study need to work with the hexacode, decoding critical insights.

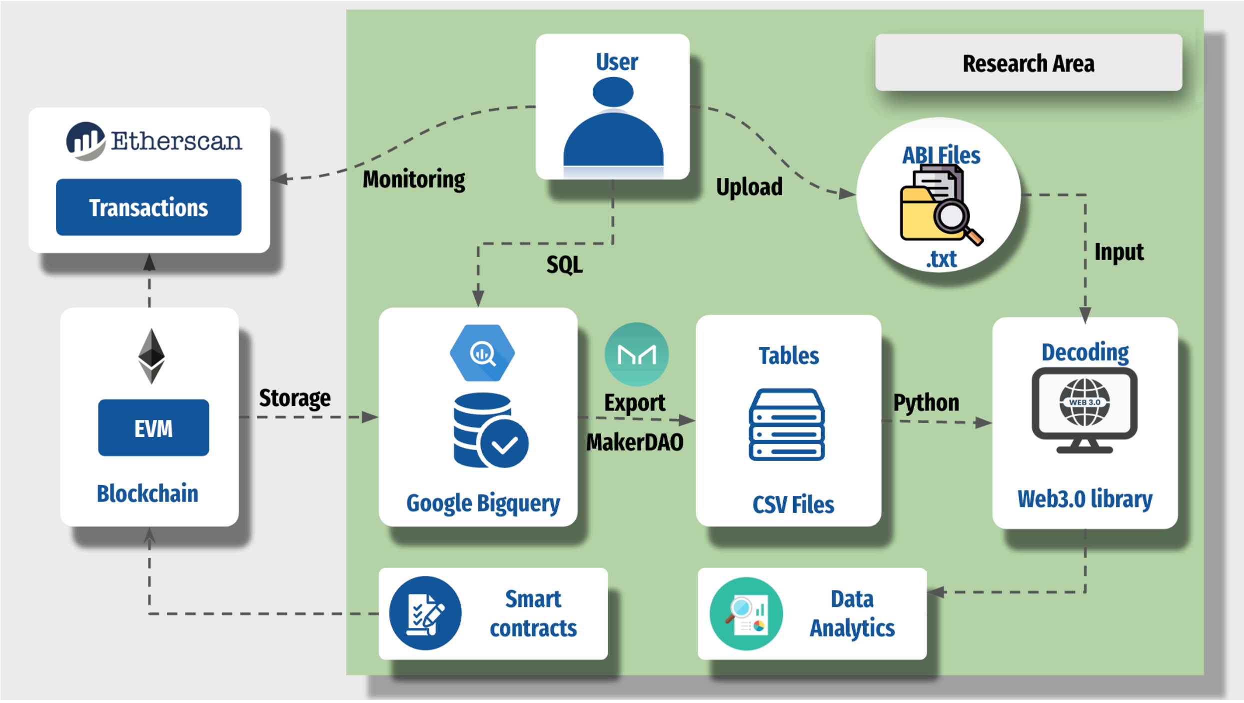

Figure 1.2: Overview of research area

1.4 Objectives

In basically, the objective of this research project is to construct a complete dataset, infuse it with relevant financial characteristics, and develop a mathematical model designed to estimate the likelihood of default for loans within the MakerDAO ecosystem. This multimodal method seeks to provide useful insights into the risk assessment and lending dynamics of decentralized finance platforms on MakerDAO.

Create a Comprehensive Dataset: The first sub-goal is to create a robust dataset encompassing loan portfolios originating from MakerDAO. This dataset will act as the primary source of information for the succeeding phases of the project. It will feature a diversified range of loan portfolios with varying financial characteristics, lending histories, and borrower profiles.

Enhance Dataset with Financial features: The second sub-goal entails enhancing the dataset with essential financial features related to borrowing. Loan amounts, interest rates, collateral kinds, and borrower profiles are examples of these features. This enrichment is critical for a thorough examination of borrowing habits inside the MakerDAO ecosystem.

Create and Improve a Specialized Probability of Default Model: The third sub-objective focuses on the development and improvement of a specialized mathematical model. The goal of this model is to assess and predict the likelihood of default in the context of MakerDAO's financing activity. The model will combine several parameters based on the enriched dataset to assess the chance of borrowers defaulting on their obligations.

1.5 Literature review

1.5.1 Blockchain and smart contacts

Blockchain is a digital ledger technology that connects all parties by storing databases on each node [10]. Each node stores transactions that occur on the platform, and no one can delete/rewrite or edit them, making the blockchain immutable. If they do, they must do so for all nodes on the platform, which is impossible to manage. According to blockchain concepts, when a user conducts a transaction, there will be peer and peer node nodes with no central party. After transactions are written into blocks, each block contains a unique cryptographic hash, and they are broadcasted and copied to all nodes, resulting in all nodes having the same information and each block being linked by chain. All transactions can be traded back to see their history.

Smart contacts are automatically generated contacts that are stored and used on the blockchain platform. When actions on the platform satisfy the conditions of smart contacts, they will be processed automatically. To reassure people and reduce the possibility of fraud or error [43]. For example, if user A promises to do something with user B, when the transaction is successful, the balance from user B's account will be deducted. Smart contacts are deployed on the blockchain, which means there are no third parties to control the transaction. These procedures ensure that smart contacts are transparent, secure, and auditable. Smart contacts can be used in a variety of industries, including finance, insurance, and education. However, there is a significant factor for use, smart contacts require validators to check integrity and precision to ensure that they are working properly.

1.5.2 Decentralized applications(DeFi)

Decentralization Finance (DeFi) is a financial system [8] based on blockchain technology that uses peer-to-peer networking to make transactions low cost, transparent, and accessible. Users with internet access can access decentralized applications (dapps). DeFi transactions are immutable and recorded on the public blockchain. DeFi applications are built on blockchain platforms such as Ethereum and provide users with access to financial services such as lending, borrowing, trading, and investing. Some popular DeFi application platforms include lending platforms, decentralized exchanges (DEXs), and stablecoins.

DeFi, on the other hand, is not without risks. Smart contracts can be vulnerable, and there have been reports of DeFi platform hacks and exploits. DeFi is also fraught with regulatory uncertainty, as regulators wrestle with how to apply existing regulations to these new decentralized financial systems. Despite these obstacles, DeFi is a rapidly expanding field, with billions of dollars in value locked in DeFi protocols and a thriving ecosystem of developers and users building and utilizing these platforms.

1.5.3 Ethereum

Ethereum is a decentralized [39], open-source blockchain platform founded in 2015 by Vitalik Buterin and a team of developers. Ethereum is intended to be a programmable, flexible platform that can support a wide range of decentralized applications (Dapps), such as smart contracts and decentralized finance (DeFi) applications. The Ethereum Virtual Machine (EVM), a runtime environment that executes smart contracts on the Ethereum network, is a key feature of Ethereum. Smart contracts are self-executing programs that automatically carry out the terms of a contract between two or more parties, and they are a key component of many decentralized applications.

In addition to smart contracts, Ethereum supports a native digital currency called Ether (ETH), which is used to pay transaction fees and to reward miners for network security. After

Bitcoin, Ether is the second-largest cryptocurrency by market capitalization. Since its inception, Ethereum has undergone several upgrades, including the recent London hard fork, which introduced several new features, such as the Ethereum Development Proposal (EIP)-1559, which aims to improve the efficiency of the Ethereum network by introducing a new fee structure.

Ethereum has evolved into a critical infrastructure for the development of decentralized applications and the blockchain ecosystem as a whole. Many DeFi applications, like as decentralized exchanges (DEXs), lending platforms, and stablecoins, are constructed on top of the Ethereum network. The Ethereum network also serves as the foundation for a large number of non-fungible tokens (NFTs), which are one-of-a-kind digital assets that can represent anything from artwork to collectibles.

1.5.4 MakerDAO

MakerDAO is a platform for decentralized finance (DeFi) formed on the Ethereum blockchain. It is one of the most widely used DeFi protocols, with the goal of providing a stablecoin called DAI that is pegged to the US dollar. MakerDAO generates DAI through a new mechanism of collateralized debt positions (CDPs). Users can deposit cryptocurrency ether (ETH) into a CDP, which then generates DAI. To ensure that the DAI generated is adequately collateralized, the value of the ETH in the CDP must remain above a certain threshold.

The DAI generated by users can be used to access a variety of DeFi services, such as lending and borrowing, trading, and investing. DAI is an appealing option for those who want to use cryptocurrency for transactions but want to avoid the volatility of other cryptocurrencies such as Bitcoin because it is stable. MakerDAO is governed by a decentralized autonomous organization (DAO) that is controlled by MKR token holders. MKR token holders can vote on proposals to change the protocol or its governance structure, and they are accountable for the DAI token's stability.

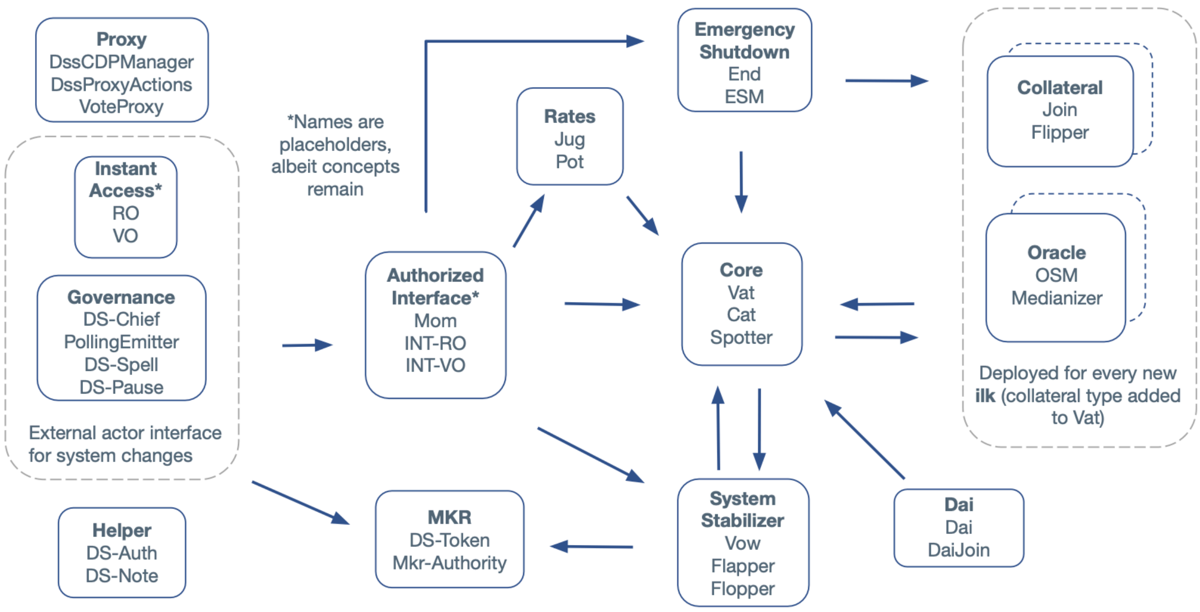

Figure 1.3: MakerDAO's Smart Contract Modules[24]

1.5.5 Geometric Brownian Motion

The ”Geometric Brownian Motion” (GBM) is popular model for using in a mathematical model [13], it is used in finance to describe the stochastic movement of the price of an asset over time [18]. GBM is named after its usage of both a geometric (exponential growth) and a Brownian motion (random fluctuation) component. The following is the theory underlying Geometric Brownian Motion: **Components of Geometric Brownian Motion:**

- **Deterministic Drift** (μ)**:** expected return or every growth rate of the asset's price
- **Volatility** (σ)**:** σ typically estimated from historical price data
- **Time** (t)**:** time period about asset's price
- **Wiener Process** (dW_t)**:** continuous-time stochastic process

The Geometric Brownian Motion model is defined by the following stochastic differential equation:

$$dS_t = \mu S_t dt + \sigma S_t dW_t \tag{1.1}$$

Where:

S_t is the price of the asset at time t.
dS_t is the change in the asset price over a small time increment dt.
μ is the deterministic drift or expected return.
σ is the volatility.
dW_t is the Wiener process (Brownian motion) term.

1.5.6 Passage times of levels

The passage time distribution [1] is a probability distribution that indicates how likely it is for a stochastic process to reach a specified level within a given time interval. It's a basic term in the study of stochastic processes, particularly continuous-time processes like Brownian motion, where you want to know how long it takes for the process to reach a certain level. The passage time distribution reveals information on the timing of such events.

The probability density function (PDF) $f(t)$ can explain the passing time distribution for a continuous-time stochastic process $X(t)$ crossing a given level a for the first time. The PDF represents the likelihood of reaching the level a at a given time t.

The passage time distribution formula is often expressed as:

$$f(t) = \frac{d}{dt} P(T \leq t) \tag{1.2}$$

Where:

$f(t)$ is the probability density function (PDF) of the passage time.
$P(T \leq t)$ is the cumulative distribution function (CDF) of the passage time, representing the probability that the first passage time is less than or equal to t.
T is the random variable representing the first passage time.

1.6 Publications

1. Chaleenutthawut, Y., & Davydov, V. & Evdokimov, M. & Kasemsuk, S. & Kruglik, S. & Melnikov, G. & Yanovich, Y. (2023). ”Loan Portfolio Dataset from MakerDAO Blockchain Project.” Manuscript submitted for publication in the *IEEE Journals*.

1.7 Individual contributions

1. Create a dataset with loan portfolios from MakerDAO.
2. Equip it with financial characteristics related to borrowing.
3. Develop a specialized mathematical model for the probability of default.

Chapter 2
Problem statement

Due to concerns about privacy and the highly sensitive nature of financial data, accessing and using information from traditional financial institutions for research purposes can be difficult. Such institutions are subject to stringent rules and confidentiality obligations, which prohibit the public publication of information about customers. Given these limits, academics have resorted to new sources of financial data to analyze and investigate emerging financial trends and events.

Decentralized Finance (DeFi) data, which is available on blockchain networks, is one such alternate source. DeFi symbolizes a paradigm shift in the financial industry, ushering in a new era of decentralized and open-access financial products and services. DeFi, in contrast to traditional financial systems that are tightly regulated by centralized bodies, is founded on the ideals of transparency, immutability, and accessibility. Smart contracts, which are self-executing agreements with the terms of the contract between buyer and seller explicitly put into code, enable this breakthrough.

MakerDAO, a significant DeFi player, demonstrates this emerging financial environment. It is a loan and borrowing network that runs on the Ethereum blockchain. The introduction of the stablecoin Dai, which is utilized as an incentive for borrowers, is one of its distinguishing qualities. Dai is intended to hold its value over time, making it an appealing option for people seeking stability in a volatile cryptocurrency market.

The key challenges in this research can be summarized as follows:

1. **Data Accessibility:** Traditional financial banking data, which is rich in risk assessment insights, is often inaccessible due to privacy constraints. To address this, we intend to create a relevant dataset from publicly available DeFi sources, with a particular focus on loan portfolios within MakerDAO.

2. **Blockchain Data Interpretation:** While blockchain technology is secure and transparent, its non-human-readable format presents a unique issue. Data interpretation and comprehension on the blockchain are intrinsically difficult. Our goal is to provide solutions that make this data more understandable and interpretable, allowing for a better understanding of the borrowing and lending dynamics inside the MakerDAO ecosystem.

3. **Probability of Default Modeling:**We intend to create a specific mathematical model based on the dataset that has been created. This model will be used to forecast the likelihood of loan default within MakerDAO. The ability to forecast the likelihood of debtors defaulting on their payments is critical for risk assessment and financial decision-making.

This research project aims to close the data accessibility gap in the financial research domain by building a meaningful dataset from publicly available DeFi sources, making blockchain data more human-readable and understandable, and developing a specialized model for estimating the probability of default. The project intends to contribute to the knowledge and risk assessment of lending activities within the decentralized financial landscape by addressing these difficulties.

Chapter 3
Methodology

3.1 Mathematical Models

This study focuses on a single type of collateral. Moving forward, we will only examine this one specific type.

To obtain a loan in the Maker protocol, a user must have a Vault. Vaults are unique to each user and cannot be transferred. Ethereum addresses, which are 42-character hexadecimal strings, represent users. While an address does not reveal information about the real-world owner, some users may indirectly or directly disclose their identity. Even if anonymous, a user can have multiple Vaults, and we keep track of which Vault belongs to which user.

We can determine if a Vault has an active loan at a given time by checking if its DAI debt is non-zero. Therefore, we define the beginning of a loan as when the DAI debt changes from zero to a positive number. We define the end of a loan as when the DAI debt becomes zero from any non-zero value. A loan can be active or ended, either through successful repayment or liquidation. A single Vault can have multiple loans, and all loans have non-overlapping beginning-to-end time intervals.

3.1.1 Balance

When a user borrows `DAI` in the Maker project, they need to provide collateral. Without the loss of generality, we will refer to the collateral asset as `ETH`. The loan starts at t_0 and lasts until T, which can be either the liquidation time, full repayment time, or maximum observed time if the loan is still active at the point T.

The amount of collateralization assets at any given time t is denoted by $a(t)$ (an example is shown in Figure 3.1). The blockchain records updates to the collateral balance as a piece-wise constant function, represented by update times τ and corresponding changes $\Delta a(\tau)$ due to collateral deposits or withdrawals, and liquidation processes.

The maximum allowed debt is determined by the collateral price in `DAI` and the minimum allowed collateralization ratio $r_{\text{min}}(t)$. Oracles provide the `ETH/DAI` exchange rate $e(t)$, which is typically consistent with centralized exchange rates except in cases of extremely high transaction fees [22]. The minimum allowed collateralization ratio $r_{\text{min}}(t)$ is a piece-wise linear function with small slopes at non-constant intervals to ensure platform stability. Debts in Maker project are over-collateralized. So $r_{\text{min}}(t) > 1$.

Let $d(t)$ be the debt at time t (an example is shown in Figure 3.2). Interest is charged on the active debt, with the logarithm of the interest over time denoted by $f(t)$. If no actions are taken on the debt during an interval $(t_1, t_2]$, then the debt at time t_2 can be calculated as

$$d(t_2) = d(t_1) \cdot \exp\left(\int_{t_1}^{t_2} f(t)dt\right). \tag{3.1}$$

The log-interest $f(t)$ is piece-wise constant by design of the platform. If $f(t)$ is constant during $(t_1, t_2]$, then $d(t_2) = d(t_1) \cdot \exp\left(f(t_2) \cdot (t_2 - t_1)\right)$. So the collateral balance is piece-wise exponential. The function breaks are due to the debt repayment or getting more, and a liquidation process.

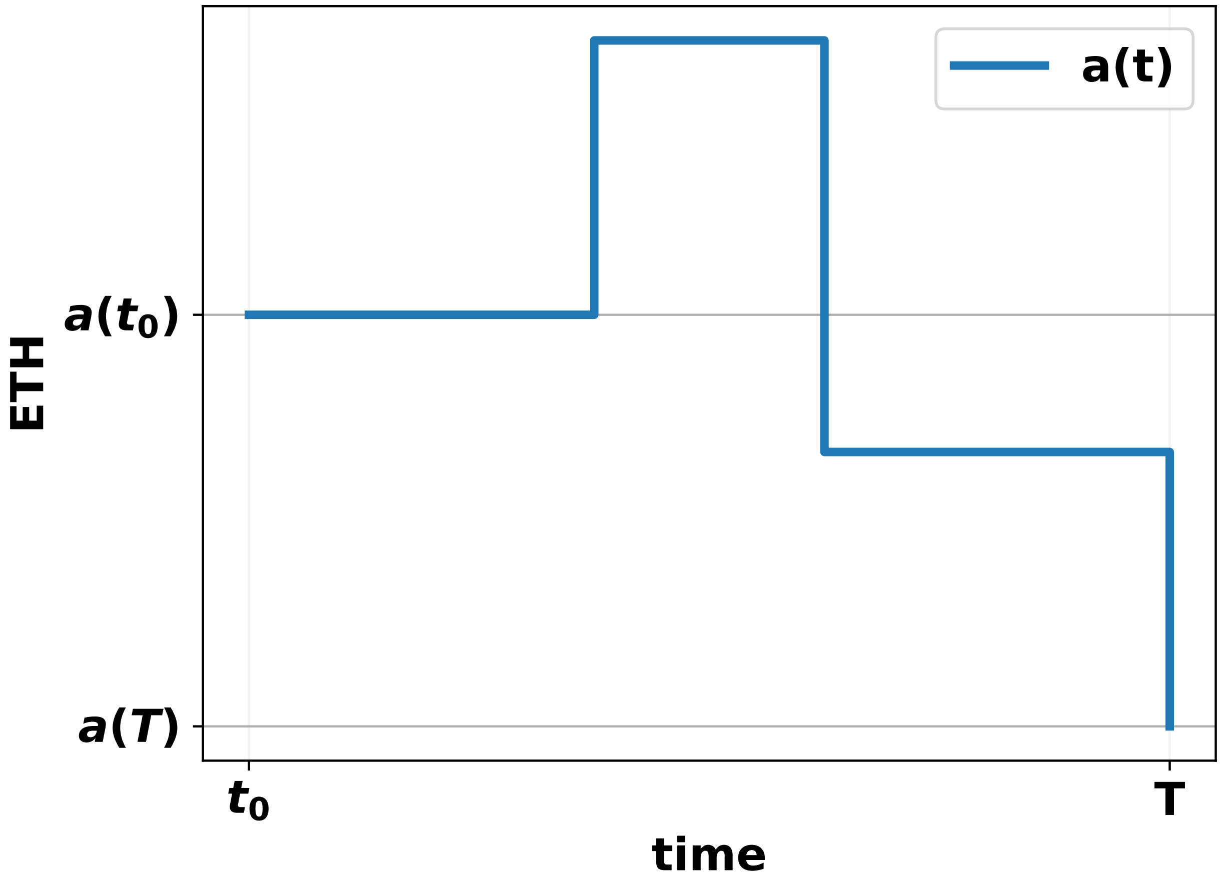

Figure 3.1: The amount of collateralization assets $a(t)$ as a function of time

Changes in the log-interest cause derivative breaks without function breaks.

The current collateralization ratio $r(t)$ for $d(t) > 0$ equals (an example is shown in Figure 3.3)

$$r(t) = \frac{e(t) \cdot a(t)}{d(t)}.$$

If $d(t) = 0$, we can set $r(t) = +\infty$. If $r(t)$ drops below $r_{\text{min}}(t)$ at any point in time, the platform triggers the liquidation. The collateralization requirement check normally is near real-time. And the borrower is responsible for paying the interest during the liquidation period.

3.1.2 Loss Given Default

Loss given default (LGD) refers to the portion of an asset that is lost in the event of a borrower defaulting [6]. In the Maker protocol, debts are typically over-collateralized, resulting in losses for users in most cases. We can represent a user's balance at time t as

$$\texttt{Bal}(t) = a(t) \cdot e(t) - d(t), \tag{3.2}$$

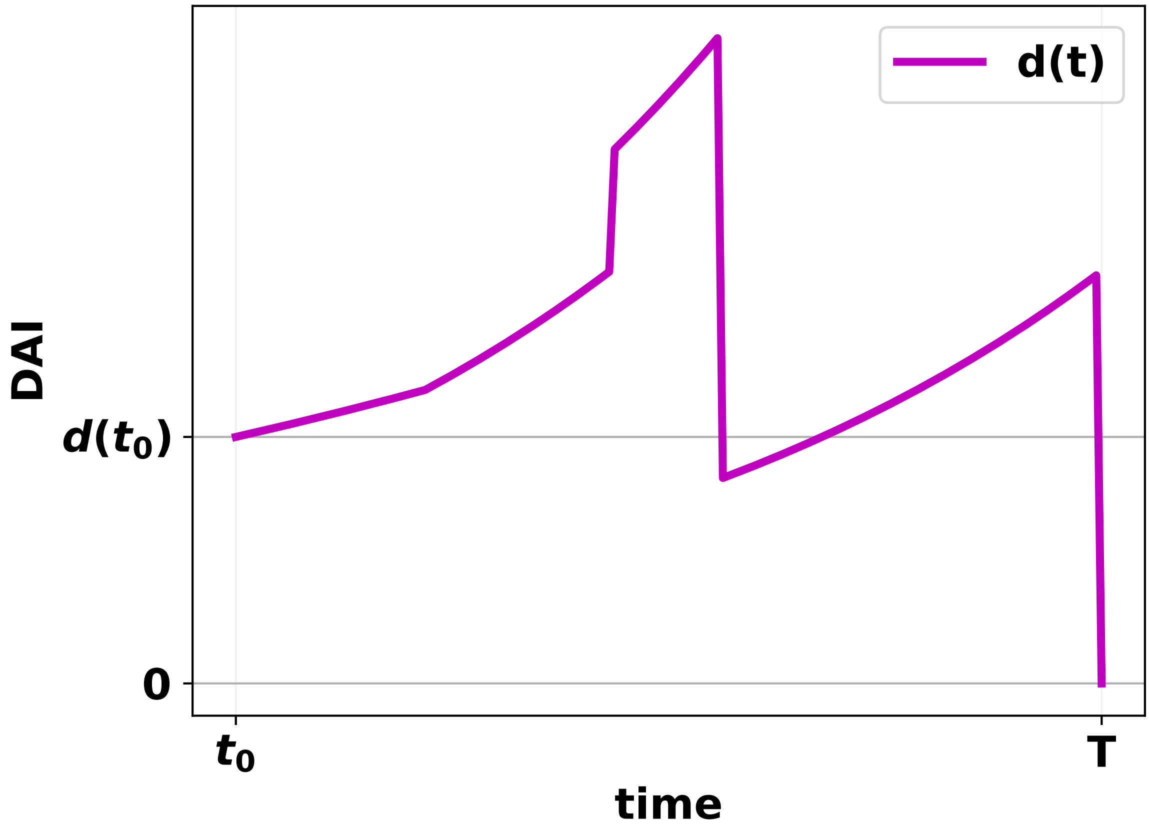

Figure 3.2: The debt $d(t)$ as a function of time

and the left-side limit $\varphi(t-) = \lim_{\tau \uparrow t} \varphi(\tau)$ for any given function φ. To calculate LGD for a user's collateral liquidation at time t, we use the following formula

$$\mathtt{LGD}(t) = \frac{\mathtt{Bal}(t-) - \mathtt{Bal}(t)}{\mathtt{d}(t-)}. \tag{3.3}$$

A positive value for LGD indicates a loss for the user, while a negative value indicates a gain. To determine the average of D user defaults at times $t_1, \ldots, t_D$, we use a weighted average:

$$\begin{aligned} \overline{\mathtt{LGD}} &= \frac{\sum_{d=1}^{D} d(t_d-) \cdot \mathtt{LGD}(t_d)}{\sum_{d=1}^{D} d(t_d-)} \\ &= \frac{\sum_{d=1}^{D} \mathtt{Bal}(t_d-) - \mathtt{Bal}(t_d)}{\sum_{d=1}^{D} d(t_d-)}. \end{aligned} \tag{3.4}$$

3.1.3 Annual Equivalent Rate

The interest rate on the platform changes over time and is charged in second-wise intervals. If a loan is liquidated, the loss of collateral value during liquidation is calculated as $(a(T) - a(T-)) \cdot e(T)$. The log-equivalent rate (LER) is a constant log-interest rate that results in the same final debt, including any liquidation losses, for a debt from t_0 to T with debt changes $\Delta d_0, \ldots, \Delta t_N$ at times $t_0, \ldots, t_N$ respectively.

To find the LER, we use the cumulative debt at time T with LER $= x$, denoted by $h(x)$,

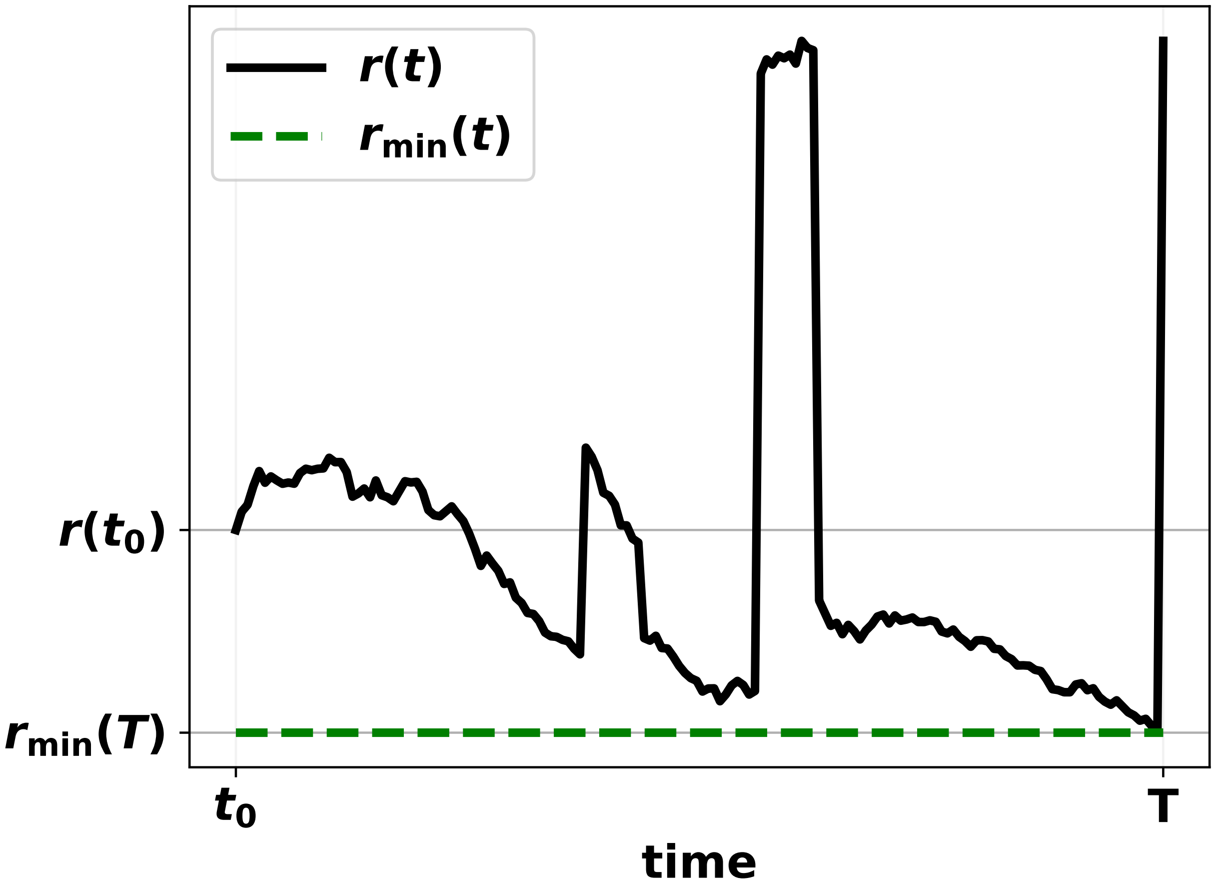

Figure 3.3: The collateralization ratio $r(t)$ and the minimum allowed collateralization ratio $r_{\text{min}}(t)$ as functions of time

which is calculated as

$$h(x) = \sum_{n=1}^{N} \Delta d_n \cdot \exp\left(x(T - t_n)\right). \tag{3.5}$$

The LER is then determined by solving the following equation:

$$h(x) = d(T) + (a(T) - a(T-)) \cdot e(T), \tag{3.6}$$

where $h(x)$ equals the final debt plus any liquidation losses. The function $h(x)$ is monotonically increasing for $x > 0$ since $d(t) > 0$ for $t \in (t_0, T)$. Therefore, if there is no default, the LER falls within the range of $[\min_{t\in(t_0,T)} f(t), \max_{t\in(t_0,T)} f(t)]$. However, if there is a default, the LER can be large and the solution of (3.6) may be unstable. To avoid this issue, we only consider values of LER that are less than or equal to a fixed constant $f_{\text{max}} > 0$.

To determine the average of D users at time t, we use a weighted average:

$$\overline{\texttt{LER}} = \frac{\sum_{i=1}^{D} d_i(t) \cdot \texttt{LER}_i}{\sum_{i=1}^{D} d_i(t)}. \tag{3.7}$$

3.1.4 Probability of Default

The probability of default (PD) is a risk assessment parameter commonly used by financial institutions. It is a financial term that describes the likelihood of default over a particular time horizon. Let us consider a dataset of loans from Maker platform in the format of time intervals as follows:

- N intervals of the time till default for default debts: $t_1, \ldots, t_N$
- M intervals of the time during which there were no defaults: $\tau_{N+1}, \ldots, \tau_{N+M}$. These intervals correspond to either active debts or returned debts. τs are intervals from the debt opening until the dataset generation for the active debts. τs are intervals from the debt taking until debt repayment for the returned debts

and consider different debt models.

Poisson Model

The classic finance baseline model is a Poisson model that assumes all debts are independent and have an exponential distribution with an unknown parameter $\lambda > 0$ for time until default [38]. This simplification allows for the estimation of λ [41, 37]. However, this assumption does not hold for Maker data since all debts are based on the same collateral type but with different collateralized ratios.

Statement 1. Let λ be the parameter of the exponential distribution. Let $X_1, \ldots, X_{N+M}$ be independent and identically-distributed (i.i.d.) random variable from the exponential distribution with a parameter λ. Let $x_1, \ldots, x_{N+M}$ be realizations of $X_1, \ldots, X_{N+M}$. Given $x_1, \ldots, x_N$ and deterministic parameters $y_{N+1}, \ldots, y_{N+M}$ such that $\forall n = N+1, \ldots, N+M : x_n > y_N$, the maximum likelihood estimator (MLE) $\hat{\lambda}$ of the parameter λ is

$$\hat{\lambda} = \frac{N+M}{\sum_{n=1}^{N} x_n + \sum_{m=N+1}^{N+M} y_m}. \tag{3.8}$$

Proof. The likelihood defines as

$$\begin{aligned} L_{N,M}(\lambda) &= L(\lambda, x_{N+1}, \ldots, x_{N+M} | x_1, \ldots, x_N, y_{N+1}, \ldots, y_{N+M}) \\ &= \prod_{n=1}^{N+M} \lambda \cdot \exp(-\lambda x_n) \cdot \prod_{n=N+1}^{N+M} I(x_n \geq y_n), \end{aligned} \tag{3.9}$$

where $I(A)$ is an indicator function of the event A, i.e., $I(A) = 1$ if A is true and $I(A) = 0$ if A in not true. The likelihood is non-negative, and $L_{N,M}(\lambda)$ equals 0 if any $x_n < y_n, n = N+1, \ldots, N+M$. So the maximum of $L_{N,M}(\lambda)$ is for $x_{N+1}, \ldots, x_{N+M} : x_n \geq y_n, n = N+1, \ldots, N+M$.

$L_{N,M}(\lambda)$ is a decreasing function of $x_n, n = N+1, \ldots, N+M$ for all λ in the region $x_n \geq y_n, n = N+1, \ldots, N+M$. So the maximum of $L_{N,M}$ is at $x_n = y_n, n = N+1, \ldots, N+M$, and the maximization of (3.9) is equivalent to the classic problem for the exponential distribution

$$\begin{aligned} L_{N+M}(\lambda) &= L(\lambda | x_1, \ldots, x_N, y_{N+1}, \ldots, y_{N+M}) \\ &= \prod_{n=1}^{N} \lambda \cdot \exp(-\lambda x_n) \cdot \prod_{n=N+1}^{N+M} \lambda \cdot \exp(-\lambda y_n) \to \max_{\lambda}, \end{aligned} \tag{3.10}$$

with the log-likelihood expressed as

$$l_{N+M}(\lambda) = (N+M)\ln\lambda - \lambda \cdot \left(\sum_{n=1}^{N} x_n + \sum_{n=N+1}^{N+M} y_n\right).$$

The observation that MLE $\hat{\lambda}$ of λ is given by (3.8) finishes the proof.

The probability of the default for a single debt during T,where X is an exponential random variable with parameter λ can be written as

$$PD(T) = P(X < T) = 1 - \exp(-\lambda T). \tag{3.11}$$

As the likelihood is functional equivariant [41], the MLE for PD is

$$\hat{PD}(T) = 1 - \exp(-\hat{\lambda} T), \tag{3.12}$$

where $\hat{\lambda}$ is given by (3.8). It is important to note that the model assumes independence between debts, so the covariance between different users is zero.

Brownian Motion Model

Another model considers the minimal allowed collateralization ratio $r_{\text{min}}(t)$ in comparison to the actual user's collateralization $r(t)$. We assume that the logarithm of $e(t)/e_0$ follows a Brownian motion with zero mean and an unknown standard deviation $\sigma > 0$. Therefore, $\frac{1}{\sigma}(\ln \frac{e(t)}{e_0})$ is a Brownian motion $\mathtt{B}_t$ with zero mean and unit variance.

Let us denote

$$\psi(x) = \int_0^T \frac{|x + fs|}{\sqrt{2\pi s^3}} e^{-\frac{(|x+fs|)^2}{2s}} ds \tag{3.13}$$

for fixed parameters f and T.

Theorem 1. If

1. the normalized exchange rate $\frac{1}{\sigma}(\ln \frac{e(t)}{e_0})$ for a given constant $\sigma > 0$ is a Brownian motion $\mathtt{B}_t$ with zero mean and unit variance
2. the borrower has a debt d_0 and collateral a_0 at time $t = 0$
3. the borrower has no actions with debt and collateral during $t \in (0, T]$
4. the platform's interest rate $f \geq 0$ and the minimum collateralization ratio $r_{\text{min}} > 0$ are constant,

then the probability of the borrower's default during the time interval $(0, T]$ and its variance are given by

$$PD = \psi(x_{\text{min}}) \tag{3.14}$$

and

$$\mathtt{var} = \psi(x_{\text{min}}) \cdot (1 - \psi(x_{\text{min}})), \tag{3.15}$$

respectively, where

$$x_{\text{min}} = \frac{1}{\sigma} \ln\left(\frac{d_0 \cdot r_{\text{min}}}{a_0 \cdot e_0}\right). \tag{3.16}$$

Proof Firstly, let the stability fee $f = 0$. Then $a(t) = a(0) \equiv a_0$ and $d(t) = d(0) \equiv d_0$. Therefore, a debt default is equivalent to the existence of such $t > 0$ that

$$e(t) = \frac{d_0}{a_0} \cdot r_{\text{min}} \equiv e_{\text{min}}. \tag{3.17}$$

A debt default occurs when the Brownian motion $\mathtt{B}_t$ reaches the level x_{min} (3.16). Let us enote $T_C = \inf\{t > 0 : \ \mathtt{B}_t = C\}$ for $C < 0$. Then for $T > 0$ [37] from the reflexion principle we have that

$$P(T_{x_{\text{min}}} < T) = \int_0^T \frac{|x_{\text{min}}|}{\sqrt{2\pi s^3}} e^{-\frac{x_{\text{min}}^2}{2s}} ds, \tag{3.18}$$

therefore the distribution density $p_C(t) = \frac{C}{\sqrt{2\pi t^3}} e^{-\frac{C^2}{2t}}$.

Now, let stability fee be nonzero constant $f \geq 0$. Then $d(t) = d_0 \cdot e^{ft}$ and a default is equivalent to the existence of such $t > 0$ that

$$e(t) = \frac{d_0 \cdot e^{ft}}{a_0} \cdot r_{\text{min}} \equiv e_{\text{min}}(t).$$

I.e., a default is the passage of level

$$x_{\text{min}}(t) = \frac{1}{\sigma} \ln \left(\frac{d_0 \cdot r_{\text{min}}}{a_0 \cdot e_0} \right) + ft = x_{\text{min}} + ft \tag{3.19}$$

by Brownian motion $\mathtt{B}_t$ (see Figure 3.4).
Let

$$T_{C,f} = \inf\{t > 0 : \ \mathtt{B}_t = C + ft\}. \tag{3.20}$$

Then for $C < 0$ and $f > 0$

$$P(T_{x_{\text{min}},f} < T) = \int_0^T \frac{|x_{\text{min}} + fs|}{\sqrt{2\pi s^3}} e^{-\frac{(x_{\text{min}}+fs)^2}{2s}} ds. \tag{3.21}$$

And from (3.13): $PD = \psi(x_{\text{min}})$. As the default is a Bernoulli random variable, its variance is given by (3.15) and theorem statement follows. **End of proof.**

Theorem 2. If, in addition to the assumptions 1) - 4) of Theorem 1,

5) the second borrower has a debt $\tilde{d}_0$ a collateral $\tilde{a}_0$ at time $t = 0$,

then the covariance of two borrowers' defaults during time interval $(0, T]$ equals

$$\mathtt{cov} = \psi(\min\{x_{\text{min}}, y_{\text{min}}\}) \cdot (1 - \psi(\max\{x_{\text{min}}, y_{\text{min}}\}))$$

where

$$y_{\text{min}} = \frac{1}{\sigma} \ln \left(\frac{\tilde{d}_0 \cdot r_{\text{min}}}{\tilde{a}_0 \cdot e_0} \right). \tag{3.22}$$

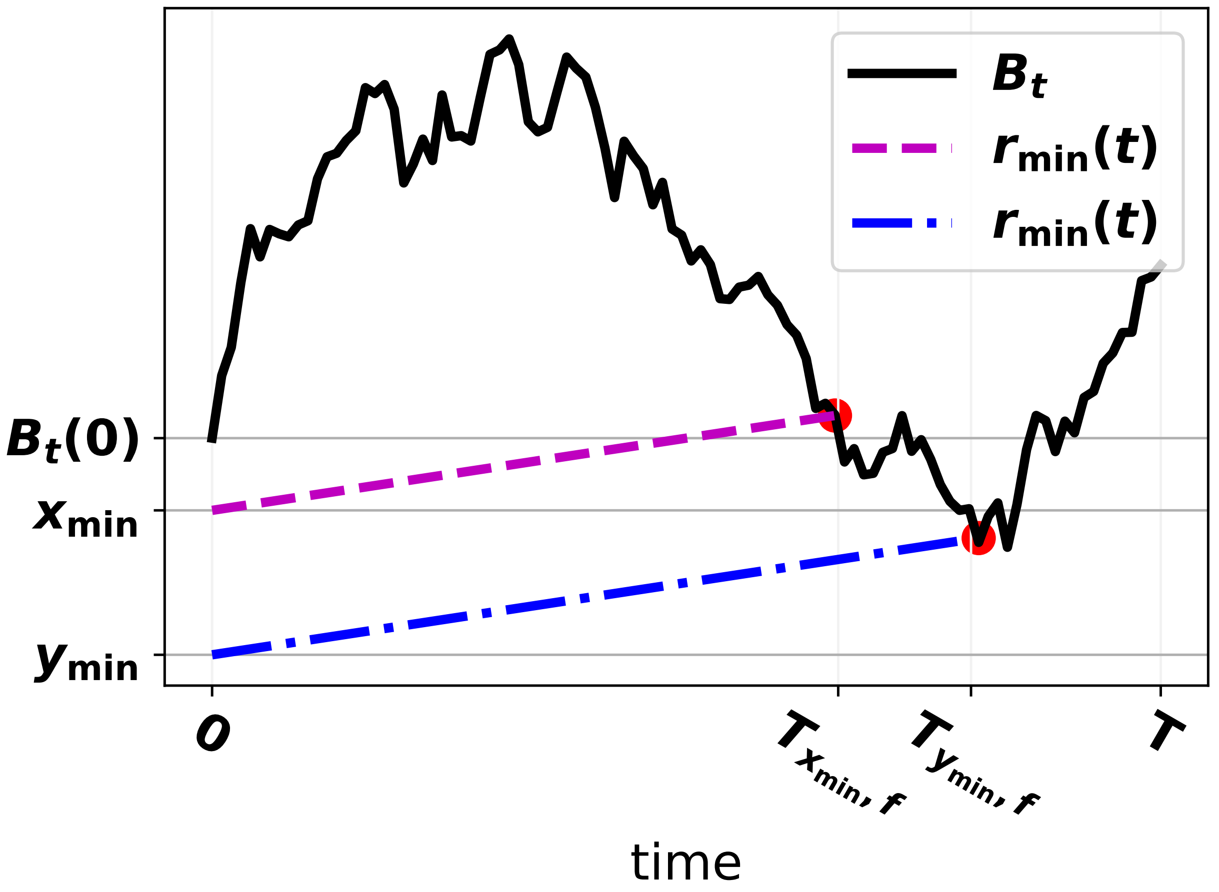

Figure 3.4: Borrower default can be described as a Brownian motion level passage. The black solid curve represents the normalized log-exchange rate of the collateral. The magenta dashed line indicates the minimum allowed rate before default for a user starting from $x_{\min}$. Similarly, the blue dash-dotted line represents the minimum allowed rate before default for a user starting from $y_{\min}$

Proof. The second user has a single asset as a collateral with a default at level $y_{\min}(t) = y_{\min} + ft$, where $y_{\min}$ is given by (3.22). Without loss of generality, $y_{\min} \leq x_{\min}$. Then the probability of the passage of both $x_{\min}$ and $y_{\min}$ is

$$
\begin{aligned}
&P(T_{x_{\min},f} < T; T_{y_{\min},f} < T) \\
&= P(T_{y_{\min},f} < T) = \psi(y_{\min}).
\end{aligned}
\tag{3.23}
$$

Denote $I(A)$ the indicator of the event $T_{x_{\min},f} < T$, i.e., $I(A) = 1$, if $T_{x_{\min},f} < T$, and $I(A) = 0$, if $T_{x_{\min},f} \geq T$. Denote $I(B)$ the indicator of the event $T_{y_{\min},f} < T$. The mathematical expectations and the covariance can be written in (3.24) and (3.25), in turn.

$$
\begin{aligned}
EI(A) &= P(T_{x_{\min},f} < T) = \psi(x_{\min}) \\
EI(B) &= P(T_{y_{\min},f} < T) = \psi(y_{\min})
\end{aligned}
\tag{3.24}
$$

$$
\begin{aligned}
\mathtt{cov}(I(A), I(B)) &= EI(A)I(B) - EI(A)EI(B) \\
= 1 \cdot P(T_{x_{\min},f} < T; T_{y_{\min},f} < T) &- EI(A) \cdot EI(B) \\
&= \psi(y_{\min}) - \psi(x_{\min}) \cdot \psi(y_{\min}) \\
&= \psi(y_{\min}) \cdot (1 - \psi(x_{\min})) \\
= \psi(\min\{x_{\min}, y_{\min}\}) \cdot (1 &- \psi(\max\{x_{\min}, y_{\min}\})).
\end{aligned}
$$

The observation that (3.25) coincides with (3.22) finishes the proof.

As the standard deviation of the Brownian motion $\ln\left(\frac{e(t)}{e_0}\right)$ is unobservable, we can estimate it. More precisely, given a sample $\{(t_n, e(t_n)\}_{n=0}^{N}$, where $t_0 < t_1 < \cdots < t_N$ the maximum likelihood estimator is

$$\hat{\sigma} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\ln\left(\frac{e(t_n)}{e(t_{n-1})}\right)\right)^2 / (t_n - t_{n-1}).} \tag{3.25}$$

Models Comparison

Both the Poisson process and Brownian motion provide models for predicting probability of default PD. However, the true model cannot be observed directly. To test the accuracy of these models, we generated a dataset of daily defaults and utilized both models to predict defaults one day in advance. Initially, the models assume that the data is stationary. To verify this hypothesis, we employed the Augmented Dickey-Fuller ADF test [12]. The ADF test is based on the autoregressive model and assesses unit roots in time series, which causes trends. The null hypothesis of the ADF test is the presence of a unit root in a time series. If the p-value (the probability that the null hypothesis is true) is less than a given significance level (usually 0.05), then the null hypothesis is rejected, indicating that the time series is stationary.

To compare the fit of models to data, we use four quantities [38, 41, 31]:

- **Kullback-Leibler divergence (KL)** measures the divergence between two probability distributions. It quantifies the amount of information lost when one distribution is used to approximate another.
- **Total Variation (TV)** measures the divergence between two probability distributions. It is defined as half the sum of the absolute differences between the corresponding probabilities in the two distributions.
- **Relative Root Mean Squared Error ($RMSE$)** and Relative Mean Absolute Error ($RMAE$) are quadratic and linear aggregations of point-wise distances between two datasets, respectively. In our specific case, $RMAE$ is equivalent to TV, but we retain both terms since they are commonly used by machine learners and statisticians.

For each quantity, a smaller value indicates a better fit of the theoretical model to the empirical data.

Chapter 4
Numerical experiments

4.1 Dataset

MakerDAO is the decentralized lending platform on the Ethereum blockchain technology. DAI tokens are generated when an investor deposit the collateral assets such as ETH via smart contract. We make use of the publicly available data of the smart contract on the Ethereum mainchain from November 11$^{\text{th}}$, 2019 to July 31$^{\text{th}}$, 2023. The 3^{rd} party tool such as Infura provide APIs to access data from Ethereum network in a structured. We focus only ETH-collateralized Vault, which is sufficiently volatile collateral to trigger liquidation and the most popular collateral asset in MakerDAO protocol Figure 4.1.

We use the abilities of the widely available Google Cloud platform to acquire access to the MakerDAO dataset. We use SQL instructions to connect to the Ethereum (ETH) smart contract and obtain the appropriate data. To retrieve the relevant information, this approach requires querying the contract's storage and events.

After successfully decoding the data, it is structured and organized to retain its useful context. The processed data is subsequently saved in a CSV (Comma-Separated Values) file format. CSV files are a popular technique for storing tabular data, allowing for easy exchange, analysis, and manipulation using a range of software applications.

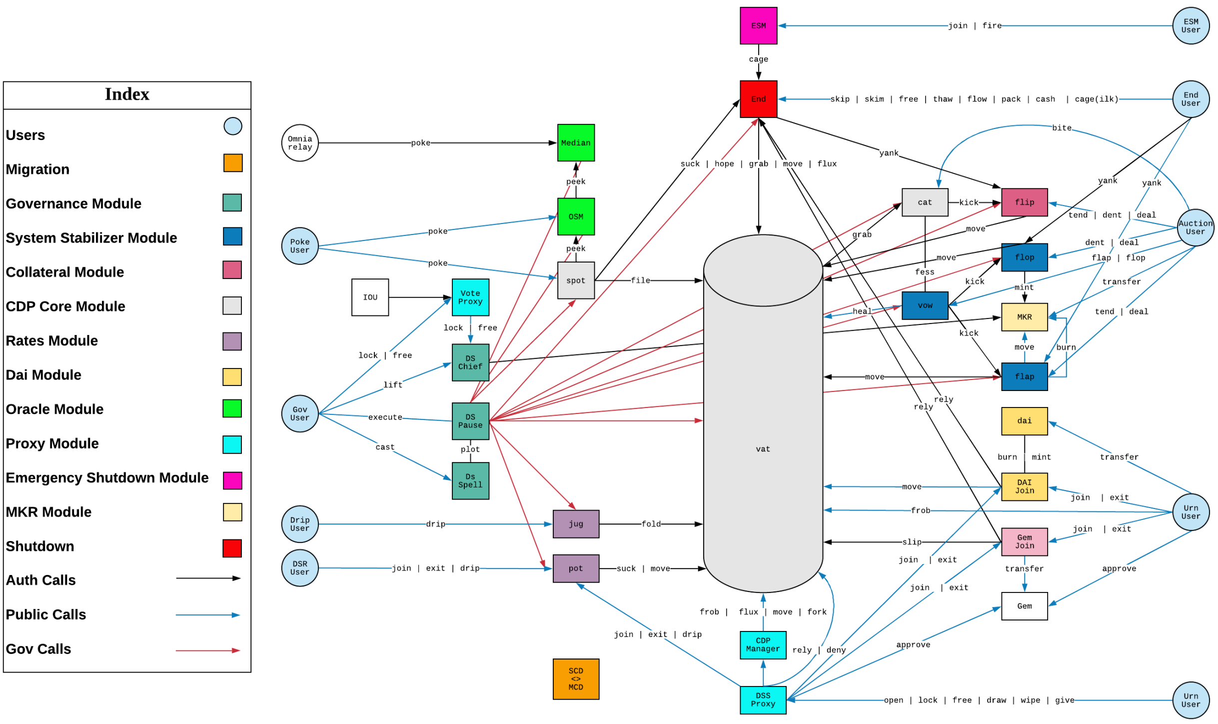

Figure 4.1: Data Structure on MakerDAO [26]

By taking those actions, we ensure that the MakerDAO dataset, which was sourced from the Ethereum blockchain via Google Cloud and then managed using SQL, Web3.0, and Python, is easily accessible and understandable for a variety of purposes ranging from investigation and evaluation to application development and beyond.

MakerDAO's smart contract details

In MakerDAO smart contract [24], we collect main data from VAT module. The VAT (Vaults) module is an important part of the MakerDAO, because this module holds the fundamental of the protocol and management of Vaults, which are smart contracts that allow users to lock up collateral and create DAI stablecoins. The VAT module is critical to the stability and security of the Maker Protocol.

- **Vault Creation:** Users start the Vault creation process by engaging with the VAT module's smart contract.
- **Collateral Management:** The collateral assets stored within each Vault. Users can use a variety of collateral, including ETH, BAT, USDC, and others.
- **DAI Generation:** When a user places collateral in a Vault, they can produce DAI stablecoins. The VAT module keeps track of how much DAI is earned against the locked collateral.
- **Collateralization Ratio:** For each Vault, the VAT module requires a specified Collateralization Ratio. When a Vault is undercollateralized, this ratio is used to assess whether it should be liquidated.
- **Liquidation:** The VAT module initiates the liquidation process if the value of the locked collateral in a Vault drops below the necessary Collateralization Ratio. The collateral is auctioned off to cover the balance of the DAI debt during the liquidation process.
- **Stability Fee:** Each Vault's stability charge is calculated and accumulated by the VAT module. The stability charge is the interest rate that Vault owners pay to generate DAI.

Functions	Descriptions	Arguments
frob	generate/return DAI, lock/unlock assets	i, u, v, w, dink, dart
fold	modify the debt multiplier, lock/unlock assets	i, u, rate
move	transfer stable coin between user	src, dst, rad
flux	transfer collateral between user	ilk, wad, src, dst
fork	splitting vault	ilk, src, dst, dart, dink
grab	liquidate a vault	i, u, v, w, dink, dart

Table 4.1: Function Descriptions and Arguments

4.1.1 Data Preparation

We begin by collecting data from public dataset platforms. Google Bigquery provides Ethereum data, which includes datasets from the MakerDAO platform. Smart contract data contains transaction code and input data in hexacode format. We can extract the function name and arguments after decoding with the Web3.0 library in Python, but the transactions cannot define user ids. In this work, we must first establish the usr id for each transaction before proceeding to define activities such as locking/unlocking assets, generating/paying back stable coins, and so on. The criteria

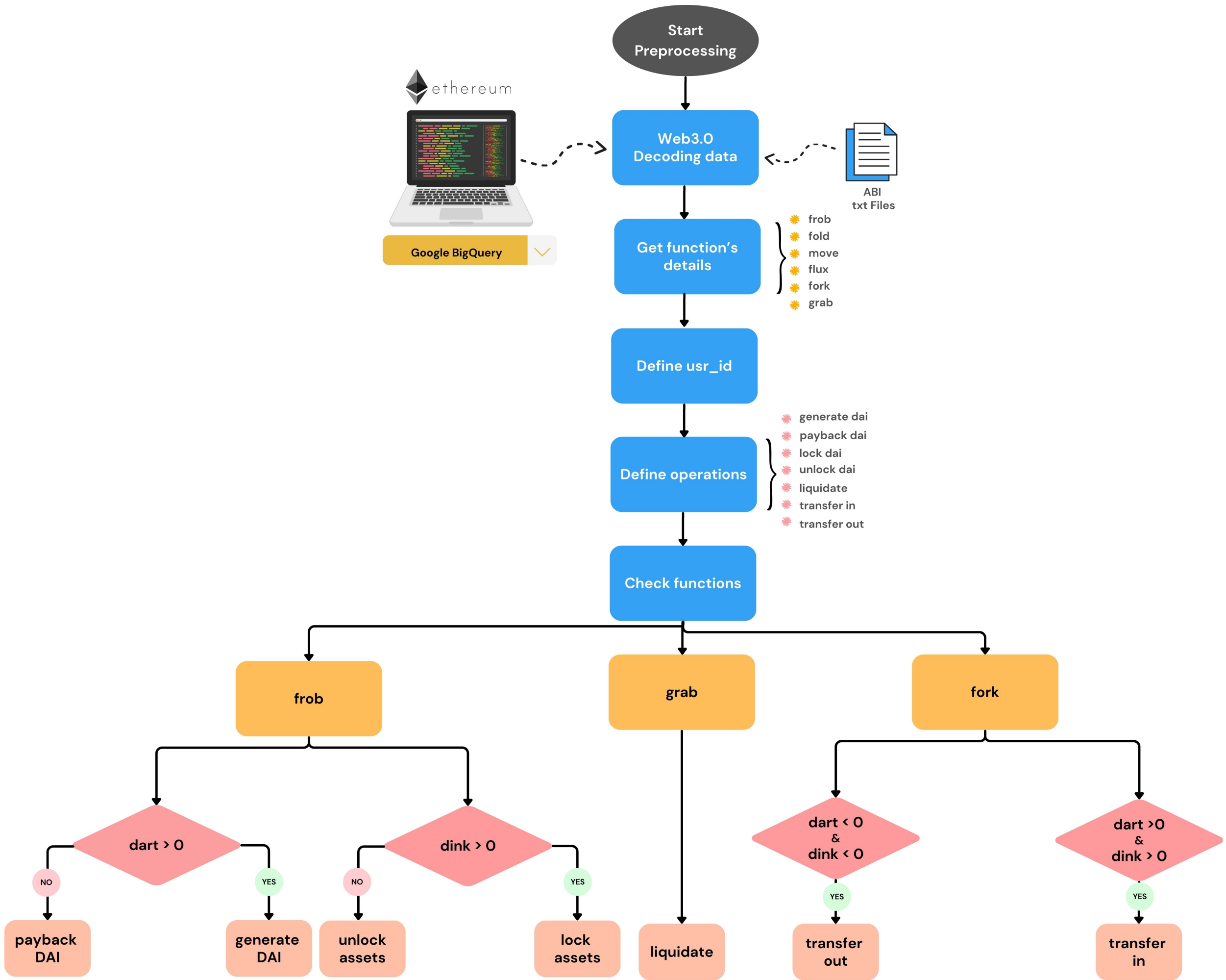

Figure 4.2: Work flow data pre-processing part1

can define by function frob grab and fork (see Table 4.1) and the process can be clarified by the workflow in Figure 4.2.

To compute debt in USD MakerDAO provides a rate to convert DAI to USD in the fold function from the VAT module; the amount must be calculated by the total value of each transaction. To convert asset value to USD, we may use Oracle. In this study, we focus on ETH-A, and we can acquire the price from MakerDAO chainlog (PIP_ETH) [28].

In terms of liquidation, we calculated collateral ratios following by platform [25] for each transaction using by formula 4.1 and liquidation ratio can be found in SPOT module by using SPOT ABI address [28].

$$\texttt{Collateral ratio} = \frac{\texttt{Lock collateral}}{\texttt{Issued dai}} \tag{4.1}$$

After the initial preprocessing, the table contains transactions by user, and a user can have more than 1 debt, thus for the next numerical experiment, we need to extract all debts from each user based on the user's status see processing data in Figure 4.3. In section 5, we present query code for obtaining data from MakerDAO based on this research and there is also code for data pre-processing available on GitHub Repository [20].

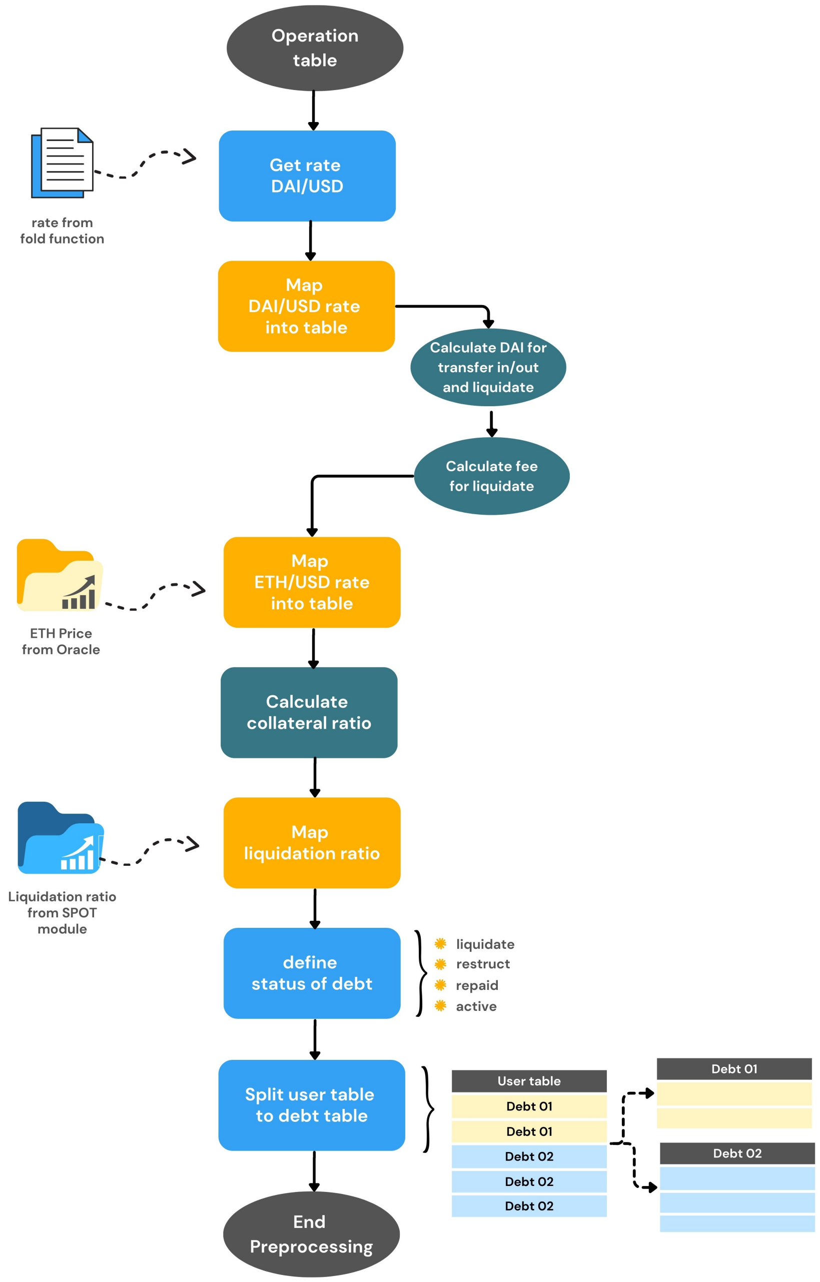

Figure 4.3: Work flow data pre-processing part2

4.1.2 Dataset Structure

MakerDAO accepted a variety of cryptocurrencies as collateral assets, various assets included USDC and USDT stablecoins, as well as various ERC-20 tokens, see from the Figure 4.4. The most popular and widely used of which was Ethereum (ETH). Since the platform's start in November of 2019 until now, 2023 MakerDAO has more than 40 different assets that users can use as collateral to earn Dai. However, Ethereum (ETH) is the most widely used asset for this purpose. In DeFi platforms, this is a prevalent scenario in which a significant amount of the collateral value is concentrated in one or a few major assets.

This research is focused on the `ETH`-collateralized risk program A debts (`ETH-A`) within the MakerDAO protocol deployed on the Ethereum network. This program has the largest number of debts (137,441 out of 259,048) and debt volume (13.4 billion `DAI` out of 36.9 billion `DAI`). We utilized publicly available data from November 11th, 2019 (the first debt start in the considered asset) to July 31st, 2023, accessed via the Big Query project by Google. The collected raw data

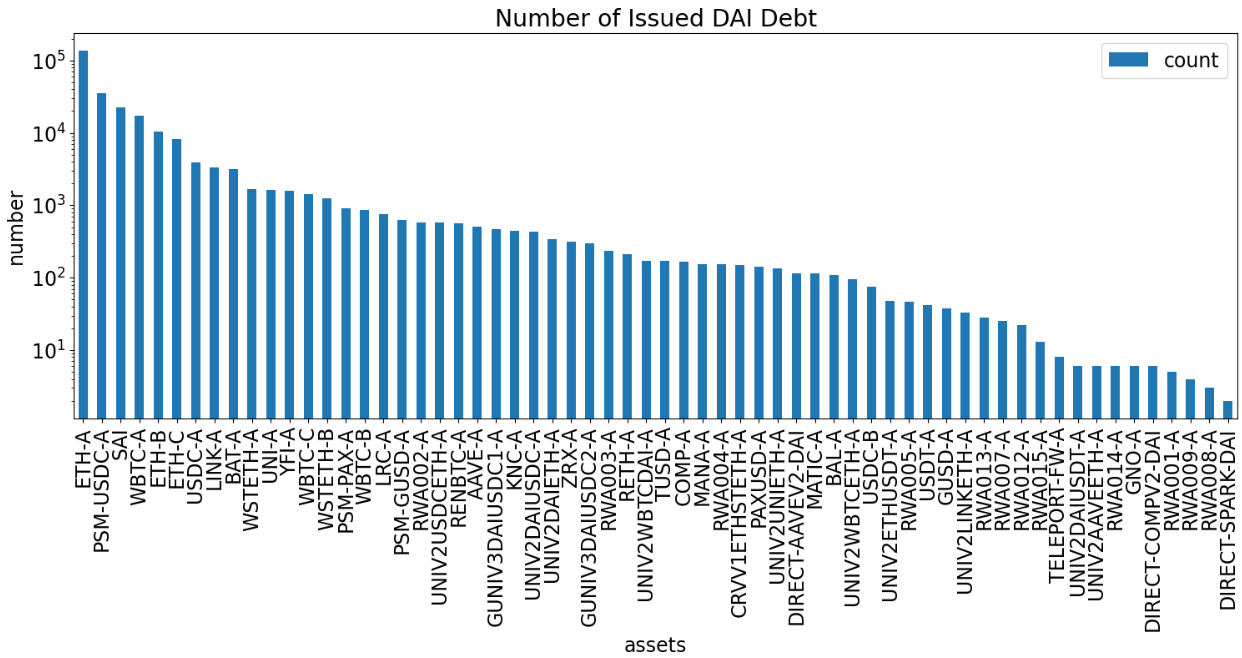

Figure 4.4: Number of debts issued for each assets

was decoded and further processed using Python. To verify specific information and ensure data correctness, we used a third-party API Ethereum provider, Infura.

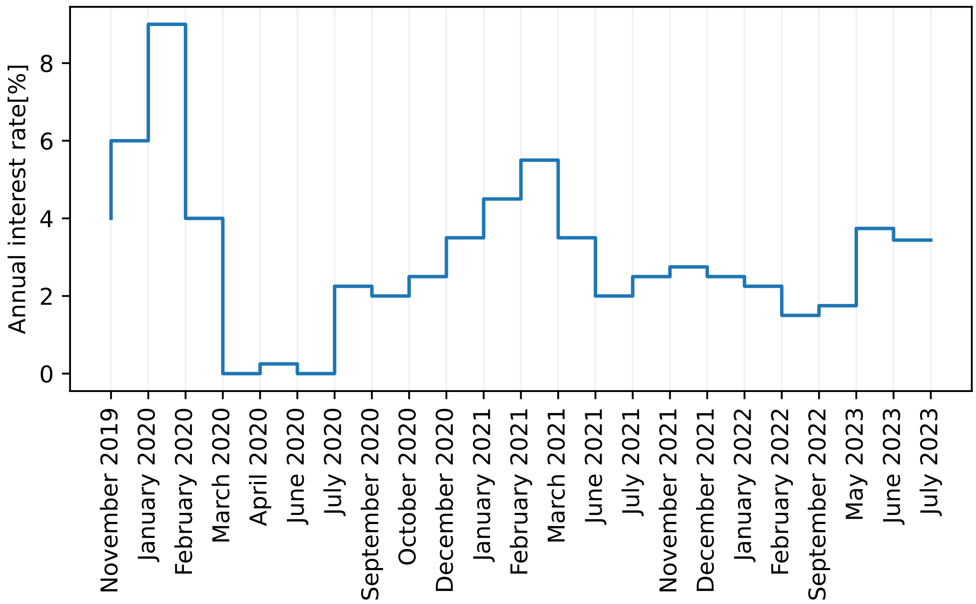

Figure 4.5: Annual Maker's interest rate for `ETH-A`

After processing all the internal terms such as `frob` and `fold` and focusing solely on the borrower-related aspect of the Maker protocol, we collected a loan portfolio dataset. The dataset comprises two parts: *system* and *borrower* data.

The *system* data contains common parameters, such as the `ETH/DAI` exchange rate $e(t)$, which is used to estimate the collaterization ratio since the `ETH-A` program deals with `ETH` as collateral. The exchange rate is provided by oracles and typically corresponds to the centralized exchanges rate [22]. Maker's loans are overcollateralized, and the minimum allowed collaterization ratio is defined as $r_{\min}(t)$. The platform receives interest rates from borrowers, and the dataset contains the log-interest $f(t)$.

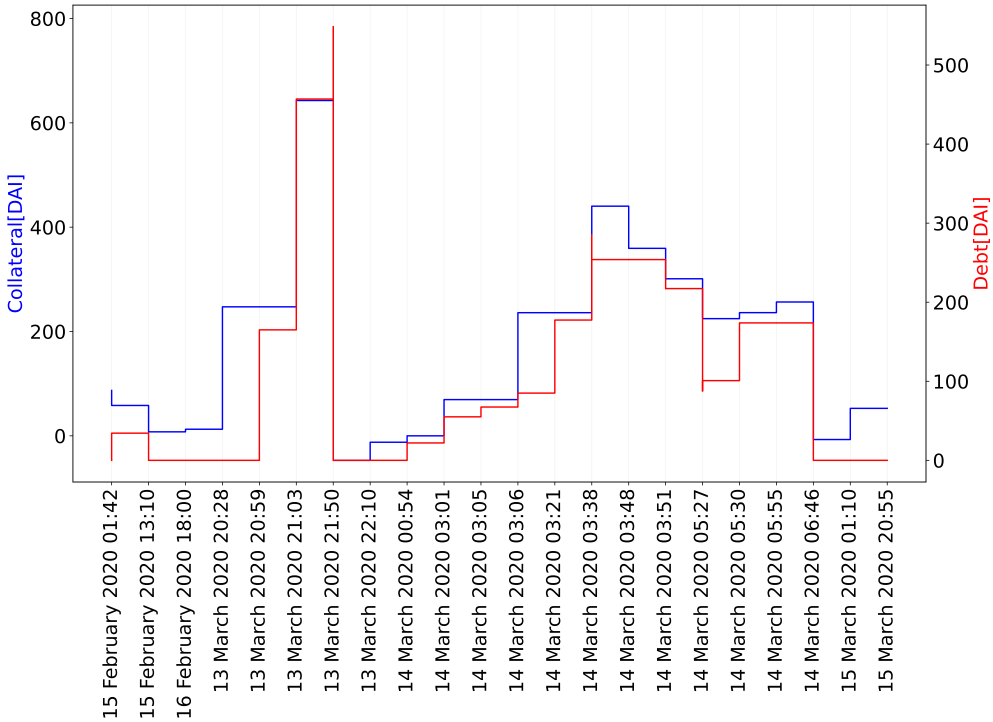

Figure 4.6: DAI debt over time for user 0x4032EE21404af045f6ba8022Cf4607950c87A39A

The *borrower* data contains borrowers' catalog and debt details. The borrower catalog lists all borrowers and their debts, each with start and end times, status, and loan actions (see Table 4.2). The possible statuses are

- **Repaid:** the borrower returned the debt
- **Liquidated:** the debt is fully repaid via liquidation process
- **Restructured:** the debt is partially repaid via liquidation process, and a new debt started immediately after the liquidation
- **Active:** the loan is active by the end of the observation period (July 31st, 2023).

The raw data only includes reference points for these parameters. Our utility functions allow obtaining their values over time and plotting system parameters for the entire period (see Figure 4.5) and loan characteristics for its lifetime (see Figures 4.6).

4.2 Results

The complete table can be presented in term of transaction table (see Table 4.2). This table displays all transactions for user id 0x931dBd7001D14112D17304B78d305c4FE317E571, which includes three debts (see Table 4.3). Before starting the numerical experiment, we calculate the Log equivalent rate (LER) and the Loss given default (LGD). We need to compute debt accumulate per day before we run an experiment about the probability of defaults using the Possion model and the Brownian motion model. The procedure depicted in the Figure 4.7

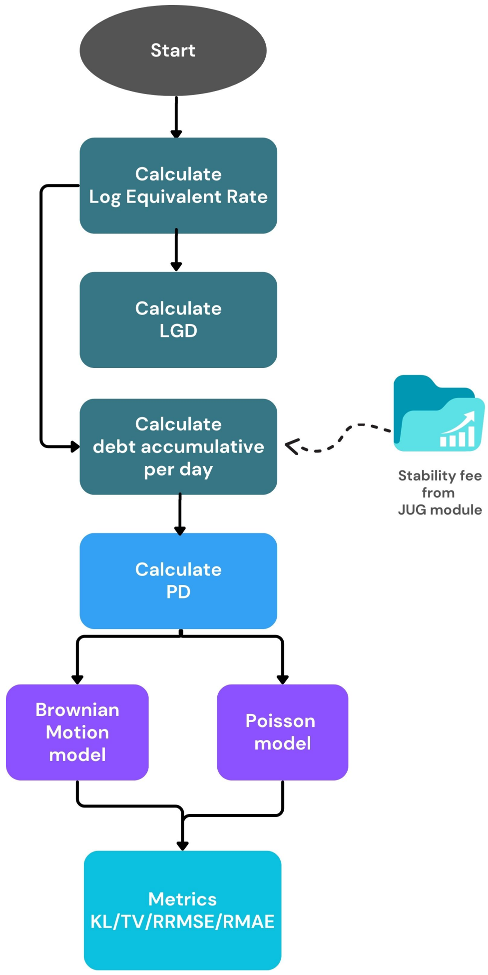

Figure 4.7: Processing MakerDAO dataset

4.2.1 Balance

The *borrower* data contains the details of all the debts. For our perposes we have to find the number of debts in Figure 4.8 and total collateral amount (total value locked, `TVL`) and debt in Figure 4.9, where the balance is their difference.

The number of borrowers experienced a remarkable surge initially, but came to a halt during the unfortunate event of Black Thursday in 2020. The debt amount peaked in mid-2021, only to witness a decline due to the cryptocurrency market downturn known as the crypto winter in the same year. Currently, the total debt in `ETH-A` is steadily decreasing towards zero as Maker DAO transitions into the Spark Lend Protocol, marking a significant development.

4.2.2 Loss Given Default

The information on debts in the liquidated and restructured statuses can be presented (see Table 4.3), including average loss given default values. A ”positive value for LGD” represents a loss for the user when a default occurs, whereas a ”negative value for LGD” indicate a gain. A positive results, on the other hand, indicates that the platform gains benefit, while a negative value indicates that the platform loses benefit.

The monthly average `LGD` (3.4) is depicted in Figure 4.10. Maker receives a fixed percentage of 13% from each auction (represented by the dashed black horizontal line), which indicates the typical loss level in an efficient market. The efficiency of the auction is a critical factor [16] and tends to decrease during periods of declining `ETH` prices.

Timestamp [yy-mm-dd hh-mm-ss]	Action	Collateral [ETH]	Debt [DAI]	Fee [DAI]	Debt [USD]	Rate [DAI/USD]	Rate [ETH/USD]	Collateral Rate	Liquidation Rate
20-02-15 01:42:39	lock asset	0.3000	0.0000	0.0000	0.0000	1.013186	286.474769	0.0000	1.5
20-02-15 01:42:39	generate dai	0.0000	34.3770	0.0000	33.9296	1.013186	286.474769	2.5000	1.5
20-02-15 13:10:55	unlock asset	-0.1000	0.0000	0.0000	0.0000	1.013288	284.858357	1.6573	1.5
20-02-16 18:00:01	liquidate	-0.2000	-34.3770	-0.0122	-33.9296	1.013545	249.382446	1.4509	1.5
20-03-13 20:28:37	lock asset	0.0404	0.0000	0.0000	0.0000	1.019139	121.323152	0.0000	1.5
20-03-13 20:59:40	lock asset	2.0000	0.0000	0.0000	0.0000	1.019144	121.323152	0.0000	1.5
20-03-13 21:03:20	generate dai	0.0000	165.0297	0.0000	161.9297	1.019144	121.323152	1.5000	1.5
20-03-13 21:03:20	lock asset	3.1989	0.0000	0.0000	0.0000	1.019144	121.323152	3.8517	1.5
20-03-13 21:03:20	generate dai	0.0000	245.1908	0.0000	240.5850	1.019144	121.323152	1.5495	1.5
20-03-13 21:50:37	generate dai	0.0000	46.7635	0.0000	45.8847	1.019151	130.835000	1.5000	1.5
20-03-13 21:50:37	lock asset	1.0593	0.0000	0.0000	0.0000	1.019151	130.835000	1.8033	1.5
20-03-13 21:50:37	generate dai	0.0000	91.3185	0.0000	89.6025	1.019151	130.835000	1.5030	1.5
20-03-13 22:10:47	liquidate	-6.2986	-548.3024	-0.0045	-538.0020	1.019154	129.463665	1.4872	1.5
20-03-14 00:54:08	lock asset	0.2500	0.0000	0.0000	0.0000	1.019178	136.220527	0.0000	1.5
20-03-14 03:01:40	generate dai	0.0000	22.0042	0.0000	21.5897	1.019197	132.025000	1.5000	1.5
20-03-14 03:01:40	lock asset	0.4161	0.0000	0.0000	0.0000	1.019197	132.025000	3.9966	1.5
20-03-14 03:01:40	generate dai	0.0000	33.0255	0.0000	32.4035	1.019197	132.025000	1.5981	1.5
20-03-14 03:05:13	lock asset	0.1000	0.0000	0.0000	0.0000	1.019198	132.025000	1.8380	1.5
20-03-14 03:06:46	generate dai	0.0000	12.3999	0.0000	12.1663	1.019198	132.025000	1.5000	1.5
20-03-14 03:06:46	lock asset	0.2265	0.0000	0.0000	0.0000	1.019198	132.025000	1.9435	1.5
20-03-14 03:06:46	generate dai	0.0000	17.6159	0.0000	17.2841	1.019198	132.025000	1.5409	1.5
20-03-14 03:21:27	lock asset	1.0000	0.0000	0.0000	0.0000	1.019201	133.573992	3.1296	1.5
20-03-14 03:38:30	generate dai	0.0000	92.3912	0.0000	90.6504	1.019203	133.573992	1.5000	1.5
20-03-14 03:38:30	lock asset	1.5000	0.0000	0.0000	0.0000	1.019203	133.573992	2.6292	1.5
20-03-14 03:38:30	generate dai	0.0000	107.7055	0.0000	105.6762	1.019203	133.573992	1.6361	1.5
20-03-14 03:48:09	unlock asset	-0.2760	0.0000	0.0000	0.0000	1.019204	133.573992	1.5068	1.5
20-03-14 03:48:09	payback dai	0.0000	-31.2139	0.0000	-30.6258	1.019204	133.573992	1.6920	1.5
20-03-14 03:51:13	unlock asset	-0.3179	0.0000	0.0000	0.0000	1.019204	133.573992	1.5248	1.5
20-03-14 03:51:13	payback dai	0.0000	-36.6915	0.0000	-36.0002	1.019204	133.573992	1.7823	1.5
20-03-14 05:27:38	unlock asset	-0.4331	0.0000	0.0000	0.0000	1.019220	132.160000	1.5000	1.5
20-03-14 05:27:38	payback dai	0.0000	-129.5872	0.0000	-127.1437	1.019220	132.160000	3.7177	1.5
20-03-14 05:27:38	unlock asset	-0.5674	0.0000	0.0000	0.0000	1.019220	132.160000	2.8621	1.5
20-03-14 05:30:09	generate dai	0.0000	13.1757	0.0000	12.9272	1.019220	132.160000	2.4881	1.5
20-03-14 05:30:09	lock asset	0.0854	0.0000	0.0000	0.0000	1.019220	132.160000	2.6001	1.5
20-03-14 05:55:06	generate dai	0.0000	73.0000	0.0000	71.6232	1.019223	132.160000	1.5081	1.5
20-03-14 06:46:34	lock asset	0.1500	0.0000	0.0000	0.0000	1.019231	133.595000	1.6398	1.5
20-03-15 01:10:55	liquidate	-2.1336	-173.8252	-0.0203	-170.5509	1.019317	121.240000	1.4881	1.5
20-03-15 20:55:55	lock asset	0.4748	0.0000	0.0000	0.0000	1.019407	123.468601	0.0000	1.5
20-03-15 23:01:50	unlock asset	-0.4748	0.0000	0.0000	0.0000	1.019417	122.476715	0.0000	1.5

Table 4.2: Loan actions of user `0x931dBd7001D14112D17304B78d305c4FE317E571`

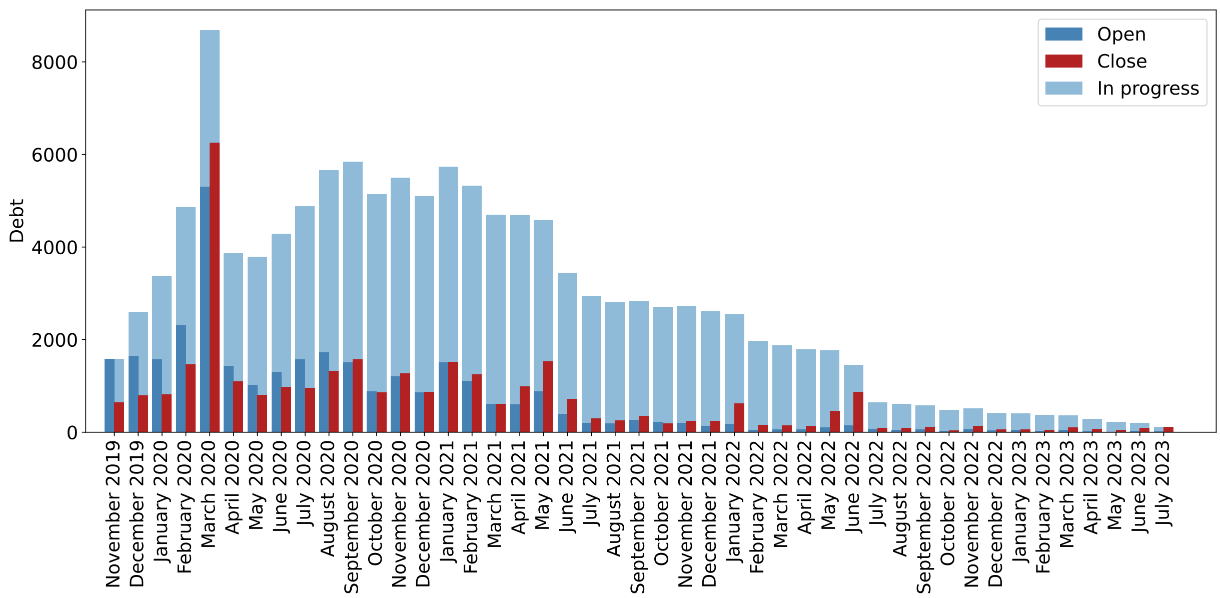

Figure 4.8: The number of `ETH-A` debts

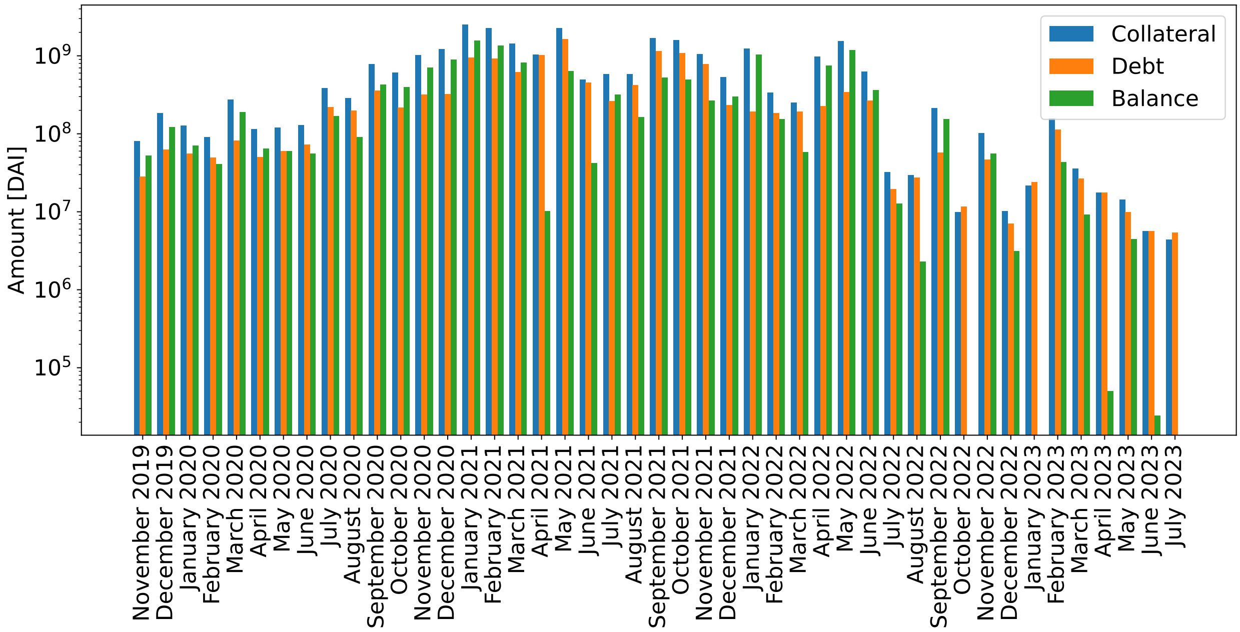

Figure 4.9: The total collateral amount and borrowers' balance

debt_number	start_date	end_date	status	avg LGD
01	2020-02-15 01:42:39	2020-02-16 18:00:01	liquidated	0.157793
02	2020-03-13 20:59:40	2020-03-13 22:10:47	restructured	0.428179
03	2020-03-14 03:01:40	2020-03-15 01:10:55	restructured	0.156983

Table 4.3: Debts of user id `0x931dBd7001D14112D17304B78d305c4FE317E571`

4.2.3 Annual Equivalent Rate

The monthly average `AER` (3.7) is shown in Figure 4.11. Although the `AER` for returned debts aligns with the annual Maker's interest rate for `ETH-A` (Figure 4.5), the majority of liquidated debts have an `AER` that exceeds 100%.

4.2.4 Probability of Default

Experimental proof of the Theorem 1

According to Brownian motion model described in Section 3.1.4, the probability of default for a given period of time can be estimated by Formula (3.14). We can prove this formula experimentally by simulating the Brownian motion $\mathtt{B}_t$ of the normalized exchange rate $\frac{1}{\sigma}(\ln \frac{e(t)}{e_0})$ (see Figure 3.4) multiple times and check if it has an intersection with $x_{\min}(t)$ from Formula (3.19). The experimental setup will be the following:

1. For each given $x_{\min}$ (3.16) and each given stability fee f compute $x_{\min}(t)$ values for time period $t \in (0, T)$. Let $D = 0$ be the counter of defaults.

2. For i in 1, ..., N:

   (a) Simulate random process $\mathtt{B}_t$.

   (b) If $\mathtt{B}_t$ intersects with $x_{\min}(t)$, update counter of defaults $D \to D + 1$, else do nothing.

3. Compute the empirical probability of default as $PD_{\text{Empirical}} = \frac{D}{N}$.

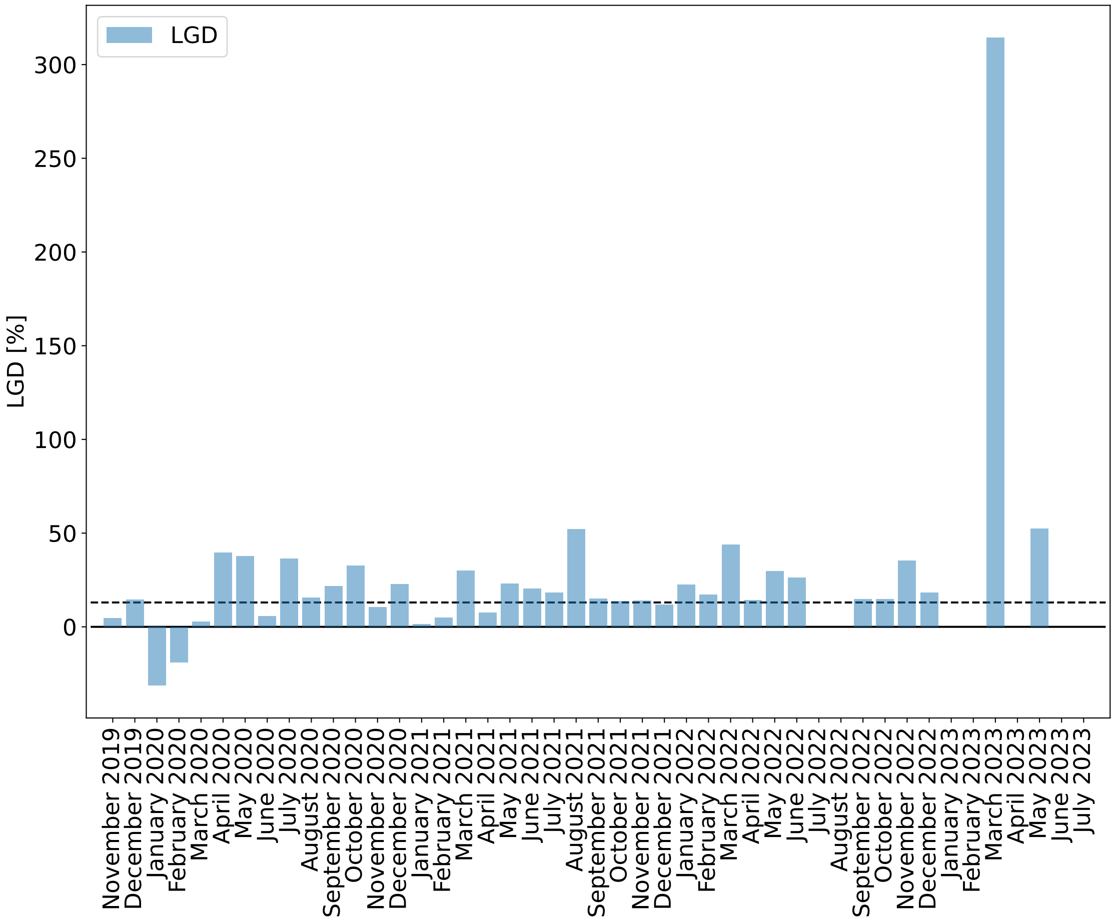

Figure 4.10: Monthly average LGD

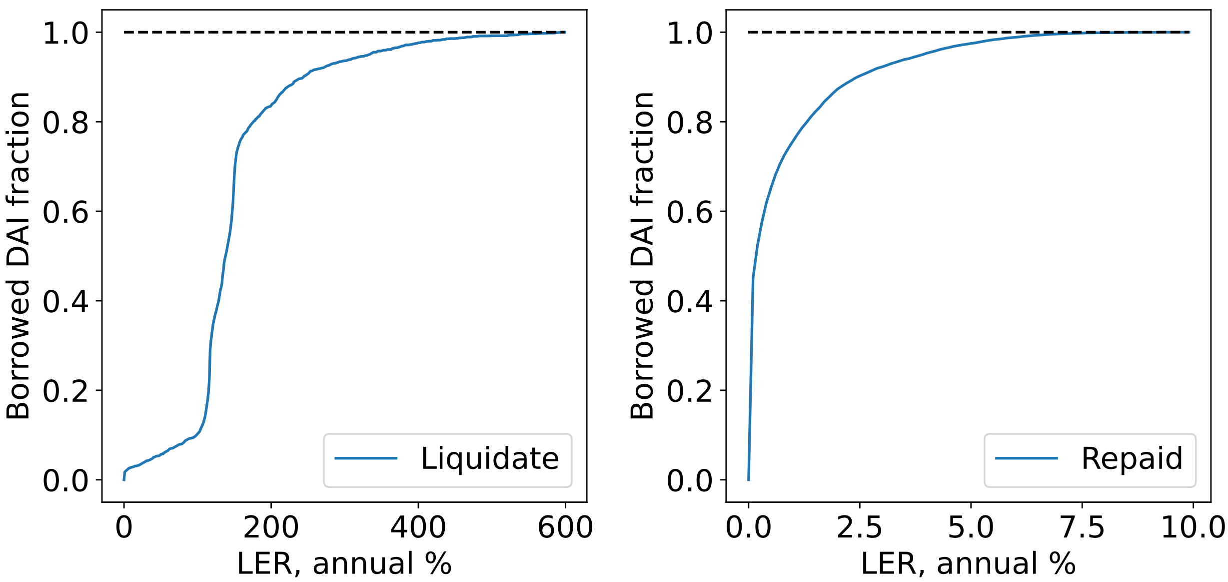

Figure 4.11: Monthly average AER for liquidated (left) and returned (right) debts

4. Compare the empirical probability of default $PD_{\text{Empirical}}$ with the theoretical probability of default $PD_{\text{Theoretical}}$ estimated by Formula (3.14).

The number of $\mathtt{B}_t$ sampling N should be reasonably big in order to take into account diverse scenarios of possible $\mathtt{B}_t$. We chose $N = 1000$ in our experiment. We used Quad method [30] for the numerical integration.

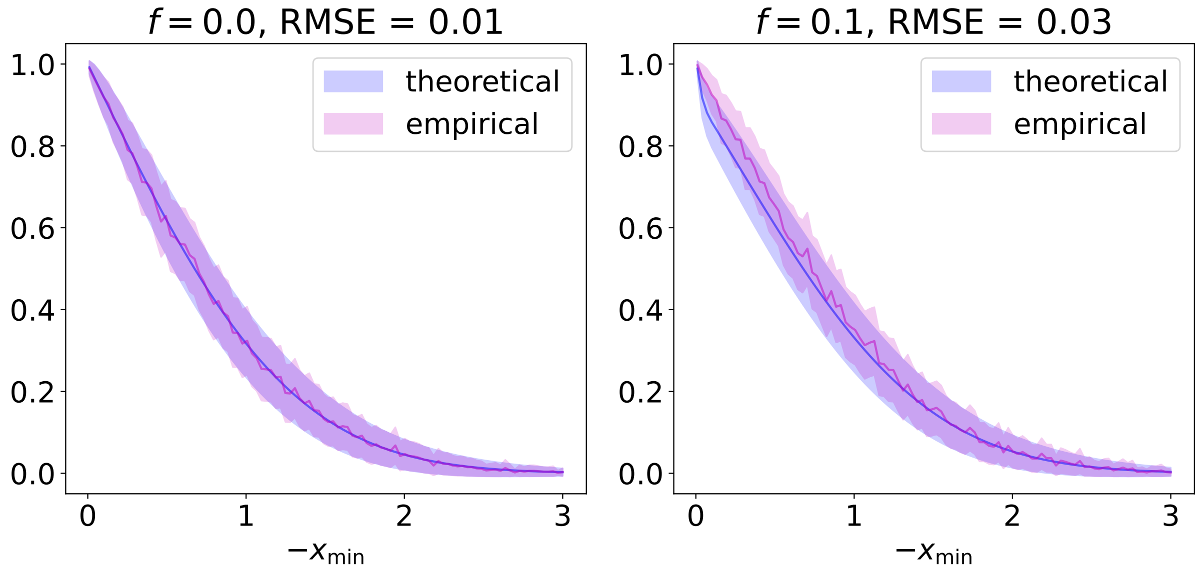

Figure 4.12: Empirical probability of default $PD_{\text{Empirical}}$ and theoretical probability of default $PD_{\text{Theoretical}}$ estimated by Formula (3.14) for various x_{min} and stability fee f.

After performing this experiment, we can see that theoretical and empirical probability of default match for $f = 0.0$ and $x_{\text{min}} \in (-3, 0)$ and that theoretical slightly underestimate empirical for $f = 0.1$. In practice, we will further use f values close to 0, so the Theorem 1 will apply with high accuracy.

Predictions by Brownian motion and Poisson model

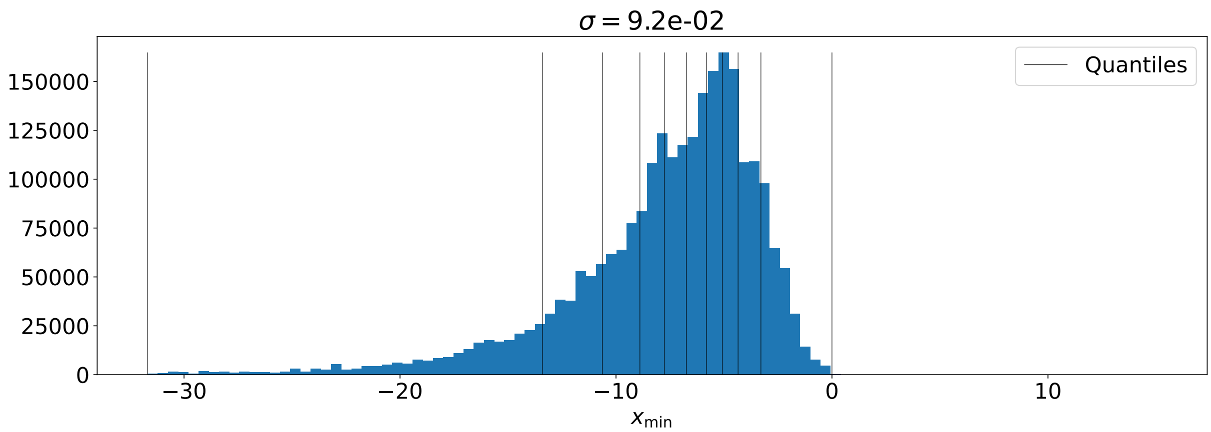

Figure 4.13: The distribution of x_{min} and x_{min} 10%-quantiles.

The number of daily defaults has an `ADF` test statistic of -8.89, which is significantly smaller than the 1% significance level of -3.45. This statistic enables us to reject the null hypothesis of data non-stationarity and conclude that the time series is stationary.

Empirical value of defaults was equal to 0 for each debt and for each day, except it was equal to 1 for the last day of debts that were actually liquidated. We used RRMSE, RMAE and TV metrics (see Section 3.1.4) to compare empirical value of defaults with the predicted by both Brownian motion and Poisson.

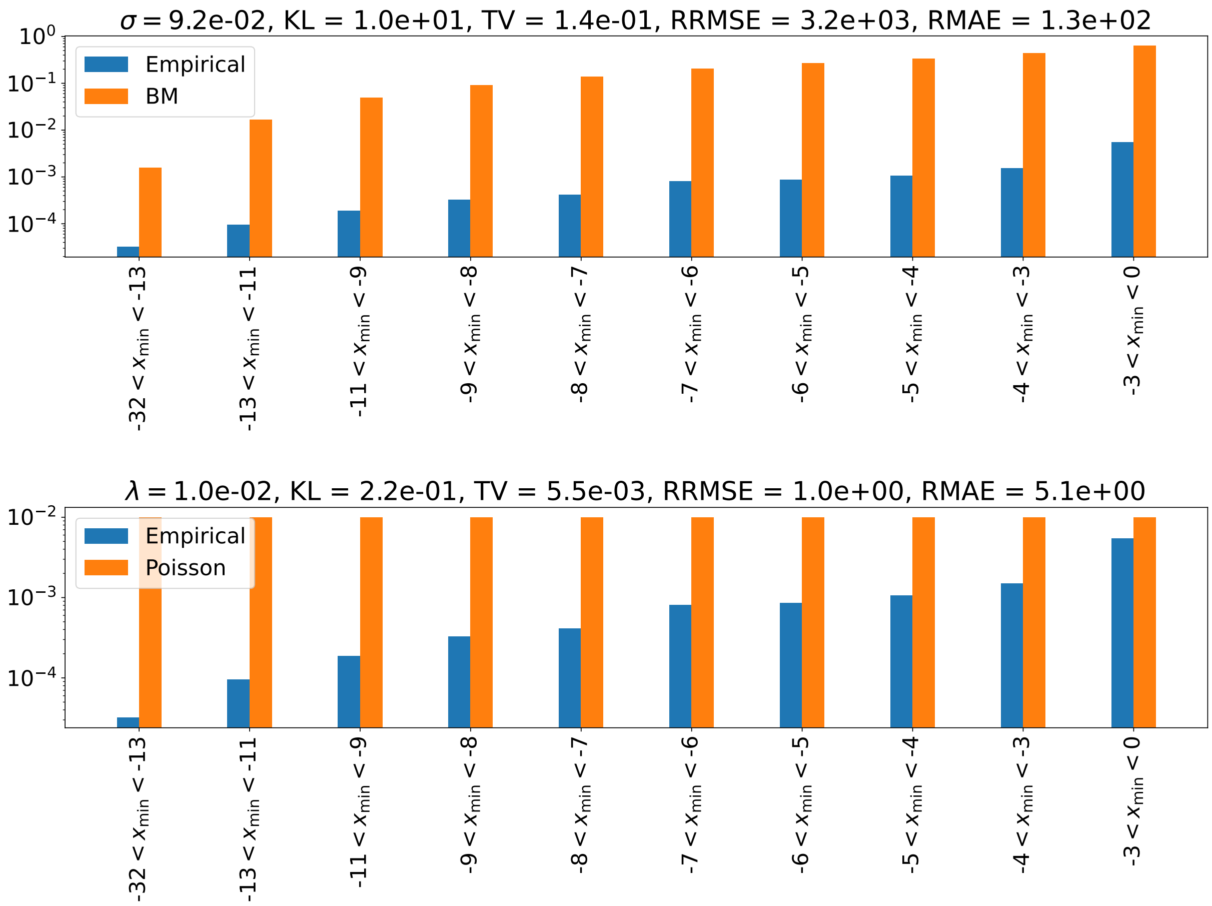

Figure 4.14: Mean Empirical probability of default $PD_{\text{Empirical}}$ and mean theoretical probability of default $PD_{\text{Theoretical}}$ estimated by Brownian motion model (top) and by Poisson model (bottom) for each 10%-quantile of x_{min}.

According to the Poisson model (see Section 3.1.4), probability of default for each day and for each debt is the same number estimated by Formula (3.12). If the time period is $T = 1$ day and estimated $\hat{\lambda} = 0.010$ day^{-1}, then this number is equal to $\hat{PD}(T) \approx T\hat{\lambda} = 0.010$.

We have computed a day-ahead actual number of defaults together with the predictions by Poisson and Brownian motion models. For each debt we studied the range of days since the second day of debt to the last day of it. For each given day we calculated x_{min} at the beginning of the day 3.16 using collateral and debt values at that moment. Standard deviation estimated by Formula (3.25) was equal to $\hat{\sigma} = 0.092$.

To calculate KL, we split all calculated x_{min} values into 10%-quantiles (see Figure 4.13). Then we estimated mean probability of default, both Empirical and predicted by Brownian motion (see Figure 4.14, top) and Poisson models (see Figure 4.14, bottom). Then we compared the given mean probabilities by KL metric.

It is seen that both Poisson model and Brownian motion model overestimate mean theoretical probabilities $PD_{\text{Theoretical}}$ for each quantile. However, estimated $\hat{\lambda}$ and $\hat{\sigma}$ are only approximations and are not guaranteed to lead to perfect accuracy. From the Figure 4.14 we can see that it is possible to decrease the mean predicted probabilities of defaults of both models by decreasing values of $\hat{\lambda}$ and $\hat{\sigma}$, which may possibly lead to the improvement of KL metric value.

We can optimize them to find lowest KL metric possible for each model. The results of the optimization are presented in Figure 4.15. We can see that the optimal values of $\hat{\lambda}$ and $\hat{\sigma}$ are different from the previously given ones. We can also outline that Brownian motion model leads to less KL error than Poisson model.

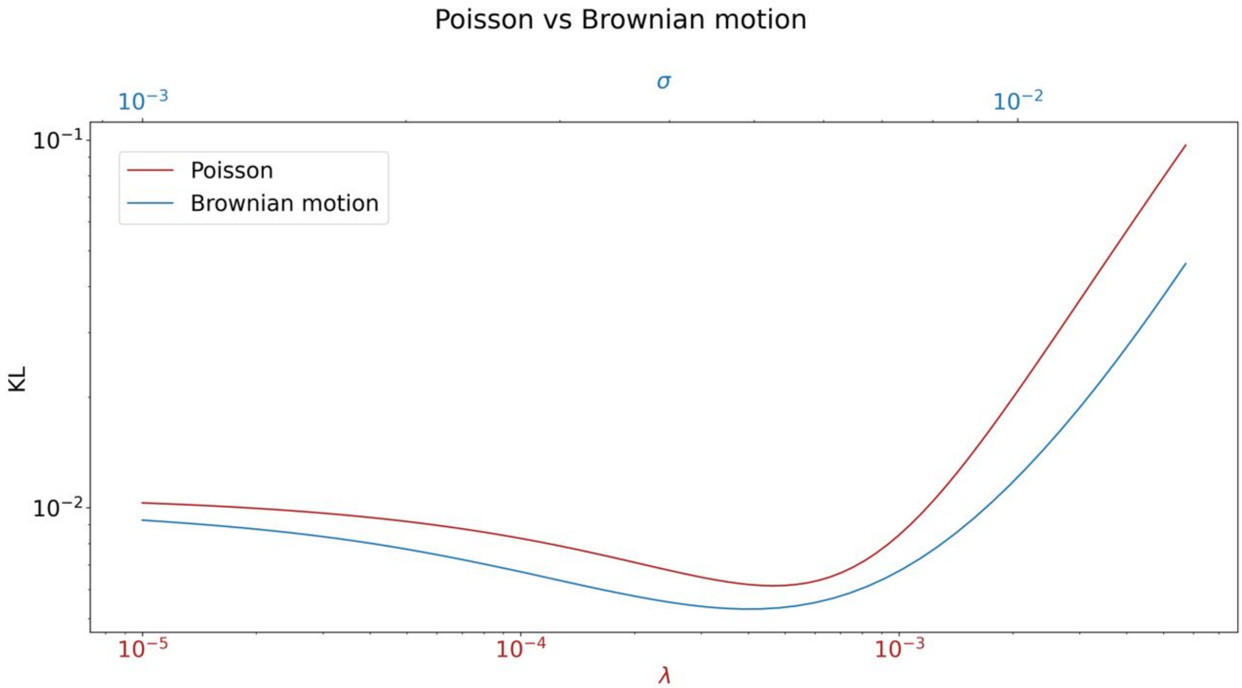

Figure 4.15: Poisson KL metric value vs $\hat{\lambda}$ value and Brownian motion KL metric value vs $\hat{\sigma}$ value. The optimal values are $\hat{\lambda}_{\text{opt}} = 4.6 \cdot 10^{-4}$ and $\hat{\sigma}_{\text{opt}} = 4.8 \cdot 10^{-3}$.

	KL	TV	MSE	MAE
Poisson process	0.0061	**0.00077**	**1.00**	**0.72**
Brownian motion	**0.0053**	0.00078	1.08	**0.72**

Table 4.4: Comparison of PD models. The lower the value in each column, the better the fit to the data. The best result in each column is highlighted.

As result, the Brownian motion model is found to be superior to the baseline Poisson process model in describing and predicting debt defaults (see Table 4.4). This conclusion is supported by the KL divergence, which measures the difference between the two probability distributions. The regression-specific MSE is worse for the Brownian motion model compared to the Poisson process model. However, the MAE and TV are comparable between the two models.

Both the MSE and MAE values are close to one, indicating that both models have poor predictive power from a regression perspective. This finding suggests that further analysis is needed.

Overall, the findings confirm the hypothesis that the Brownian motion model is better suited for capturing real-world data and demonstrates its superiority in fitting probabilistic distributions in collateral-based DeFi lending.

Chapter 5
Discussion and conclusion

This research focuses on analyzing the lending aspect of the Maker protocol in the DeFi space from a traditional finance perspective. The author have gathered a unique dataset comprising loan portfolios sourced from the MakerDAO project, making it the first dataset of its kind in the DeFi field. This publicly available dataset contains essential financial characteristics related to borrowing, including balance, loss given default, annual equivalent rate, and probability of default. The current version of the dataset covers only the most popular Maker's borrowing program called `ETH-A`. However, the author plan to expand the dataset to include other programs and new Spark Loan data in future work.

In addition to collecting this dataset, the author have developed a specialized mathematical model tailored specifically to the Maker project. This model allows them to estimate the probability of default by considering the presence of crypto-collateral and utilizing Brownian motion passage levels. The proposed model outperformed the Poisson process baseline model on the loan portfolio dataset. By incorporating borrowing-driven financial characteristics into the dataset and developing this model, the author provide a comprehensive understanding of both individual loan defaults and the correlation among different loans.

Expanding the analysis to include other borrowing programs beyond `ETH-A` presents challenges in finding the default correlation of level passage times between two correlated Brownian motions representing different collateral types. The author acknowledge this as a future work. However, this also opens up opportunities to estimate the platform's risk, where simultaneous defaults of a significant portion of borrowers could pose a threat.

The findings of this study offer valuable insights into lending practices in DeFi projects and help bridge the gap between traditional finance and blockchain-based financial services. This research contributes to the understanding of how DeFi lending operates and offers a standardized approach to analyzing and evaluating loan portfolios in the DeFi space. Furthermore, the methodology can be extended to other DeFi lending platforms such as Compound and Aave.

Acknowledgements

Sincere gratitude belong to my supervisor, Yury Yanovich, for his continuous dedication and support during this academic journey. Yury's constant commitment to advise me, his willingness to engage in boundless talks, and his constant efforts to assist address every difficulty that came during this project were critical to its success. Yury gave up important time and expertise to ensure that this research not only meet but exceeded its objectives.

I'd also like to thank all of my supporters, whose encouragement whether it was from friends, family, or coworkers, your encouragement encouraged me to persevere and push the boundaries of my knowledge. This endeavor would not have been possible without your unfailing faith in my talents.

Finally, I want to express my gratitude to Yury Yanovich and all of my supporters for their contributions to making this research a reality. Your efforts have left an unforgettable imprint on this project, and hence on my academic career. Thank you for your confidence in my abilities.

Appendix A: SQL Queries

Listing 1: Query data from VAT module

```
SELECT
    block_timestamp,
    block_number,
    transaction_index,
    trace_address,
    transaction_hash,
    input,
    ARRAY_TO_STRING(
        ARRAY(
            SELECT CHR(CAST(num AS INT64))
            FROM UNNEST(SPLIT(trace_address, ','))
            AS num
        ),
        ","
    ) as trace_addr_str
FROM `bigquery-public-data.crypto_ethereum.traces`
WHERE DATE(block_timestamp) >= "2019-11-12"
    and DATE(block_timestamp) <= "2023-07-31"
    and trace_address IS NOT NULL
    and call_type = "call"
    and to_address =
    LOWER("0x35D1b3F3D7966A1DFe207aa4514C12a259A0492B")
    and status = 1
ORDER BY
    CAST(block_number as INT64),
    CAST(transaction_index as INT64),
    trace_addr_str
```

Listing 2: Query ETHfrom PIP_ETH

```
SELECT
    block_timestamp,
    output
FROM `bigquery-public-data.crypto_ethereum.traces`
WHERE DATE(block_timestamp) >= "2019-11-12"
        and from_address =
        LOWER("0x65c79fcb50ca1594b025960e539ed7a9a6d434a3")
        and to_address =
        LOWER("0x81fe72b5a8d1a857d176c3e7d5bd2679a9b85763")
        and block_timestamp < '2023-08-01'
```

Listing 3: Query Liquidation Ratio from SPOT

```
SELECT
      block_timestamp,
      input
FROM    bigquery-public-data.crypto_ethereum.traces
WHERE   DATE(block_timestamp) >= "2019-11-12"
        and DATE(block_timestamp) < "2023-08-01"
        and trace_address IS NOT NULL
        and call_type = "call"
        and to_address =
        LOWER("0x65c79fcb50ca1594b025960e539ed7a9a6d434a3")
        and status = 1
```

Listing 4: Query Stabiity Fee from JUG

```
SELECT  block_timestamp, input
FROM    bigquery-public-data.crypto_ethereum.traces
WHERE   DATE(block_timestamp) >= "2019-11-01"
        and DATE(block_timestamp) < "2023-08-01"
        and trace_address IS NOT NULL
        and call_type = "call"
        and to_address =
        LOWER("0x19c0976f590D67707E62397C87829d896Dc0f1F1")
        and status = 1
```

Bibliography

[1] A. Molini, P. Talkner, G. K., and Porporato, A. First passage time statistics of brownian motion with purely time dependent drift and diffusion. Physica A: Statistical Mechanics and its Applications.

[2] AAVE. AAVE Protocol Whitepaper v1.0. *Github*, December (2020), 1–21.

[3] Alam, M. N., and Ali, M. M. Loan Default Risk Prediction Using Knowledge Graph. *KST 2022 - 2022 14th International Conference on Knowledge and Smart Technology* (2022), 34–39.

[4] Annisa, M., and Rusdah. Prediction of Non-Performing Loans for Credit Application Analysis of Rural Bank Using Random Forest. In *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)* (2022), vol. 2022-October.

[5] Azoulay, T., Carl, U., and Rottenstreich, O. Allowing Blockchain Loans with Low Collateral. In *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)* (5 2023), IEEE, pp. 1–9.

[6] Basel Committee on Banking Supervision. *The Basel Framework*. Bank for International Settlements, 2022.

[7] Bunz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., and Maxwell, G. Bulletproofs: Short Proofs for Confidential Transactions and More. In *2018 IEEE Symposium on Security and Privacy (SP)* (5 2018), vol. 2018-May, IEEE, pp. 315–334.

[8] Buterin, V. A next-generation smart contract and decentralized application platform. *white paper* (2014).

[9] Buterin, V. Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform. *Ethereum*, January (2014), 1–36.

[10] Buterin, V. On Public and Private Blockchains - Ethereum Blog, 2015.

[11] Chen, Y.-R., Leu, J.-S., Huang, S.-A., Wang, J.-T., and Takada, J.-I. Predicting Default Risk on Peer-to-Peer Lending Imbalanced Datasets. *IEEE Access 9* (2021), 73103–73109.

[12] Elliott, G., Rothenberg, T. J., and Stock, J. H. Efficient Tests for an Autoregressive Unit Root. *Econometrica 64*, 4 (7 1996), 813.

[13] Ermogenous, A. Brownian motion and its application in the stock market. *University of Dayton eCommons* (2005).

[14] Escamilla, J. A. L., and Yanovich, Y. Data Mining of Compound DeFi Project. *Proceedings of the 2022 5th International Conference on Blockchain Technology and Applications* (12 2022), 16–23.

[15] Feinstein, B. D., and Werbach, K. The Impact of Cryptocurrency Regulation on Trading Markets. *SSRN Electronic Journal* (8 2021).

[16] Gu, W. C., Raghuvanshi, A., and Boneh, D. Empirical Measurements on Pricing Oracles and Decentralized Governance for Stablecoins. *Cryptoeconomic Systems 1*, 2 (10 2021).

[17] Hayden, E., Porath, D., and Westernhagen, N. V. Does diversification improve the performance of German banks? Evidence from individual bank loan portfolios. *Journal of Financial Services Research 32*, 3 (2007).

[18] Hull, J. C. *A model of the behavior of stock prices*. Pearson, 2017.

[19] Islam, S. R., Eberle, W., and Ghafoor, S. K. Credit Default Mining Using Combined Machine Learning and Heuristic Approach. *Arxiv* (7 2018), 1–7.

[20] Kasemsuk, S. Data-mining-for-makerdao, 2023.

[21] Khairi, A., Bahri, B., and Artha, B. A Literature Review of Non-Performing Loan. *Journal of Business and Management Review 2*, 5 (2021).

[22] Kjaer, M., Di Angelo, M., and Salzer, G. Empirical Evaluation of MakerDAO's Resilience. In *2021 3rd Conference on Blockchain Research and Applications for Innovative Networks and Services, BRAINS 2021* (9 2021), IEEE, pp. 193–200.

[23] Leshner, R., and Hayes, G. Compound: The Money Market Protocol. *compound.finance* (2019), 1–8.

[24] Maker Foundation. Maker Protocal 101.

[25] MakerDAO. The Maker Protocol: MakerDAO's Multi-Collateral Dai (MCD) System, 2020.

[26] MakerDAO. Vat detailed documentation, 2023.

[27] Makri, V., Tsagkanos, A., and Bellas, A. Determinants of non-performing loans: The case of Eurozone. *Panoeconomicus 61*, 2 (2014).

[28] MetaMask. MakerDAO Deployment Registry.

[29] Meyer, E., Welpe, I. M., and Sandner, P. Decentralized Finance—A systematic literature review and research directions. In *ECIS 2022 Research Papers* (2021), Elsevier BV, pp. 1–25.

[30] Piessens, R., de Doncker-Kapenga, E., Überhuber, C. W., and Kahaner, D. K. *Quadpack: a subroutine package for automatic integration*, vol. 1. Springer Science & Business Media, 2012.

[31] Rasmussen, C. E. Gaussian Processes in Machine Learning. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3176. Springer Verlag, 2004, pp. 63–71.

[32] Rossi, S. P., Schwaiger, M. S., and Winkler, G. How loan portfolio diversification affects risk, efficiency and capitalization: A managerial behavior model for Austrian banks. *Journal of Banking and Finance 33*, 12 (2009).

[33] Rossikhin, V., Burdin, M., and Mykhalskyi, O. Legal Regulation Issues of Cryptocurrency Circulation in Ukraine. *Baltic Journal of Economic Studies 4*, 3 (10 2018), 254–258.

[34] Schär, F. Decentralized Finance: On Blockchain- and Smart Contract-based Financial Markets. *SSRN Electronic Journal* (2020).

[35] Schar, F. Decentralized Finance: On Blockchain- and Smart Contract-Based Financial Markets. *Review 103*, 2 (2021), 153–174.

[36] Serengil, S. I., Imece, S., Tosun, U. G., Buyukbas, E. B., and Koroglu, B. A Comparative Study of Machine Learning Approaches for Non Performing Loan Prediction. In *Proceedings - 6th International Conference on Computer Science and Engineering, UBMK 2021* (2021).

[37] Shiryaev, A. N. *Essentials of stochastic finance*. World Scientific, 1999.

[38] Shiryayev, A. N. *Probability*. Graduate Texts in Mathematics 95. Springer-Verlag, New York, 1984.

[39] Vogelsteller, F., and Buterin, V. EIP-20: ERC-20 Token Standard, 2015.

[40] Voigt, P., and von dem Bussche, A. *The EU General Data Protection Regulation (GDPR)*. Springer International Publishing, Cham, 2017.

[41] Wasserman, L. *All of Statistics*. Springer Texts in Statistics. Springer, New York, NY, 2004.

[42] Zetzsche, D. A., Arner, D. W., and Buckley, R. P. Decentralized Finance (DeFi). *SSRN Electronic Journal* (2020).

[43] Zhu, L., Wu, Y., Gai, K., and Choo, K. K. R. Controllable and trustworthy blockchain-based cloud data management. *Future Generation Computer Systems 91* (2 2019), 527–535.