

**Cryptocurrency Liquidity Prediction for Market Stability**

High Level Design

Domain: Machine Learning

Creator: Somes Dash

Date: 22.05.2025

Orissa, INDIA

+91 8895319373

[Somes.dash1995@gmail.com](mailto:Somes.dash1995@gmail.com)

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date issued** | **Version** | **Description** | **Author** |
| May 22, 2025 | 1.1 | First Draft | Somes Dash |
|  |  |  |  |
|  |  |  |  |

Contents

[Abstract 3](#_Toc110595993)

[Introduction 3](#_Toc110595994)

[What is High-Level Design Document? 3](#_Toc110595995)

[Scope 4](#_Toc110595996)

[General Description 4](#_Toc110595997)

[Definitions 4](#_Toc110595998)

[Product Description 4](#_Toc110595999)

[Problem Statement 4](#_Toc110596000)

[Proposed solution 5](#_Toc110596001)

[Further improvements 5](#_Toc110596002)

[Data requirements 5](#_Toc110596003)

[Tools used 5](#_Toc110596004)

[Hardware Requirements 7](#_Toc110596005)

[Constraints 7](#_Toc110596006)

[Assumptions 7](#_Toc110596007)

[Design Details 7](#_Toc110596008)

[Process Flow 7](#_Toc110596009)

[Event Log 8](#_Toc110596010)

[Error Handling 9](#_Toc110596011)

[Performance 9](#_Toc110596012)

[Reusability 9](#_Toc110596013)

[Application Compatibility 9](#_Toc110596014)

[Resource Utilization 9](#_Toc110596015)

[Dashboards 10](#_Toc110596016)

[KPIs (Key Performance Indicators) 10](#_Toc110596017)

[Conclusion 10](#_Toc110596018)

# Abstract

The cryptocurrency market is known for its rapid fluctuations and inherent volatility, where liquidity is a key factor influencing trading efficiency and market stability. A lack of liquidity can trigger sharp price movements and elevate risks for both traders and exchanges. This project focuses on building a machine learning-based model to predict liquidity levels in the crypto market by examining various indicators such as trading volume, market behavior, exchange availability, and social sentiment. Utilizing historical data, the model incorporates advanced algorithms like LSTM, Random Forest, and Decision Tree regressors to anticipate potential liquidity constraints. The predictions aim to support traders and exchange platforms in proactively managing risks and making well-informed decisions. This report outlines the complete development process, emphasizing the overall system structure and data pipeline used to construct the prediction framework.

# Introduction

## What is High-Level Design Document?

The objective of this High-Level Design (HLD) document is to offer a comprehensive architectural perspective on the Cryptocurrency Liquidity Prediction project, effectively connecting the project’s functional requirements with its technical execution. This document serves as a foundational guide for development, detailing the system architecture, key components, and data flow interactions. It is also intended to identify potential design issues or inconsistencies early in the development process.

Specifically, the HLD aims to:

* Clearly outline all major design elements of the liquidity prediction system
* Explain the structure of user interfaces and the mechanisms for data input and output
* Specify the necessary software and hardware interfaces to support system functionality
* Define performance benchmarks needed for accurate and timely liquidity predictions
* Illustrate the architecture of the predictive pipeline, including machine learning models and data processing stages
* Cover essential non-functional requirements such as security, reliability, scalability, maintainability, portability, and efficient resource usage to ensure a robust and high-performing system

## Scope

This High-Level Design (HLD) document presents the structure and architecture of the Cryptocurrency Liquidity Prediction system. It highlights essential components such as data sources, feature engineering processes, the architecture of the machine learning models, and the user interaction workflow. The document also describes the design of the data processing pipeline, model training and prediction procedures, and the dashboard used for displaying forecasts and trading insights.

Written in clear, accessible language, the HLD ensures that stakeholders, data scientists, developers, and system administrators can easily understand the system. It outlines the technology stack, data storage solutions, and system integration points, emphasizing how the design enables early detection of liquidity issues. This helps traders and exchange platforms manage risks and make more informed market decisions.

# General Description

## Definitions

## Product Description

## he Cryptocurrency Liquidity Predictor (CLP) is a machine learning-based regression model designed to predict liquidity levels in cryptocurrency markets. By analyzing various market factors such as trading volume, transaction patterns, exchange listings, and social media sentiment, the system provides early warnings of potential liquidity crises. This enables traders and exchange platforms to make informed decisions, manage risks effectively, and optimize trading strategies.

| **Term** | **Description** |
| --- | --- |
| **CLPP** | Cryptocurrency Liquidity Prediction Project |
| **Flask** | A lightweight Python web framework used to build the backend API for deployment |
| **Git** | A version control system used to track code changes and collaborate |
| **GitHub** | A code hosting platform used for pushing and storing the project repository |
| **Render** | A cloud-based platform used to deploy and host web applications |
| **VS Code** | Visual Studio Code, an IDE used for writing, editing, and pushing the code |
| **UI** | User Interface that displays model predictions to users |
| **API** | Application Programming Interface; allows communication between UI and model |
| **joblib** | Python library used for saving and loading trained machine learning models |
| **H5 File** | A file format used to save the LSTM deep learning model (e.g., lstm\_model.h5) |
| **Scikit-learn** | A machine learning library used to build traditional models like Random Forest |
| **TensorFlow** | A deep learning framework used to build and train the LSTM model |

## Problem Statement

## To develop a machine learning-based solution for predicting cryptocurrency liquidity levels by analyzing various market factors such as trading volume, transaction patterns, exchange listings, and social media activity. The goal is to provide early warnings of liquidity crises to assist traders and exchanges in making informed decisions and managing risks effectively.

## The aim of this project is to build and deploy a robust predictive model that can accurately forecast cryptocurrency liquidity levels in real-time, helping stakeholders detect potential liquidity shortages and optimize trading strategies.

## Proposed solution

## The project applies standard Data Science steps—data exploration, cleaning, feature engineering, model training, and testing—on crypto market data. It includes a user-friendly frontend for real-time liquidity level predictions based on input market factors.

## Further improvements

The liquidity prediction model can be integrated into trading platforms or crypto apps, providing users with real-time insights and risk alerts through an intuitive interface. Future work could include incorporating more data sources like social media trends and improving model accuracy with advanced algorithms.

## Data requirements

The data requirements are defined by the project's objective—predicting cryptocurrency liquidity levels using various market indicators. The dataset should ideally include the following features:

* **Trading Volume**: 24h volume of each coin/token.
* **Price Movements**: Historical prices including open, close, high, and low.
* **Market Capitalization**: Overall value of each coin/token in circulation.
* **Time-Based Features**: Date, day of the week, month to detect temporal patterns.

## These data points are essential for training models that can accurately detect patterns and anticipate potential liquidity crises.

## Tools used

Python was the primary programming language used to develop the cryptocurrency liquidity prediction model.

Key libraries and frameworks include:







* For visualization tasks, matplotlib, seaborn and plotly were used
* For visualization tasks, matplotlib, seaborn and plotly were used
* Anvil and Flask were used for building the web application and server to run the code
* Apache Cassandra was used to storage and retrieval of data
* GitHub is used as version control system
* NumPy and Pandas were used to clean and interpret data
* Scikit-learn was used to cross validate and compare different models
* CatBoost Regressor was used to build the final model

## Hardware Requirements

* Windows Server, Linux, or any operating system that can run as a webserver, capable of delivering HTML5 content.
* Minimum 1.10 GHz processor or equivalent.
* Between 1-2 GB of free storage
* Minimum 512 MB of RAM
* 3 GB of hard-disk space

## Constraints

The front-end must be user friendly and should not need any one to have any prior knowledge in order to use it.

## Assumptions

The main objective of this project is to implement the use case as previously mentioned (2.3 problem statement) for new dataset that comes through the UI. It is assumed that all aspects of this project have the ability to work together as the designer is expecting and also the data on which our model is trained is as correct as possible

# Design Details

## Process Flow

For accomplishment of the task, we will use a trained Machine Learning model. The process flow diagram is shown below:

**Data Preparation**

**Model**

**Development**

**Deployment**

## Event Log

## **🔹 Event Log Table**

| **Term** | **Description** |
| --- | --- |
| **Event Log** | The system should log every event so that the user can track internal processes. |
| **Step 1** | The system identifies at what level logging is required (e.g., info, warning, error). |
| **Step 2** | The system logs each and every system flow for full traceability. |
| **Step 3** | Developers can choose the logging method (e.g., file-based logging or database logging) based on project needs. |
| **Step 4** | The system must remain responsive and should not hang or slow down even if a large number of logs are generated. |
| **Purpose** | Logging is essential for debugging and monitoring the application; it helps identify and resolve issues quickly and efficiently. Logging is mandatory. |

## Error Handling

**🔹 Error Handling Table**

| **Term** | **Description** |
| --- | --- |
| **Error Handling** | The system must detect and respond to errors gracefully during execution. |
| **Definition** | An error is defined as any unexpected behavior or input that falls outside the normal intended usage. |
| **User Feedback** | When an error occurs, the system should display a clear and user-friendly explanation of what went wrong. |
| **Purpose** | To ensure robustness and improve user experience by preventing crashes and guiding users during failures. |

# Performance

**🔹 Performance Table**

| **Term** | **Description** |
| --- | --- |
| **Performance** | The Cryptocurrency Liquidity Prediction system aims to accurately forecast liquidity levels (low, medium, high) to assist traders and exchanges. |
| **Accuracy** | High prediction accuracy is crucial to prevent misleading stakeholders and to support effective risk management. |
| **Use Cases** | The model’s predictions will be used by cryptocurrency traders, exchanges, and market analysts. |
| **Model Retraining** | Periodic retraining of the machine learning models (Random Forest, LSTM, Meta-model) is essential to maintain and improve prediction accuracy as market conditions evolve. |
| **Scalability** | The system should handle growing data volumes and user requests without degradation in performance. |

## Reusability

The code written and the components used should have the ability to be reused with no problems.

## Application Compatibility

**🔹 Application Compatibility Table**

| **Term** | **Description** |
| --- | --- |
| **Application Compatibility** | The project’s components are integrated primarily using Python, which acts as the communication interface between modules. |
| **Component Roles** | Each component—data preprocessing, model training, prediction, and deployment—has distinct responsibilities handled via Python scripts and libraries. |
| **Integration** | Python ensures seamless data transfer and interoperability between machine learning models, web backend (Flask), and frontend UI. |
| **Dependencies** | Compatibility with libraries like scikit-learn, TensorFlow, Flask, joblib, and others is maintained to ensure smooth operation. |

## Resource Utilization

When any task is performed, it will likely use all the processing power available to it until finished.

# Dashboards

As and when, the system starts to capture the historic/ periodic data for a user, the dashboards will be included display charts over time with progress on various indicators or factors.

**KPIs (Key Performance Indicators)**

**🔹 KPIs (Key Performance Indicators)**

| **KPI** | **Description** |
| --- | --- |
| **Prediction Latency** | The time taken by the system to process input data and display the predicted liquidity level. |
| **Processing Efficiency** | The computational resources and processing power required to run the machine learning models smoothly. |
| **Memory Usage** | The amount of RAM and memory consumed by the application during model inference and data processing on the deployment server. |
| **Model Accuracy** | The predictive accuracy of the machine learning models in correctly forecasting liquidity levels. |
| **Scalability** | The ability of the system to handle increasing volumes of data and user requests without performance degradation. |

.

**Conclusion**

In summary, this project presented a comprehensive overview of the architecture, design, technologies, and performance aspects involved in developing the Cryptocurrency Liquidity Prediction system. By leveraging advanced machine learning models and robust data processing techniques, the system effectively forecasts liquidity levels, providing timely insights to traders and exchange platforms. This predictive capability has the potential to significantly enhance market stability and risk management in the dynamic cryptocurrency ecosystem. The project demonstrates a scalable and efficient solution that can support informed decision-making for various stakeholders in the crypto market.