Real-Time Cryptocurrency Price Forecasting System: Design and Implementation of a Data Engineering Pipeline

YI-CHUN HUANG, 36228559, y.huang65@lancaster.ac.uk

This report outlines the design and implementation of a real-time data processing pipeline for cryptocurrency price forecasting. The system, built using Docker, NiFi, Python, MySQL, and Grafana, efficiently ingests and transforms high-velocity data from the CoinCap API. A Python-implemented ARIMA model is used for the forecasting task. The results, stored in a MySQL database, are visualized using Grafana, demonstrating the effectiveness of a well-structured data processing pipeline in extracting valuable insights from real-time data.

Additional Key Words and Phrases: Data Engineering, Real-time Data Processing, Cryptocurrency Price Forecasting, Apache NiFi, Python, ARIMA Model, MySQL, Grafana, CoinCap API

ACM Reference Format:
Yi-Chun Huang. 2023. Real-Time Cryptocurrency Price Forecasting System: Design and Implementation of a Data Engineering Pipeline. 1, 1 (May 2023), 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The digital era has ushered in a new age of finance, with cryptocurrencies rising as a significant disruptor [1]. Their volatile nature presents both a lucrative opportunity for investors and a challenge for those looking to understand and predict market movements. This dichotomy forms the central problem addressed in our project: designing an effective system to forecast cryptocurrency prices, enabling stakeholders to make informed financial decisions.

The challenge lies not only in the inherent unpredictability of cryptocurrency prices but also in the complex data engineering tasks necessary to build a robust data processing pipeline. These tasks include the ingestion of large volumes of data in real time, the transformation and cleaning of this data to ensure its reliability, and the need for efficient storage and retrieval systems to facilitate downstream processing and analysis. Additionally, the real-time nature of the data and the requirement for rapid forecasting introduces further complexity.

Our motivation for undertaking this project stems from the significant impact of accurate price forecasting on decision-making in the cryptocurrency market. Given the large financial stakes and the rapid growth of digital currencies, developing a system that can help anticipate market movements is of great value. In particular, we aim to leverage an ARIMA model, widely recognized for its effectiveness in time series forecasting [2], to predict future cryptocurrency prices.

In this project, we design and implement an end-to-end data processing system that ingests data from the CoinCap API, cleans and processes the data using Apache NiFi, applies an ARIMA model for forecasting via a Python script, stores the results in a MySQL database, and finally, visualizes the original and forecasted data using Grafana. This

Author's address: Yi-Chun Huang, 36228559, Lancaster, y.huang65@lancaster.ac.uk.

system embodies the application of core data engineering concepts and techniques to a real-world problem, offering valuable insights into the design and operation of data-driven systems.

2 BACKGROUND

The field of cryptocurrency price forecasting has been an active area of research and development, spurred by the dramatic rise of digital currencies over the past decade. Various approaches have been employed to tackle this problem, ranging from traditional statistical to complex machine learning algorithms [3].

One of the most widely applied methods for time series forecasting, including cryptocurrency prices, is the ARIMA model. A multitude of studies have documented the application of ARIMA models for predicting financial market movements, demonstrating their effectiveness and robustness in different market conditions. For instance, a study by Poongodi et al. (2020) applied ARIMA models to forecast Bitcoin prices and found that the model could provide reasonably accurate forecasts [4], and a study by Dhinakaran et al. (2022) used the ARIMA model to perform upward/downward binary classification problems applied to cryptocurrency price predictions, and the results confirmed that the ARIMA model could be used in exchange rate prediction [5].

Moreover, the design and implementation of data engineering pipelines for handling and processing financial market data, including cryptocurrencies, is a well-established practice. For example, a study by Mohapatra et al. (2019) developed a real-time data processing system for cryptocurrency data, providing real-time insights for decision makers [6]. A study by Bang, J., and Choi, M. J. (2019) introduced a real-time fraud detection method to help investors stay out of the market when a pull-up scheme is in place [7].

Given this background, our project seeks to build upon these prior works, leveraging the proven capabilities of ARIMA models and robust data engineering techniques to develop an end-to-end data processing system for cryptocurrency price forecasting.

3 DATASETS AND CHARACTERISTICS

The data for this project, which is critical to cryptocurrency price forecasting, is sourced from the CoinCap API, an acclaimed resource that provides real-time data on countless cryptocurrencies. Its continuous updates on market activity and price fluctuations for over 1,000 cryptocurrencies make it an ideal source of data for our predictive analysis.

The data available from the CoinCap API is characterised by the sheer volume of data due to the wide variety of cryptocurrencies it covers and the frequency with which the data is updated. This large amount of data provides a rich source of information for our predictive models to capture and learn from the complex patterns of cryptocurrency price movements.

Given CoinCap's reputation as a trusted provider of cryptocurrency data, the authenticity of the data is guaranteed, which means that the data is reliable and authentic. This helps us to confidently base our forecasts on this data, knowing that it accurately represents the true dynamics of the cryptocurrency market.

Diversity is another essential feature of our data set. It contains a variety of characteristics associated with each cryptocurrency, such as current price, market capitalisation and trading volume. These different data points provide a comprehensive view of the market situation and are valuable inputs to the ARIMA model, enhancing its predictive power.

Given the real-time nature of the CoinCap API, the speed of the data is a key aspect. This constant flow of data is critical to our project as it allows the forecasting model to be updated with the latest market conditions, thereby improving the accuracy of forecasts.

Despite the significant advantages, CoinCap AP data also has some limitations. Reliance on a single data source may introduce biases, and issues such as downtime or errors may impact data collection. In addition, data is collected in minimum units of minutes, where seconds would be recommended for practical purposes.

4 DESIGN APPROACH

Our data pipeline consists of several interrelated stages, each with a unique role in processing data, from ingestion to visualisation. This modular and sequential design makes the handling and processing of data more efficient, making it more manageable and robust.

(1) `Ingestion`: The first stage of our pipeline involves data ingestion, with data coming from the CoinCap API in real time. We do this using Apache NiFi, a powerful data integration tool. Apache NiFi's ability to handle high-speed data streams makes it an ideal tool for ingesting real-time data from the CoinCap API.

(2) `Transformation`: After ingestion, the data is cleaned and transformed using Apache NiFi and Python to prepare it for analysis. This stage involves removing any inconsistencies or errors in the data and adjusting the structure and format of the data to ensure its quality and reliability. The transformed data is then imported into the predictive model.

(3) `Analysis`: The cleaned and transformed data is fed into an ARIMA model implemented in Python. ARIMA models are a popular choice for time series forecasting as they are able to handle trends and seasonality, making them particularly suitable for forecasting cryptocurrency prices.

(4) `Storage`: After analysis, the raw and forecast data is stored in a MySQL database. MySQL was chosen for its efficiency, reliability and widespread use. It facilitates the retrieval of data for further analysis or visualisation.

(5) `Visualisation`: Finally, Grafana was used to visualise the raw and predictive data. grafana's ability to connect directly to MySQL and its intuitive dashboard interface make it an excellent tool for presenting the data and predictive model results in a meaningful and accessible way.

Figure 1 illustrates the process design for this project. Each stage of the pipeline has been designed with robustness and flexibility in mind, ensuring that the system can be changed and adapted in a timely manner according to demand.

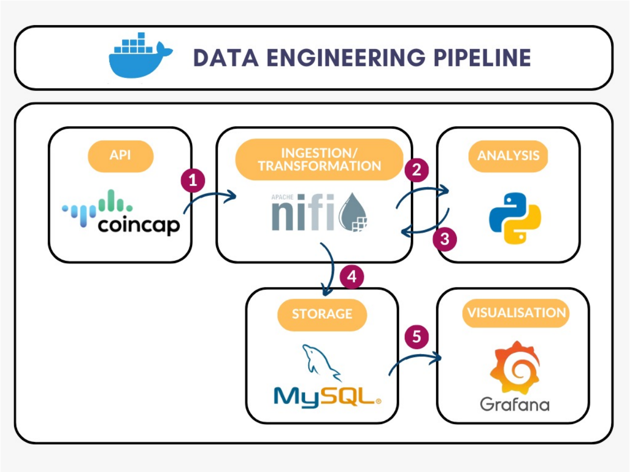


Fig. 1. Pipeline design

5 IMPLEMENTATION

Our project implementation employs a stack of technologies, each chosen for its unique capabilities and suitability to the tasks at hand. This stack includes Apache NiFi, Python, MySQL, and Grafana, integrated to create a seamless data processing pipeline. The integration of these technologies is facilitated through Docker, a platform that allows us to create, deploy, and run applications using containerization. Docker ensures the seamless interaction between different components of our pipeline, providing a unified and resilient environment for the system's operation.

5.1 Apache NiFi

Within a Docker container, we run Apache NiFi to handle the data ingestion and transformation. NiFi connects to the CoinCap API to ingest real-time cryptocurrency data. It then performs various transformation tasks, ensuring that the data is consistent and suitable for analysis. NiFi's dataflow management capabilities and built-in processors make it an ideal choice for these tasks.

This project uses data from the top five cryptocurrencies in the world in terms of volume: Bitcoin, Ethereum, Tether, Binance Coin and USD Coin.

(1) `GenerateFlowFile`: First, we used the GenerateFlowFile processor in NiFi to create a FlowFile every 30 seconds. The reason for this is that the minimum unit of data for this project is minutes, so over-calling data can cause a lot of unnecessary waste of resources and further clog the pipeline.
(2) `InvokeHTTP`: Use the InvokeHTTP processor to send a request to an external HTTP server to obtain the specified data from the server and pass it on as a FlowFile to the next processor for subsequent processing.
(3) `SplitJson`: Since the data is in JSON format, use the SplitJson processor to convert a FlowFile containing multiple JSON objects into a single JSON object, allowing us to easily obtain the most important data.
(4) `ExecuteStreamCommand`: Use the ExecuteStreamCommand processor to analyse the data in Python, such as data cleaning and modelling. The results are then passed on as a FlowFile to the next processor for subsequent processing.
(5) `EvaluateJsonPath`: Use the EvaluateJsonPath processor to extract the values of specified fields from JSON data and store them in the FlowFile attribute, keeping only the data that meets the specified criteria. This reduces the amount of data, increases processing efficiency, and makes it easier for other processors to access these field values and perform post-processing.
(6) `ConvertJSONToSQL`: Use the ConvertJSONToSQL processor to convert a FlowFile containing JSON data into an INSERT SQL statement and then store the data in MySQL.
(7) `ReplaceText`: Use the ReplaceText processor to change the INSERT SQL statement to REPLACE. The purpose of this is to update the data in real time so that there are no conflicting errors when there is duplicate data.
(8) `ExecuteSQL`: Finally, use the ExecuteSQL processor to pass the FlowFile containing the SQL query statement to the database and execute the SQL statement.

In addition, because of the use of the REPLACE SQL query statement, an additional 86,400 second pipeline was set up for the initial data acquisition, using the INSERT SQL query statement.

5.2 Python

The ARIMA model is implemented in Python and integrated through NiFi. Python's rich ecosystem of data analysis libraries, such as pandas for data manipulation and pmdarima for implementing ARIMA models, are used to accomplish this task. Cleaned and transformed data from NiFi is fed into the Python-based ARIMA model, which then outputs the predicted cryptocurrency prices and sends the data back to FlowFile

5.3 MySQL

The raw and predicted data is stored in a MySQL database which also runs in a Docker container. This project uses five data tables to store five different cryptocurrencies, the reason for this is to be able to distribute the risk of data storage in case of errors in the pipeline and to speed up the storage of real time data to cope with the large data streams.

MySQL is a powerful and efficient relational database management system that provides fast data retrieval capabilities, which is essential for the visualisation phase.

5.4 Grafana

Finally, we use Grafana for data visualisation, taking data directly from the MySQL database and presenting it through interactive dashboards. These dashboards provide a visual representation of raw and predicted cryptocurrency prices, allowing stakeholders to see at a glance where the market is moving and thus make informed decisions.

Figure 2 shows an example of the predicted price movement of Bitcoin on the dashboard, where the green line represents the current and past price movements and the red is the predicted price movement.

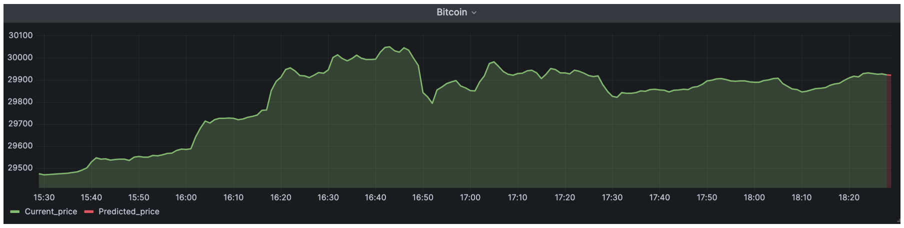
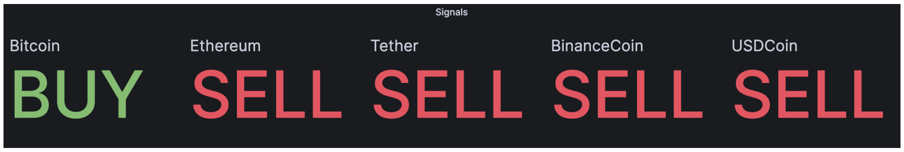


Fig. 2. Bitcoin Future Forecast

Figure 3 shows a signal indicator on the dashboard to help stakeholders quickly capture potential interest in a wide range of currencies



Signals

Bitcoin	Ethereum	Tether	BinanceCoin	USDCoin
BUY	SELL	SELL	SELL	SELL

Fig. 3. Signal to buy or sell

6 EVALUATION

The main goal of our project was to build a robust and efficient data processing pipeline capable of predicting cryptocurrency prices in real time. In implementing and running the system, we made a number of observations that allowed us to evaluate the performance of our pipeline and the effectiveness of our approach.

Firstly, we found the real-time data ingestion and transformation capabilities of Apache NiFi to be very effective. NiFi was able to reliably process the high-speed data stream from the CoinCap API and perform the necessary transformations to ensure that the data was clean and ready for analysis. No data was lost or erroneous in the process, affirming NiFi's robustness in handling large volumes of real-time data streams.

The ARIMA model implemented in Python performed well in terms of training time and was able to respond to data streams on a minute-by-minute basis.

The MySQL database efficiently handles the storage and retrieval of large amounts of data. It provides fast and seamless access to the data, allowing the data to be efficiently provided to Grafana for visualisation.

Grafana's visualisation capabilities are important for presenting data and predicting results in an easy to understand manner. The interactive dashboard allows us to explore the data and forecasts in depth, providing valuable insights into the performance of the predictive models.

Overall, the pipeline has operated efficiently and reliably, demonstrating the effectiveness of the chosen technology and design approach. However, ongoing monitoring and tuning of the system would be beneficial to ensure that its performance is maintained or improved over time.

7 CONCLUDING REMARKS

The implementation and execution of this project offered numerous valuable insights. It reinforced the importance of a well-designed data engineering pipeline in handling real-time data streams and serving them for analytical purposes. Through this project, we learned how to effectively integrate different technologies (NiFi, Python, MySQL, Grafana) in a Docker environment to build a robust, efficient, and scalable data processing system.

The implications of our design are significant. The system we built can handle high volumes of data and process it in real-time, making it suitable for many real-world applications beyond cryptocurrency price forecasting. The modular design allows for the addition or replacement of components, making the system highly adaptable.

However, like any design, our system has its limitations. Our entire system is online and there are not many options for models in order to meet the real-time nature of forecasting. While the ARIMA model is robust, it may not capture all the nuances of the cryptocurrency market.

There are several avenues to explore in terms of improvement. Integrating offline and online, using offline to optimise the model before deploying it online, and updating the model in real time in response to cryptocurrency market fluctuations, would not only speed up the online system, but would also provide a higher degree of confidence in the forecast results. It may also be beneficial to experiment with different or more complex forecasting models. We have found that the larger the number of cryptocurrencies selected, the more likely it is to cause some flow blocking. Therefore, as it gets larger, we should merge the flow files as well as the database in a timely manner, and perform regular performance tuning and system optimisation.

Ultimately, this project has been a valuable learning experience in applied data engineering, demonstrating the power of a well-designed data processing pipeline in extracting insights from real-time data. The knowledge and skills gained through this project will undoubtedly benefit future data engineering efforts.

REFERENCES

[1] Brett Scott, John Loonam, and Vikas Kumar. Exploring the rise of blockchain technology: Towards distributed collaborative organizations. *Strategic Change*, 26(5):423–428, 2017.

[2] Shakir Khan and Hela Alghulaiakh. Arima model for accurate time series stocks forecasting. *International Journal of Advanced Computer Science and Applications*, 11(7), 2020.

[3] Ahmed M Khedr, Ifra Arif, Magdi El-Bannany, Saadat M Alhashmi, and Meenu Sreedharan. Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey. *Intelligent Systems in Accounting, Finance and Management*, 28(1):3–34, 2021.

[4] M Poongodi, V Vijayakumar, and Naveen Chilamkurti. Bitcoin price prediction using arima model. *International Journal of Internet Technology and Secured Transactions*, 10(4):396–406, 2020.

[5] K Dhinakaran, J Divya, C Indhumathi, R Asha, et al. Cryptocurrency exchange rate prediction using arima model on real time data. In *2022 International Conference on Electronics and Renewable Systems (ICEARS)*, pages 914–917. IEEE, 2022.

[6] Shubhankar Mohapatra, Nauman Ahmed, and Paulo Alencar. Kryptooracle: a real-time cryptocurrency price prediction platform using twitter sentiments. In *2019 IEEE international conference on big data (Big Data)*, pages 5544–5551. IEEE, 2019.

[7] Jiwon Bang and Mi-Jung Choi. Design and implementation of storage system for real-time blockchain network monitoring system. In *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 1–4. IEEE, 2019.