Real-Time Cryptocurrency Trend Analysis

Manjot Singh
Big Data Technologies and Applications
Department of Data Analytics
SID: 017557462
manjot.singh@sjsu.edu

Darpankumar Jiyani
Big Data Technologies and Applications
Department of Data Analytics
SID: 017536623
darpankumarpareshbhai.jiyani@sjsu.edu

Sai Prasad Thalluri
Big Data Technologies and Applications
Department of Data Analytics
SID: 017512781
saiprasad.thalluri@sjsu.edu

Saqib Chowdhury
Big Data Technologies and Applications
Department of Data Analytics
SID: 017514978
saqib.chowdhury@sjsu.edu

***Abstract*—The rise of decentralized currencies operating outside conventional financial systems has significantly transformed the global financial arena. Cryptocurrencies like Bitcoin, Ethereum, and numerous altcoins have gained popularity for their high returns, blockchain transparency, and ability to support cross-border, decentralized transactions. However, while these digital assets grow in prominence, the complexities and volatility of crypto markets pose unique challenges. Regulatory ambiguity, security risks, and cryptocurrency price volatility are key obstacles to broader adoption. Addressing these demands robust systems capable of analyzing massive real-time data volumes while ensuring privacy and security. Efficiently processing and interpreting this data is crucial for delivering insights that can guide investors, regulators, and analysts in this evolving market. By leveraging big data tools like Hadoop, Spark, and Kafka, our project seeks to provide real-time trend analysis for cryptocurrencies, offering predictive insights and a more profound understanding of market behaviors. This approach aims to convert raw data into strategic insights, promoting more informed participation in the cryptocurrency space.**

***Index Terms*—Big Data, Criminology, Machine Learning, Data Visualization, Cloud Computing.**

I. Introduction

The rapid expansion of cryptocurrency has redefined the financial landscape by introducing decentralized, borderless, and highly accessible digital assets that bypass traditional financial institutions. Leading cryptocurrencies like Bitcoin, Ethereum, and a growing selection of altcoins are transforming perceptions of value exchange, asset ownership, and financial independence, creating opportunities for transactions and investments outside conventional banking systems. Despite the appeal of these digital assets and their potential for high returns, significant obstacles limit their mainstream adoption. Market volatility, unclear regulatory frameworks, and security concerns contribute to a challenging environment, making it difficult for investors, regulators, and analysts to navigate this dynamic space effectively. Understanding patterns, trends, and relationships within the cryptocurrency market is essential for informed decision-making, particularly given its volatility and rapid pace of change.

To address these complexities, our project leverages advanced big data technologies—such as Hadoop, Spark, and Kafka—to facilitate real-time cryptocurrency trend analysis. These technologies enable the processing of vast volumes of continuously changing data, allowing us to generate predictive insights and a more comprehensive understanding of market behavior. By harnessing the power of big data, our system can transform raw data into actionable insights, equipping users with valuable intelligence that can guide strategic decision-making and enhance engagement with the cryptocurrency ecosystem. This data-driven approach not only helps investors but also supports policymakers and financial analysts who are tasked with managing the complexities of the cryptocurrency market.

Ultimately, our project aims to foster a secure and insightful approach to digital assets, empowering stakeholders to participate more confidently in the evolving world of cryptocurrency. By providing a foundation for better decision-making, our project contributes to a more informed and resilient engagement with this emerging sector, promoting a more transparent, stable, and data-backed cryptocurrency landscape.

II. Literature Survey

A. Big-Crypto: Big Data, Blockchain and Cryptocurrency

The paper "Big-Crypto: Big Data, Blockchain and Cryptocurrency" by Hassani et al. (2018) examines the intersection of big data technologies, blockchain, and cryptocurrency. It highlights blockchain's unique architecture that enables secure, decentralized transactions, positioning cryptocurrencies as a transformative global financial network. The authors discuss technological advancements driving cryptocurrency adoption and address challenges such as scalability, transparency, and security. The study emphasizes the role of big data analytics in enhancing blockchain functionality, enabling predictive insights, and driving innovation within the cryptocurrency ecosystem.

B. Multiscale Characteristics of the Emerging Global Cryptocurrency Market

The paper titled "Multiscale Characteristics of the Emerging Global Cryptocurrency Market" by Marcin Wątorek et al. (2021) provides a comprehensive analysis of the cryptocurrency market using multiscale methods. The authors examine the statistical properties of cryptocurrency price fluctuations and compare them to traditional financial markets. They employ advanced statistical physics techniques to analyze nonlinear correlations and multiscale characteristics within the cryptocurrency market. Additionally, the study explores the evolving correlation structures among the top 100 cryptocurrencies by market capitalization and investigates the impact of the COVID-19 pandemic on market dynamics. This work offers valuable insights into the complex behavior and structural organization of the cryptocurrency market.

C. Cryptocurrency: A Successful Application of Blockchain Technology

The paper "Cryptocurrency: A Successful Application of Blockchain Technology" by Hashemi Joo et al. (2019) explores the role of blockchain technology in the development and adoption of cryptocurrencies. Published in Managerial Finance, the study highlights blockchain's decentralized and secure framework, emphasizing its transformative impact on financial systems, particularly in payment processes. The authors examine the technological strengths of blockchain, such as transparency and immutability, while also addressing challenges like regulatory concerns and scalability. This work provides valuable insights into how blockchain technology supports cryptocurrency applications, fostering innovation and efficiency within modern financial ecosystems.

D. What do we know about cryptocurrency? Past, present, future

The paper "What do we know about cryptocurrency? Past, present, future" by Hossain et al. (2021) provides a systematic review of cryptocurrency's evolution, its impact on financial transactions, and emerging trends. Published in China Finance Review International, the study employs a qualitative-quantitative meta-literature review to analyze the role of cryptocurrency in transforming traditional financial systems. The authors discuss the challenges, trends, and opportunities associated with this innovative investment regime and highlight unexplored research areas in the global cryptocurrency market. This comprehensive review contributes valuable insights into the current state and future agendas of cryptocurrency in the financial landscape.

E. The Challenge of Cryptocurrency in the Era of the Digital Revolution: A Review of Systematic Literature

The paper titled "The Challenge of Cryptocurrency in the Era of the Digital Revolution: A Review of Systematic Literature" by Izwan Amsyar et al. (2020) examines the evolution of cryptocurrency within the context of modern technological advancements. The authors discuss the decentralized nature of digital currencies, their potential to disrupt traditional financial systems, and the challenges they face, such as regulatory hurdles and security concerns. This study provides a comprehensive overview of the current state of cryptocurrency and its implications for the future of finance.

F. A Brief Survey of Cryptocurrency Systems

The 2017 paper by Ujan Mukhopadhyay et al. investigates cryptocurrency systems, focusing on their reliance on secure, distributed ledger structures and the role of mining in adding transactions to the blockchain. The authors examine how mining ensures consensus among users and introduces new currency units into the ecosystem. Additionally, they address the challenges associated with mining, such as high computational requirements and environmental impact. This work offers valuable insights into the foundational mechanisms and considerations that underpin cryptocurrency operations.

G. Impact of Application of Big Data on Cryptocurrency

The chapter "Impact of Application of Big Data on Cryptocurrency" by Sharma et al. (2020) explores the integration of big data technologies with cryptocurrency systems. Published in the book Cryptocurrencies and Blockchain Technology Applications, the chapter highlights the advanced architecture of cryptocurrencies, particularly their reliance on blockchain technology, which replaces traditional cryptographic methods. The authors discuss key applications of big data in analyzing cryptocurrency operations and its impact on this emerging financial domain. This study provides valuable insights into the broader applicability of big data within cryptocurrency and its transformative effects across various industries.

H. A Critical Investigation of Cryptocurrency Data and Analysis

The paper titled "A Critical Investigation of Cryptocurrency Data and Analysis" by Carol Alexander et al. (2019) examines the reliability of cryptocurrency data used in academic research. The authors highlight that less than half of the cryptocurrency papers published since January 2017 employ correct data, raising concerns about the validity of findings in the field. Their analysis underscores the importance of rigorous data verification and methodological consistency in cryptocurrency research. This study provides valuable insights into improving data standards for more reliable and meaningful analysis within the cryptocurrency domain.

III. Proposed System

The system is designed to process real-time cryptocurrency market data efficiently by combining data streaming and advanced big data algorithms. It uses Apache Kafka to stream data from the CoinGecko API, where a producer continuously sends information like cryptocurrency ID, price, market cap, and volume to a Kafka topic called coingecko. A consumer then retrieves this data and stores it in HDFS on an AWS EMR cluster, making it ready for distributed processing. Key algorithms such as MapReduce calculate average prices, Reservoir Sampling selects random subsets, and DGIM performs sliding window analysis on price trends. Techniques like Bloom Filters, Flajolet-Martin, and Locality Sensitive Hashing (LSH) are used to efficiently estimate uniqueness, test membership, and find similarities between cryptocurrencies. To ensure data privacy, the system employs Differential Privacy, and Explainable AI tools provide clear insights into the data processing. This architecture ensures scalable, real-time, and privacy-focused analysis of large cryptocurrency datasets.

A. *Methodology*

The initial step involves utilizing data.gov as a centralized source for historical crime data, which is then efficiently stored in Amazon S3. This choice of storage ensures not only scalability but also provides a cost-effective solution for managing large datasets. To orchestrate the seamless flow of data, we employ an EC2 instance in AWS with Kafka installed. Kafka acts as the backbone of our data pipeline, facilitating the efficient and real-time transfer of crime data from S3. This combination of EC2 and Kafka allows us to manage multiple data streams with low latency, ensuring that our pipeline remains responsive to incoming data. The pivotal technology in our analytical stack is Amazon EMR, where Spark is implemented for distributed computing. EMR's elastic and scalable infrastructure enables parallel processing, which is essential for handling the streaming data received from Kafka. PySpark, embedded within the EMR cluster, plays a crucial role in data transformation during both streaming and batch processing. Its ability to process vast amounts of data in parallel ensures the agility required for timely insights.

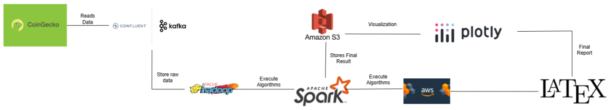


Fig. 1. Methodology

In parallel, we employ PySpark to directly access historical crime data stored in S3 for batch processing. This approach not only capitalizes on the efficiency of PySpark for ETL operations but also allows us to harmonize both real-time and historical data seamlessly.

The machine learning models, integral to our predictive analysis, are trained on the pre-processed data within the EMR cluster. Spark's MLlib provides a scalable machine learning library, enabling us to develop robust models capable of discerning patterns and predicting crime trends effectively.

B. *Data Pre-processing*

Data preprocessing is a crucial step in ensuring that the cryptocurrency market data streamed from the CoinGecko API is clean, relevant, and ready for analysis by advanced algorithms. In this project, preprocessing happens in real-time as data flows from the API to the Kafka topic and is later stored in HDFS on AWS EMR.

The raw data from the API, provided in JSON format, includes fields like id, current_price, market_cap, volume, and other details for each cryptocurrency. However, this data often contains redundant or incomplete information that could affect analysis. To address this, the preprocessing begins at the Kafka Producer stage. Here, only essential fields like id, current_price, and market_cap are extracted, reducing unnecessary data and focusing on what's needed for the algorithms.

As the data moves to the Kafka Consumer, further preprocessing takes place. The consumer parses the JSON messages, validates records, and handles any missing or null values. For example, missing prices can be replaced with an average value for that cryptocurrency, ensuring the dataset remains complete and ready for computation. Once cleaned and validated, the data is saved to HDFS in a structured format, partitioned by timestamp for efficient querying and analysis.

Some preprocessing steps are tailored for specific algorithms. For example, Flajolet-Martin requires cryptocurrency IDs to be hashed into binary strings. For Locality Sensitive Hashing (LSH), numerical values like prices and market caps are normalized to ensure consistency when analyzing high-dimensional data.

By integrating preprocessing directly into the data pipeline, the project ensures that the data stored in HDFS is clean, consistent, and structured for distributed processing. This approach improves the accuracy and performance of the algorithms while enabling real-time insights into cryptocurrency market trends.

C. *Batch and Stream Processing*

Batch and stream processing are critical components of the project's architecture, enabling both comprehensive historical analysis and real-time decision-making. Batch processing is employed to analyze historical data stored in HDFS on AWS EMR. Large datasets are processed in fixed intervals using the MapReduce framework, which efficiently computes aggregate metrics such as the average prices of cryptocurrencies. This method allows the system to handle vast amounts of static data and derive insights over longer time periods, such as historical trends or patterns in cryptocurrency market movements. Batch processing excels in scalability and reliability, making it ideal for computationally intensive operations that do not require immediate results but instead focus on long-term data aggregation and modeling.

On the other hand, stream processing enables real-time data ingestion and analysis, ensuring the pipeline operates continuously with minimal latency. The Kafka Producer streams live market data from the CoinGecko API into a Kafka topic (coingecko), while the Kafka Consumer processes this data as it arrives and writes it into HDFS. Stream processing supports algorithms like Reservoir Sampling, which selects random samples from a live data stream without requiring prior knowledge of its size, which efficiently counts binary events (e.g., price increases) over sliding windows. These techniques allow the system to generate immediate insights into ongoing market trends and fluctuations, supporting timely and actionable decisions. By combining batch and stream processing, the project balances long-term historical analysis with real-time responsiveness, offering a versatile and scalable solution for dynamic cryptocurrency markets.

D. Modeling

The modeling phase of the project is designed to apply big data algorithms to analyze real-time cryptocurrency market data effectively. The distributed processing capabilities of AWS EMR, coupled with Hadoop and Spark, allow the system to handle large-scale data efficiently. The first step in modeling involves aggregation using MapReduce, where the Mapper extracts key-value pairs (id and current_price), and the Reducer computes the average price for each cryptocurrency. This provides a scalable and efficient way to derive critical metrics like averages across large datasets stored in HDFS.

To enable sampling and membership testing on the streamed data, Reservoir Sampling and Bloom Filters are implemented. Reservoir Sampling randomly selects a fixed-size subset of cryptocurrencies, allowing for exploratory analysis without processing the entire dataset. Bloom Filters are employed to perform efficient membership checks, such as verifying whether a particular cryptocurrency (e.g., Bitcoin) exists in the stream. These techniques ensure memory and computational efficiency, even with the high volume of real-time data.

Advanced algorithm like Flajolet-Martin are used to approximate counts and detect trends in the data stream. The Flajolet-Martin algorithm estimates the number of unique cryptocurrencies in the dataset by analyzing trailing zeroes in the hashed IDs, achieving logarithmic efficiency. This algorithms are critical for handling continuous data streams where exact computation would be too resource-intensive.

For more complex analyses, Locality Sensitive Hashing (LSH) is used to identify similar cryptocurrencies based on their market cap and total volume. To protect sensitive data, Differential Privacy is applied by adding Laplace noise to attributes like market cap, ensuring privacy without compromising the integrity of aggregate results. Finally, Explainable AI (XAI) techniques, such as SHAP, are integrated to interpret model outputs, highlighting the contribution of features like price and market cap to provide insights into the model's decision making. Together, these modeling techniques transform raw cryptocurrency data into meaningful and actionable results.

E. Evaluation Methods

The evaluation of the project focuses on assessing the performance, accuracy, scalability, privacy, and interpretability of the implemented big data pipeline. The goal is to ensure that the system handles real-time streaming data efficiently, provides accurate insights, scales seamlessly, and maintains both privacy and interpretability. Algorithmic accuracy:he algorithmic accuracy is evaluated by comparing the outputs of algorithms like MapReduce and Flajolet-Martin with expected results. For example, the MapReduce algorithm is tested for correctness by comparing computed average prices against manually calculated values on smaller datasets. Approximation algorithms such as Flajolet-Martin are validated by comparing their outputs (e.g., estimated unique counts or sliding window "1" counts) against exact computations for controlled datasets. Metrics such as Mean Absolute Error (MAE) and percentage errors are used to quantify accuracy.

Scalability: The system's scalability is evaluated by simulating larger data streams and analyzing its ability to process them in real time. Kafka throughput is measured by increasing the rate and volume of data ingestion, while the performance of distributed algorithms like MapReduce and Reservoir Sampling is tested on larger datasets within AWS EMR. Metrics such as latency, throughput, and resource utilization (CPU and memory) are monitored to ensure that the system can handle growing data volumes without significant performance degradation.

Privacy Preservation: To evaluate privacy preservation, the effectiveness of Differential Privacy is tested by adding Laplace noise to sensitive attributes such as cryptocurrency prices and market caps. The results are compared with those from the original dataset to verify that aggregate trends remain intact while individual data points are obscured. The privacy parameter ϵ is tuned to balance data protection with output accuracy, ensuring compliance with data privacy requirements while retaining analytical utility.

Finally, the overall system reliability is tested under real-world conditions by introducing failure scenarios such as sudden data volume surges, Kafka broker failures, or missing data. Recovery time, data integrity, and system uptime are measured to ensure the pipeline is robust and reliable. Performance metrics, including latency, throughput, and resource utilization, are also monitored to confirm that the system performs consistently and effectively under various conditions. These evaluation methods ensure that the pipeline meets the objectives of accuracy, scalability, privacy, and usability in real-time cryptocurrency market analysis.

F. Technical Difficulties

The development and implementation of this big data pipeline involved several technical challenges, particularly due to the real-time nature of the system, the complexity of integrating multiple tools and services, and the need for scalability, privacy, and interpretability. Below are the primary technical difficulties encountered during the project:

1. Real-Time Data Integration Integrating real-time streaming data from the CoinGecko API into Kafka and subsequently into HDFS on AWS EMR posed significant challenges. The variability in API response times and occasional data inconsistencies, such as missing or invalid fields, required robust error handling and validation mechanisms. Ensuring that the Kafka Producer streamed data at consistent intervals without overloading the topic was critical to maintaining a smooth pipeline flow

2. Kafka and AWS EMR Integration Setting up the Kafka Consumer to interact with HDFS on AWS EMR required precise configuration. Ensuring compatibility between Kafka's streaming architecture and the batch-oriented nature of

Hadoop and HDFS was challenging. Writing data in a structured format while avoiding performance bottlenecks during ingestion required extensive tuning of the consumer code and Spark configurations. Additionally, managing Kafka offsets and ensuring no data loss during failures was a key technical hurdle.

3. Distributed Algorithm Implementation Implementing algorithms like MapReduce, Flajolet-Martin in a distributed environment came with its own set of difficulties. These algorithms had to be optimized for Spark and Hadoop frameworks to ensure they could process large-scale data efficiently. For instance, Flajolet-Martin's reliance on hashing which ensures correctness and scalability when distributed across multiple nodes in the EMR cluster.

4. Privacy Preservation Incorporating Differential Privacy into the pipeline required balancing the trade-off between accuracy and privacy. Adding Laplace noise to sensitive attributes like cryptocurrency prices and market caps while preserving overall trends was non-trivial. Determining the right privacy parameter ϵ to ensure sufficient privacy without overly distorting the data required extensive experimentation and fine-tuning.

5. Fault Tolerance and Reliability Building a fault-tolerant system to handle potential failures, such as network disruptions, Kafka broker downtime, or node failures in the EMR cluster, was critical. Ensuring data consistency and recovery mechanisms, such as replaying Kafka messages using offsets or managing Spark job checkpoints, added complexity to the system.

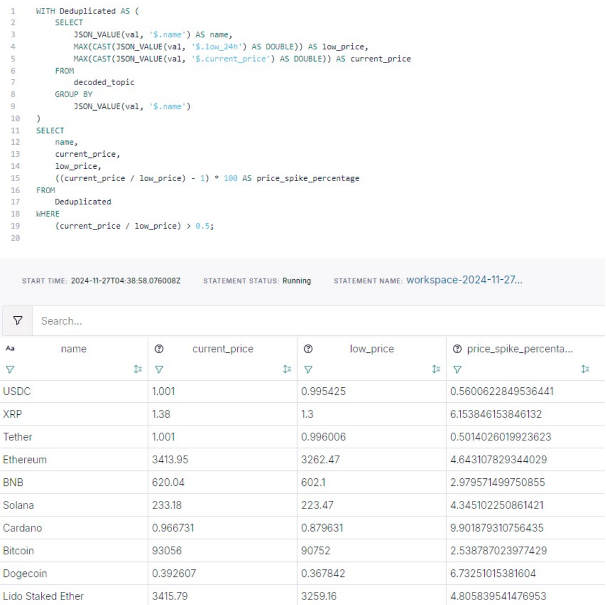
IV. Innovation

The project introduces a highly innovative approach by combining real-time data streaming, distributed processing, and advanced big data algorithms into a fully managed pipeline using AWS and Kafka. Unlike traditional batch-processing systems, this pipeline seamlessly integrates real-time cryptocurrency market data from the CoinGecko API via Kafka, processes it in a scalable manner using AWS EMR, and applies cutting-edge algorithms like Flajolet-Martin, and Locality Sensitive Hashing (LSH). This integration allows for continuous analysis of high-velocity data streams, providing near-instantaneous insights into trends, patterns, and market dynamics. The incorporation of Reservoir Sampling and Bloom Filters enhances efficiency by enabling lightweight sampling and membership testing, reducing computational overhead while maintaining accuracy.

What sets this project apart is the inclusion of Differential Privacy and Explainable AI (XAI) within a big data processing pipeline. Differential Privacy ensures that sensitive cryptocurrency data, such as prices and market caps, is protected through the addition of Laplace noise, balancing privacy with analytical utility. Explainable AI tools, like SHAP, are integrated to provide transparency into insights and predictions, enabling end users to understand the driving factors behind market trends. The combination of real-time streaming, advanced big data algorithms, privacy preservation, and interpretability represents a novel and comprehensive solution for scalable cryptocurrency market analysis, making this system both powerful and user-centric.

V. Results

A. Flink Queries

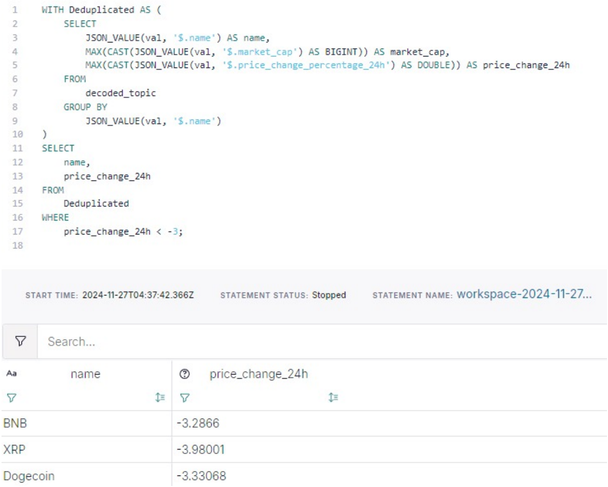


```sql
WITH Deduplicated AS (
    SELECT
        JSON_VALUE(val, '$.name') AS name,
        MAX(CAST(JSON_VALUE(val, '$.low_24h') AS DOUBLE)) AS low_price,
        MAX(CAST(JSON_VALUE(val, '$.current_price') AS DOUBLE)) AS current_price
    FROM
        decoded_topic
    GROUP BY
        JSON_VALUE(val, '$.name')
)
SELECT
    name,
    current_price,
    low_price,
    ((current_price / low_price) - 1) * 100 AS price_spike_percentage
FROM
    Deduplicated
WHERE
    (current_price / low_price) > 0.5;
```

START TIME: 2024-11-27T04:38:58.076008Z STATEMENT STATUS: Running STATEMENT NAME: workspace-2024-11-27...

name	current_price	low_price	price_spike_percenta...
USDC	1.001	0.995425	0.5600622849536441
XRP	1.38	1.3	6.153846153846132
Tether	1.001	0.996006	0.5014026019923623
Ethereum	3413.95	3262.47	4.643107829344029
BNB	620.04	602.1	2.979571499750855
Solana	233.18	223.47	4.345102250861421
Cardano	0.966731	0.879631	9.901879310756435
Bitcoin	93056	90752	2.538787023977429
Dogecoin	0.392607	0.367842	6.73251015381604
Lido Staked Ether	3415.79	3259.16	4.805839541476953

Find cryptocurrencies where the current price is at least 5% higher than their 24-hour low price, ensuring unique names.

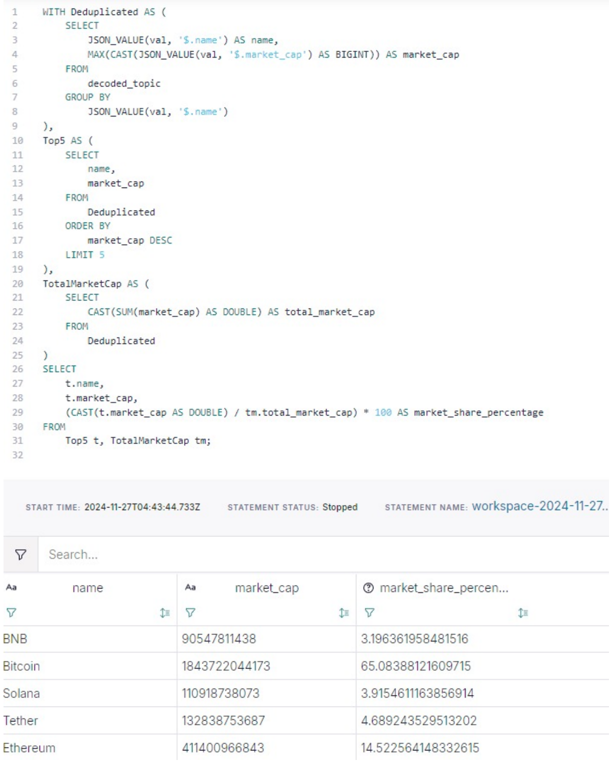


```sql
WITH Deduplicated AS (
    SELECT
        JSON_VALUE(val, '$.name') AS name,
        MAX(CAST(JSON_VALUE(val, '$.market_cap') AS BIGINT)) AS market_cap,
        MAX(CAST(JSON_VALUE(val, '$.price_change_percentage_24h') AS DOUBLE)) AS price_change_24h
    FROM
        decoded_topic
    GROUP BY
        JSON_VALUE(val, '$.name')
)
SELECT
    name,
    price_change_24h
FROM
    Deduplicated
WHERE
    price_change_24h < -3;
```

START TIME: 2024-11-27T04:37:42.366Z STATEMENT STATUS: Stopped STATEMENT NAME: workspace-2024-11-27...

name	price_change_24h
BNB	-3.2866
XRP	-3.98001
Dogecoin	-3.33068

Identifying cryptocurrencies with a price drop of more than 3% in the last 24 hours, ensuring unique names by deduplication.

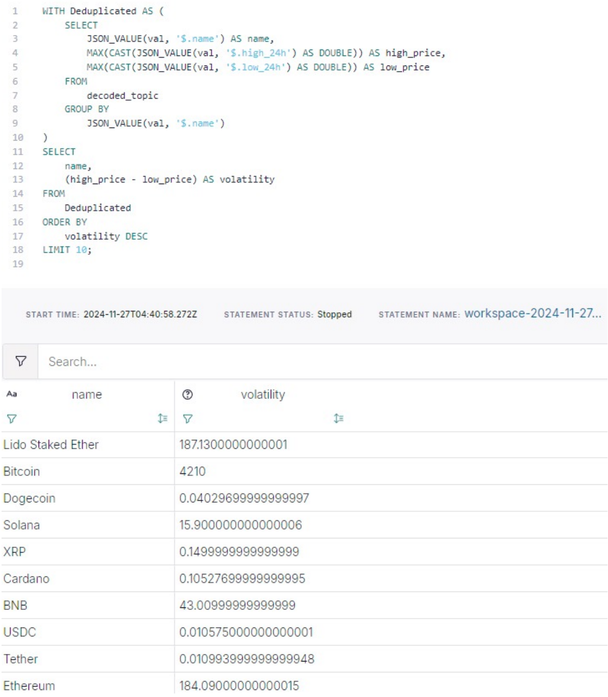


```
WITH Deduplicated AS (
    SELECT
        JSON_VALUE(val, '$.name') AS name,
        MAX(CAST(JSON_VALUE(val, '$.market_cap') AS BIGINT)) AS market_cap
    FROM
        decoded_topic
    GROUP BY
        JSON_VALUE(val, '$.name')
),
Top5 AS (
    SELECT
        name,
        market_cap
    FROM
        Deduplicated
    ORDER BY
        market_cap DESC
    LIMIT 5
),
TotalMarketCap AS (
    SELECT
        CAST(SUM(market_cap) AS DOUBLE) AS total_market_cap
    FROM
        Deduplicated
)
SELECT
    t.name,
    t.market_cap,
    (CAST(t.market_cap AS DOUBLE) / tm.total_market_cap) * 100 AS market_share_percentage
FROM
    Top5 t, TotalMarketCap tm;
```

START TIME: 2024-11-27T04:43:44.733Z STATEMENT STATUS: Stopped STATEMENT NAME: workspace-2024-11-27..

name	market_cap	market_share_percen...
BNB	90547811438	3.196361958481516
Bitcoin	1843722044173	65.08388121609715
Solana	110918738073	3.915461163856914
Tether	132838753687	4.689243529513202
Ethereum	411400966843	14.522564148332615

Calculating the percentage market share of the top 5 cryptocurrencies by market cap.



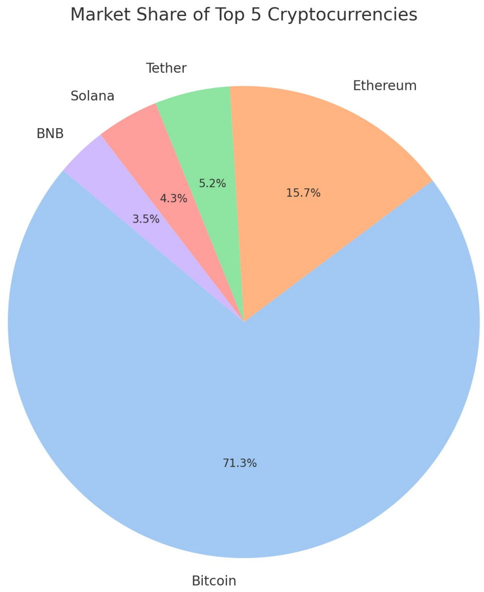
```
WITH Deduplicated AS (
    SELECT
        JSON_VALUE(val, '$.name') AS name,
        MAX(CAST(JSON_VALUE(val, '$.high_24h') AS DOUBLE)) AS high_price,
        MAX(CAST(JSON_VALUE(val, '$.low_24h') AS DOUBLE)) AS low_price
    FROM
        decoded_topic
    GROUP BY
        JSON_VALUE(val, '$.name')
)
SELECT
    name,
    (high_price - low_price) AS volatility
FROM
    Deduplicated
ORDER BY
    volatility DESC
LIMIT 10;
```

START TIME: 2024-11-27T04:40:58.272Z STATEMENT STATUS: Stopped STATEMENT NAME: workspace-2024-11-27...

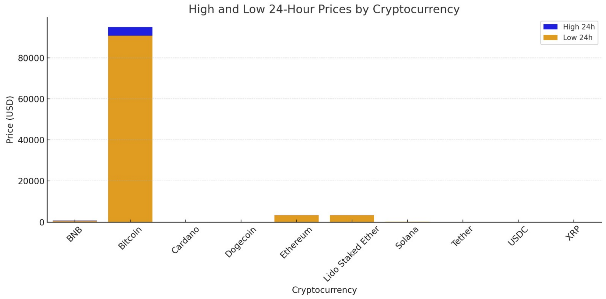
name	volatility
Lido Staked Ether	187.1300000000001
Bitcoin	4210
Dogecoin	0.04029699999999997
Solana	15.900000000000006
XRP	0.1499999999999999
Cardano	0.10527699999999995
BNB	43.00999999999999
USDC	0.010575000000000001
Tether	0.010993999999999948
Ethereum	184.09000000000015

Identifying cryptocurrencies with the highest price volatility (difference between 24-hour high and low).

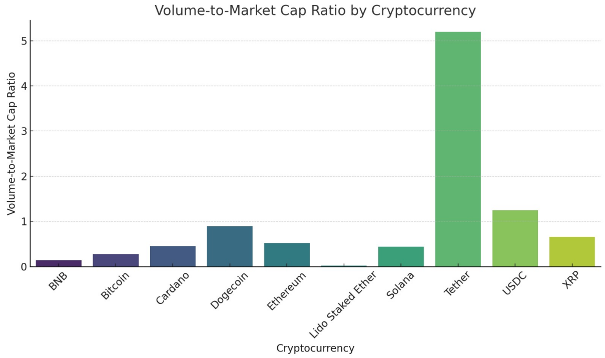
B. Visualisations



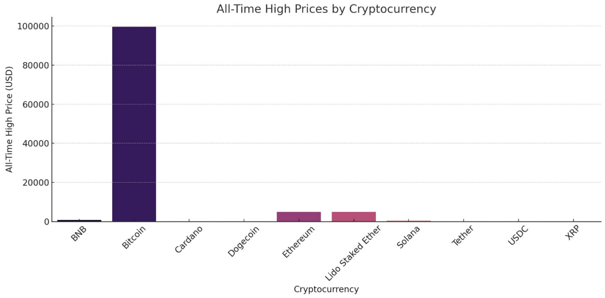
Market Share of Top 5 Cryptocurrencies



High and Low 24-Hour Prices by Cryptocurrency



Volume-to-Market Cap Ratio by Cryptocurrency



All-Time High Prices by Cryptocurrency

VI. Key Learnings

- Developing a real-time big data pipeline provided a deep understanding of integrating various technologies to process and analyze large-scale streaming data. The first key learning was mastering Apache Kafka, which served as the backbone for real-time data ingestion. Setting up the Kafka Producer and Consumer taught us how to stream and manage high-velocity data, as well as handle challenges like offset management and ensuring fault tolerance.
- Working with AWS EMR highlighted the power of distributed computing. Configuring the cluster, managing HDFS storage, and running Spark and Hadoop jobs demonstrated the importance of scalability and resource optimization in big data systems. AWS EMR provided an efficient way to process data in parallel, reducing computational time significantly.
- Using Hadoop MapReduce introduced us to the concept of distributed data processing for tasks like calculating averages and aggregations. Writing custom Mapper and Reducer scripts allowed us to work with structured data in a scalable and efficient manner, even for large datasets stored in HDFS.
- Implementing algorithms like Reservoir Sampling and Bloom Filters showed us how to manage streaming data efficiently. Reservoir Sampling taught us how to randomly sample data from an unbounded stream without needing the entire dataset in memory. Bloom Filters introduced the concept of probabilistic data structures, enabling fast membership testing with minimal memory usage.
- The application of Flajolet-Martin algorithm provided insights into approximate computations. Flajolet-Martin demonstrated how to estimate unique item counts in a data stream using hash functions.
- Learning Locality Sensitive Hashing (LSH) allowed us to explore high-dimensional data for similarity detection. This technique helped us identify cryptocurrencies with correlated price trends, providing a practical approach to clustering and similarity analysis in large datasets.
- Incorporating Differential Privacy into the pipeline was a critical learning point for ensuring data security. We learned how to protect sensitive information, such as cryptocurrency prices, by adding Laplace noise, balancing privacy with analytical accuracy. This highlighted the importance of privacy-preserving techniques in big data systems.
- These technologies, combined into a unified pipeline, demonstrated the importance of scalability, efficiency, and privacy in big data systems. The project provided hands-on experience in building a robust, real-time processing pipeline that is both innovative and practical for real-world applications.

VII. Pair Programming

In the Pair Programming section, we talk about how we worked together as a team to tackle the project. We used Google Meet to connect regularly, discuss our progress, and solve any issues we came across. These meetings gave us a chance to brainstorm ideas, debug code together, and make sure everyone was on the same page. Screen sharing made it easy to collaborate in real time and fix problems quickly. It was a great way to divide tasks, share feedback, and keep things moving smoothly. Overall, working together like this helped us stay focused and get the project done effectively.
Google Meet: 11/19/2024 - Duration: 2h45m
Google Meet Session on 11/10/2024 (Duration: 2 hours)
Google Meet Session on 11/6/2024 (Duration: 1 hour 30 minutes)
Google Meet Session on 11/2/2024 (Duration: 2 hours 15 minutes)
Google Meet Session on 10/28/2024 (Duration: 2 hours)
Google Meet Session on 10/25/2024 (Duration: 1 hour 45 minutes)
Google Meet Session on 10/15/2024 (Duration: 2 hours)
Google Meet Session on 11/14/2024 (Duration: 2 hours)

VIII. Relevance to the course

This project is highly relevant to the concepts and technologies covered in the course, as it applies big data processing techniques and algorithms to solve real-world problems. Key technologies such as Hadoop and MapReduce, which were introduced in class, were implemented to perform distributed data processing on AWS EMR. By utilizing MapReduce for calculating average cryptocurrency prices and aggregations, we gained hands-on experience in writing custom Mapper and Reducer scripts, reinforcing our understanding of its functionality and scalability.

The project also incorporates advanced algorithms like Flajolet-Martin, which was discussed in class as efficient methods for approximate computations on streaming data. We utilised Flajolet-Martin algorithm to estimate the number of distinct cryptocurrencies in the dataset. It leverages hashing and bit manipulation to provide an efficient approximation for large datasets. This implementation helped solidify our knowledge of streaming algorithms and their applications in real-time analytics.

Additionally, LSH algorithm was used to identify similar cryptocurrencies based on their market_cap and total_volume. It finds approximate nearest neighbors, which is useful for clustering or recommendation purposes. This aligns with the course's focus on handling high-dimensional datasets efficiently. Techniques like Reservoir Sampling further expanded our understanding of random sampling methods for unbounded data streams, which were crucial for exploratory analysis.

IX. Impact

This project demonstrates the powerful application of big data technologies to solve real-world problems, specifically in the domain of cryptocurrency market analysis. By leveraging advanced streaming algorithms, distributed processing frameworks like Hadoop and Spark, and privacy-preserving techniques, the project highlights how scalable and efficient systems can be built to handle high-velocity, large-scale data streams. The integration of Explainable AI ensures that the insights generated are not only accurate but also interpretable, making them actionable for analysts and decision-makers. The project's ability to process real-time data, uncover trends, and maintain privacy has significant implications for industries reliant on real-time analytics, such as finance, e-commerce, and healthcare, showcasing how big data technologies can drive informed decision-making and innovation.

X. Conclusion and Recommendations

A. Summary and Conclusions

This project successfully demonstrated how to build a real-time, scalable big data pipeline for analyzing cryptocurrency markets by combining advanced technologies and algorithms. The system starts with real-time data ingestion using Apache Kafka, streaming data from the CoinGecko API into Kafka topics. The data is then consumed and stored in HDFS on AWS EMR, enabling distributed storage and processing. By integrating streaming and distributed systems, the pipeline provides a solid foundation for efficient big data analysis.

We used key algorithms like MapReduce for data aggregation, Reservoir Sampling for random sampling, and Bloom Filters for membership testing. These methods allowed the system to efficiently handle large-scale, continuous data streams while managing memory and computational resources effectively. Advanced streaming algorithms like Flajolet-Martin was also employed to perform approximate counting and detect trends in real-time, ensuring the system could process fast-moving data without compromising performance or scalability.

The project also prioritized data privacy and transparency. By incorporating Differential Privacy, sensitive data such as cryptocurrency prices and market caps were safeguarded by adding Laplace noise, striking a balance between data protection and analytical accuracy. Additionally, Explainable AI tools like SHAP provided clear insights into the factors influencing market trends, enhancing the system's interpretability and building trust in its outputs.

In summary, this project successfully integrated real-time streaming, distributed processing, privacy preservation, and explainability into a single system for cryptocurrency market analysis. It not only showcases the practical application of big data technologies but also addresses critical challenges like scalability, privacy, and transparency, making it a valuable tool for real-world scenarios that rely on real-time analytics.

XI. Team Work

To ensure the project was successfully executed, each team member contributed significantly with specific responsibilities.

TABLE I
Team Responsibilities

Team Member	Responsibilities
Manjot Singh	- Implemented the Kafka Producer to stream real-time data from the CoinGecko API into the Kafka topic. - Developed the Kafka Consumer to read data from Kafka and write it to HDFS on AWS EMR. - Handled real-time data validation and formatting during ingestion to ensure clean data for processing. - Contributed to the Data Preprocessing section of the report and the corresponding slides for the presentation.
Darpankumar Jiyani	- Developed and implemented distributed algorithms like MapReduce, Reservoir Sampling, and Bloom Filters on AWS EMR. - Optimized the Flajolet-Martin for distributed streaming data processing. - Ensured that the algorithms performed efficiently on large datasets by tuning Spark and Hadoop configurations. - Wrote the Modeling section of the report, explaining the role of each algorithm in the pipeline. - Created slides for the presentation to explain the technical details of distributed processing.
Saqib Chowdhury	- Integrated Differential Privacy into the pipeline to secure sensitive data attributes, such as prices and market caps. - Used SHAP for Explainable AI to provide interpretability to the insights and explain the contributions of key features like price and market cap. - Conducted evaluation of the pipeline using metrics such as accuracy, scalability, and privacy preservation. - Wrote the Evaluation Methods and Impact sections of the report. - Created the presentation slides for Privacy and Explainability, ensuring they were visually engaging and easy to understand.
Sai Prasad Thalluri	- Assisted with the development of the Kafka Consumer, particularly refining data validation and integration with HDFS. - Helped optimize Reservoir Sampling and Bloom Filter implementations for efficient memory usage. - Developed Apache Flink queries to enhance real-time processing. - Compiled and formatted the final report, ensuring consistency and completeness. - Designed and finalized presentation slides, focusing on the architecture, summary, and recommendations sections.

References

[1] H. Hassani, X. Huang, and E. Silva, "Big-Crypto: Big Data, Blockchain and Cryptocurrency," *Big Data Cogn. Comput.*, vol. 2, no. 4, p. 34, 2018. doi: 10.3390/bdcc2040034

[2] M. Wątorek, S. Drożdż, J. Kwapień, L. Minati, P. Oświęcimka, and M. Stanuszek, "Multiscale characteristics of the emerging global cryptocurrency market," *Physics Reports*, vol. 901, pp. 1–82, 2021. doi: 10.1016/j.physrep.2020.10.005

[3] M. Hashemi Joo, Y. Nishikawa, and K. Dandapani, "Cryptocurrency, a successful application of blockchain technology," *Managerial Finance*, vol. 46, no. 6, pp. 715–733, 2020. doi: 10.1108/MF-09-2018-0451

[4] M. S. Hossain, "What do we know about cryptocurrency? Past, present, future," *China Finance Review International*, vol. 11, no. 4, pp. 552–572, 2021. doi: 10.1108/CFRI-03-2020-0026

[5] I. Amsyar, E. Christopher, A. Dithi, A. N. Khan, and S. Maulana, "The Challenge of Cryptocurrency in the Era of the Digital Revolution: A Review of Systematic Literature," *Aptisi Transactions on Technopreneurship (ATT)*, vol. 2, no. 2, pp. 153–159, 2020. doi: 10.34306/att.v2i2.96

[6] U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu, and R. Brooks, "A brief survey of Cryptocurrency systems," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, Auckland, New Zealand, 2016, pp. 745–752. doi: 10.1109/PST.2016.7906988

[7] S. K. Sharma, R. K. Modanval, N. Gayathri, S. R. Kumar, and C. Ramesh, "IMPACT OF APPLICATION OF BIG DATA ON CRYPTOCURRENCY," in *Cryptocurrencies and Blockchain Technology Applications*, G. Shrivastava, D.-N. Le, and K. Sharma, Eds. Wiley, 2020. doi: 10.1002/9781119621201.ch10

[8] C. Alexander and M. Dakos, "A critical investigation of cryptocurrency data and analysis," *Quantitative Finance*, vol. 20, no. 2, pp. 173–188, 2019. doi: 10.1080/14697688.2019.1641347

Appendix

A. Rubric

In table II, we describe how the rubric is met.

TABLE II
RUBRIC

Criteria	**How it is met**
Code Walkthrough	This task will be performed in the class.
Presentation Skills (Includes time management)	This task will be performed in the class.
Discussion / Q&A	This task will be performed in the class.
Demo	This task will be performed in the class.
Report Format, completeness, language, plagiarism, whether turnItIn could process it	A report is provided as per the demand.
Version Control	We have used GitHub: https://github.com/saiprasadthalluri/DATA_228_Final_Project
Lessons Learned	Section VI describes the key learnings.
Prospects of winning competition / publication	We can explore the submission of our work on Kaggle when there's a competition on this dataset.
Innovation	Section IV describes the innovations.
Evaluation Methods	Section III-E provides the evaluation of each model.
Teamwork	We worked as a team collaboratively to deliver the project.
Technical difficulty	Section III-F describes the technical difficulties in the project.
Practiced pair programming?	Section VII provides Pair programming information.
Practiced agile / scrum	We used Notion: https://www.notion.so/14b18754120b809aa9fac43a0905ebda?v=9ccda7847f5946269c9f635278614e6c
Used Grammarly / other tools for language?	We used Grammarly chrome extension. The screenshot is provided.
Slides	Slides have been submitted.
Used LaTeX	We used Overleaf. The LaTeX files are provided. We don't include screenshots due to size limits.
Used creative presentation techniques	We used Prezi for the presentation.
Literature Survey	Section II provides a literature survey.
Use of Reservoir Sampling, Bloom Filter, Flajolet-Martin, algorithms for graph streams, etc.	The code for these algorithms is provided in the GitHub link.
Use of Spark, Flink, and Kafka	The code for these is provided in the GitHub link.
Use of Locality Sensitive Hashing	The code for LSH is provided in the GitHub link.
Use of Privacy techniques like K-Anonymity, L-Diversity, Differential Privacy, etc.	We used Differential Privacy technique, can be found in the github link.
Any other tools and techniques covered in the course not included in the other criteria Think Machine Unlearning, Explainable AI (if covered in the class); Fairness; Federated Learning; Data Poisoning, Responsible AI, etc.	Used SHAP to provide insights into the models decision making. Code can be found in github link provided.
This criterion is linked to a Learning OutcomeUse of new tool(s) that were not used for any of the HW.	Used Flink to query the data on confluent. Code can be found in github link provided.