ABSTRACT

The emergence of cryptocurrency ecosystems has transformed digital finance, giving rise to integrated platforms known as crypto hubs. These hubs combine blockchain infrastructure, digital asset exchanges, decentralized applications, and analytical tools to support secure and transparent financial operations. As crypto adoption accelerates, organizations increasingly rely on these hubs not only for transactions but also for extracting real-time insights from blockchain-generated data. This creates new opportunities for applying Machine Learning (ML) techniques to enhance forecasting accuracy in highly dynamic markets.

Sales forecasting within a crypto hub environment presents unique challenges due to extreme price volatility, rapid market sentiment shifts, and high-frequency transactional data. Traditional forecasting models often fail to capture these nonlinear patterns. Machine Learning algorithms—such as ARIMA, Random Forest, LSTM networks, and Gradient Boosting—offer advanced capabilities to model complex time-series behavior. By leveraging blockchain's immutable and transparent data streams, ML models can identify hidden trends, detect anomalies, and generate more reliable predictions of sales volume, token demand, and market activity.

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on individual efforts but on the guidance, encouragement, and cooperation of intellectuals, elders, and friends. A number of personalities, in their own capacities, have helped me in carrying out this project work. We would like to take this opportunity to thank them all.

First and foremost, we would like to thank **Dr. K Channakeshavalu**, Principal, EWIT, Bangalore, for his moral support in completing our project.

We would like to thank **Dr. Raja M,** Professor, and Head of Department of ISE, EWIT, Bangalore, for his valuable suggestions and expert advice.

We deeply express our sincere gratitude to our guide **Dr. Vidhya K** Assoc Professor, Department of ISE, EWIT, Bangalore for her able guidance throughout the project and for guiding us to organize the report in a systematic manner.

We thank our Parents and all the Faculty members of the Department of Information Science & Engineering for their constant support and encouragement.

Last but not least, we would like to thank our peers and friends who provided us with valuable suggestions to improve our project work.

ANAND K **(1EW22IS009)**

BHANUPRAKASHRAOHJ **(1EW22IS017)**

CHETHAN G **(1EW22IS031)**

NANDEEP K R **(1EW22IS126)**

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This introduction explores the pivotal role of advanced data analytics and predictive modeling in two dynamic and high-stakes domains unified cryptographic management and sales forecasting. The convergence of these areas—using sophisticated analytical tools like Machine Learning (ML) to manage, secure, or derive insights from complex, often volatile data, such as that found in cryptocurrency markets—represents a cutting- edge approach to strategic business and technological security.

The proliferation of digital transactions, cloud computing, and rigorous data protection regulations has elevated cryptography from a specialized security function to an enterprise-wide foundational layer. Traditional cryptographic infrastructure, often involving multiple and diverse key management systems from various vendors, has become increasingly complex, costly, and difficult to manage. This complexity often creates security gaps and hinders agile deployment of new applications.

CryptoHub represents the industry's response to this fragmentation. It is conceptualized as a unified, centralized cryptographic platform designed to consolidate all essential cryptographic services such as Payment and General-Purpose HSMs, Enterprise Key Management, Public Key Infrastructure (PKI) and Certificate Authority (CA), and Data Protection into a single, cohesive, on-demand solution.

The core benefit of CryptoHub is infrastructure consolidation and operational efficiency. Built on a service-oriented architecture, it allows organizations to access various cryptographic functionalities on-demand, replacing the need for a series of distinct, hardware-centric buildouts. This centralization simplifies key lifecycle management from automated provisioning to comprehensive rotation and revocation.

Traditional cryptographic infrastructure, often involving multiple and diverse key management systems from various vendors, has become increasingly complex, costly, and difficult to manage. This complexity often creates security gaps and hinders agile deployment of new applications. CryptoHub represents the industry's response to this fragmentation.

This data-centric pipeline ensures that businesses can move from reactive planning to proactive, data-informed strategic decision-making.In today's security landscape, the Zero-Trust security model which operates on the principle of "never trust, always verify" is paramount. CryptoHub aligns perfectly with this model by offering a centralized platform for managing cryptographic keys and policies. It enforces strict access controls, multi-factor authentication (MFA), and role-based access controls (RBAC) to ensure that only authenticated and authorized entities can access critical resources. Furthermore, the platform is designed to incorporate future-proofing technologies, such as Post-Quantum Cryptography, ensuring that encrypted data remains secure even as computing power evolves. By unifying essential encryption, key management, and security policy orchestration, CryptoHub is an indispensable component for organizations safeguarding sensitive information in modern, complex, and distributed IT environments. CryptoHub is an indispensable component for organizations safeguarding sensitive information in modern, complex, and distributed IT environments.

In the fiercely competitive retail and e-commerce sectors, accurate sales forecasting is not merely a financial planning exercise but a critical strategic requirement. Sales forecasts form the bedrock for almost all key business decisions, including inventory management, production scheduling, resource allocation , and setting realistic sales targets. Traditional forecasting methods, such as simple time-series analysis or relying solely on subjective expert judgment (like a sales representative's confidence score), are often inadequate. They struggle to handle the sheer volume, velocity, and variety of modern business data and frequently fail to account for complex, non-linear market dynamics.

Machine Learning represents a transformative leap in sales forecasting. ML algorithms, unlike static models, can process thousands of data points simultaneously ranging from internal sales history, product features, and marketing activity to external factors like competitor pricing, seasonal indicators, and even macroeconomic trends. This capability allows ML models to identify subtle, complex patterns and non-linear relationships that human analysts or traditional statistical methods typically miss. Sales forecasts form the bedrock for almost all key business decisions, including inventory management, production scheduling, resource allocation , and setting realistic sales targets, . This capability allows ML models to identify subtle, complex patterns and non-linear relationships

Building a robust ML sales forecasting system involves a structured, multi-step process, defining clear forecasting goals,Data Collection and Preprocessing, Feature Engineering Model Training Model Validation and Deployment and Continuous Learning. This data- centric pipeline ensures that businesses can move from reactive planning to proactive, data- informed strategic decision-making.

The true innovation lies in recognizing that external, seemingly disparate data sources can significantly impact internal business outcomes. The connection between a unified cryptographic platform and sales forecasting lies in the potential to integrate complex external market signals, such as those from the highly volatile cryptocurrency market, as predictive features in an ML sales model. While CryptoHub itself is a security tool, data related to the assets it protects, or the broader digital asset economy, can become powerful inputs.

The confluence of increasingly complex digital security requirements and the demand for highly accurate business predictions defines the modern technological landscape. On one hand, the need for robust data protection across distributed systems has driven the development of platforms like CryptoHub, which aims to consolidate and streamline enterprise-wide cryptographic services, including key management and hardware security modules, into a unified, efficient, and secure architecture essential for maintaining compliance and adhering to Zero-Trust security models. On the other hand, the intense competitive pressure in sectors like retail and finance requires sophisticated predictive capabilities, pushing organizations beyond traditional statistical methods and towards Machine Learning for sales forecasting. ML algorithms offer superior accuracy by processing a massive array of internal data and dynamic external factors—such as economic indicators or even signals from highly volatile markets like cryptocurrency— to uncover non-linear relationships and subtle trends, ultimately enabling businesses to optimize resource Allocation on the security front, adopting a CryptoHub provides crucial benefits such as infrastructure consolidation that lowers the Total Cost of Ownership. This capability allows ML models to identify subtle, complex patterns and non-linear relationships that human analysts or traditional statistical methods typically miss and will not recognize them.

cloud and hybrid environments, and ensures enhanced security and compliance by centralizing key lifecycle management under a unified, Zero-Trust policy structure. Concurrently, implementing Machine Learning (ML) for sales forecasting delivers a powerful competitive edge by achieving a substantial increase in forecast accuracy through the analysis of thousands of non-linear variables, including dynamic external market data like cryptocurrency movements; this advanced analysis leads to proactive, data-driven decision-making, allowing for optimized resource allocation and inventory management, This capability allows ML models to identify subtle, complex patterns and non-linear relationships that human analysts or traditional statistical methods typically miss

1.1 Relevance of the Project

CryptoHub is a platform that supports crypto projects by managing and promoting token sales and tracking user participation and transaction data. Sales forecasting is the process of predicting future demand or revenue based on historical and current data. The relevance between them lies in the fact that data generated through CryptoHub, such as investor activity and sales trends, can be analyzed using sales forecasting techniques to estimate future token demand and fundraising performance. This helps crypto projects plan launches, allocate resources, reduce risk, and make better strategic decisions. through the analysis of thousands of non-linear variables, including dynamic external market data like cryptocurrency movements; this advanced analysis leads to proactive, data-driven decision-making, allowing for optimized resource allocation and inventory management data-driven decision-making, allowing for optimized resource allocation and inventory management

1.2 Problem Statement

In the modern internet era where there are many investors and traders hope to invest and mine cryptos as there life making skills there are many risks in this there is a need to make huge data analysis which can include cryptos past prices and the current trends and is a very time consuming thing.And there are many risks in this as there are many traders who are trading crypto trading have lost lots of money and there feature because of making wrong decisions in investing so our project aims to help crypto traders analyse the large datasets and make the price prediction for the feature and reduce there losses to minimum. . The relevance between them lies in the fact that data generated through CryptoHub.

1.2.1 GOAL

The goal of this project is to design and implement a comprehensive system that combines the capabilities of a CryptoHub-based platform with effective sales forecasting techniques to support strategic decision-making. The project aims to collect, organize, and analyze transaction and participation data generated from CryptoHub in order to identify trends, patterns, and growth opportunities. By transforming raw data into actionable insights, the project seeks to help organizations or crypto projects anticipate future demand, plan token sales more effectively, and optimize marketing and operational strategies.

A key objective of the project is to apply sales forecasting models—such as time-series analysis, trend analysis, or basic machine learning approaches—to predict future sales or fundraising outcomes with greater accuracy. This helps reduce uncertainty, manage financial risks, and improve budgeting and resource allocation. The project also focuses on demonstrating the importance of data-driven forecasting in fast-moving digital and crypto markets, where demand can change rapidly. The project also focuses on demonstrating the importance of data-driven forecasting

Additionally, the project aims to improve transparency and performance monitoring by providing clear metrics and forecast results that stakeholders can easily understand. Through this project, learners gain practical experience in data analysis, forecasting concepts, and the integration of digital platforms like CryptoHub with business intelligence techniques. Overall, the goal is to create a reliable forecasting framework that enhances planning This helps reduce uncertainty, manage financial risks, and improve budgeting and resource allocation. The project also focuses on demonstrating the importance of data-driven forecasting in fast-moving digital and crypto markets, where demand can change rapidly.

The project also focuses on demonstrating the importance of data-driven forecasting in fast-moving digital and crypto markets, where demand can change rapidly. The project also focuses on demonstrating the importance of data-driven forecasting in fast-moving digital and crypto markets, where demand can change rapidly. Through this project, learners gain practical experience in data analysis, forecasting concepts.

1.2.2 Comparison with Existing System

The existing system relies mainly on manual analysis or basic historical data reports with limited accuracy It lacks real-time data integration and predictive capabilities. Decision-making is slower and more prone to errors and uncertainty. The proposed system using CryptoHub data with sales forecasting provides automated, data- driven predictions It improves accuracy, efficiency, and strategic planning through advanced analytics.

1.2.3 Solution/Implementation

The proposed solution integrates CryptoHub data with a structured sales forecasting model. Transaction and participation data are collected, cleaned, and stored systematically. Forecasting techniques are applied to predict future sales or demand trends. Results are visualized through reports or dashboards for easy interpretation. This implementation enables faster, accurate, and data-driven decision-making for investers who invest in crypto curriencies across the year making some impact on the decisions they make and we wish to improve its accuracy efficiency and strategic planning through some analiticals.

1.2.4 Impact

The system improves forecasting accuracy and reduces business risk. It enables better planning, resource allocation, and strategic decisions for Overall efficiency and performance monitoring are significantly enhanced than to before

This project focuses on combining CryptoHub data with sales forecasting methods to build a data-driven forecasting system. It collects and analyzes transaction and participation data to identify sales and demand patterns. The system applies forecasting models to predict future sales or token demand accurately. By overcoming the limitations of manual and traditional systems, it improves accuracy and efficiency. Real- time data integration helps reduce uncertainty and financial risk. A key objective of the project is to apply sales forecasting models—such as time-series analysis, trend analysis, or basic machine learning approaches—to predict future sales or fundraising outcomes with greater accuracy. A key objective of the project is to apply sales forecasting models—such as time-series analysis, trend analysis, or basic machine learning approaches—to predict future sales or fundraising outcomes with greater accuracy This helps reduce uncertainty, manage financial risks, and improve budgeting and resource allocation. The project also focuses on demonstrating the importance of data-driven forecasting in fast-moving digital and crypto markets, where demand can change rapidly. The system applies forecasting models to predict future sales or token demand accurately.

The project also focuses on demonstrating the importance of data-driven forecasting in fast-moving digital and crypto markets, where demand can change rapidly. The project also focuses on demonstrating the importance of data-driven forecasting in fast-moving digital and crypto markets, where demand can change rapidly. Through this project, learners gain practical experience in data analysis, forecasting concepts

The existing system relies mainly on manual analysis or basic historical data reports with limited accuracy It lacks real-time data integration and predictive capabilities. Decision-making is slower and more prone to errors and uncertainty. The proposed system using CryptoHub data with sales forecasting provides automated, data- driven predictions It improves accuracy, efficiency, and strategic planning through advanced analytics.

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION TO THE LITERATURE

have been used in the past to carry out the price prediction task. There are mainly two sets of literature that are highly relevant to this work. One is financial data analysis; the other, time series data analysis.

2.1.1 Forecasting and Trading Cryptocurrencies with ML Machine Learning under Changing Market Conditions

This paper by Sebastião and Godinho (2021) focuses on applying various machine learning techniques to forecast cryptocurrency prices and evaluate their performance under different market conditions such as bull and bear phases. The main objective of the study is to analyze how well machine learning models like Linear Regression, Support Vector Machines (SVM), Random Forest, and Neural Networks can predict cryptocurrency returns and assist in trading decisions. The authors use historical price data of major cryptocurrencies and compare ML-based strategies with traditional buy- and-hold approaches.

The results show that machine learning models can outperform traditional strategies when market conditions change, highlighting the importance of adaptive ML models in crypto forecasting. In addition to model comparison, the authors emphasize the importance of feature selection and market adaptability in cryptocurrency forecasting. The study incorporates technical indicators such as moving averages, momentum, and volatility measures to enhance predictive performance. Furthermore, the research highlights that no single machine learning model performs best under all market scenarios, Furthermore, the research highlights that no single machine learning model performs best under all market scenarios, reinforcing the need for dynamic model selection. This finding is particularly relevant for real-world crypto platforms like CryptoHub, single machine learning model performs best under all market scenarios, reinforcing the need for dynamic model selection. The system applies forecasting models to predict future sales or token demand accurately. In addition to model comparison, the authors emphasize the importance of feature selection. Furthermore, the research highlights that no single machine learning model performs best under all market scenarios, reinforcing the need for dynamic model selection. The system applies forecasting models to predict future sales or token demand accurately.

2.1.2 High-Frequency Cryptocurrency Price Forecasting Using Machine Learning Model

Rodrigues and Machado (2025) present a comparative study of multiple machine learning models for high-frequency cryptocurrency price forecasting. The objective of this research into evaluate the effectiveness of different ML algorithms, such as Random Forest, Gradient Boosting, Support Vector Regression, and Neural Networks, in predicting short-term price movements of cryptocurrencies. The study uses high-frequency trading data to reflect real-time market behavior and assesses model performance using standard evaluation metrics like RMSE and MAE. The findings indicate that ensemble learning models often achieve better prediction accuracy, making them suitable for real-time crypto forecasting system. The authors further analyze the impact of data granularity on forecasting accuracy and demonstrate that high-frequency data significantly improves short-term price prediction.

The study also discusses computational efficiency, highlighting the trade-off between model complexity and execution speed, which is crucial for real-time forecasting applications. By addressing latency and scalability issues, this research provides practical insights into deploying machine learning models in live cryptocurrency monitoring systems, making it highly applicable to integrated platforms that aim to deliver instant price prediction. . By addressing latency and scalability issues, this research provides practical insights into deploying machine learning models in live cryptocurrency monitoring systems.

2.1.3 Predicting the Price of Bitcoin Using Machine Learning

McNally, Roche, and Caton (2018) present one of the early studies that applies deep learning techniques to Bitcoin price prediction. The main objective of this paper is to evaluate the effectiveness of Long Short-Term Memory (LSTM) networks in forecasting Bitcoin prices based on historical time-series data. The authors compare LSTM models with traditional machine learning methods such as ARIMA and Random Forest. Using daily Bitcoin price data, the study demonstrates that LSTM models outperform conventional approaches due to their ability to capture long-term dependencies , making it highly applicable to integrated platforms that aim to deliver instant price prediction.

2.1.4 An Assembly Stock Predictor and Recommender System

As the popularity of cryptocurrencies, and in particular Bitcoin, increased over the years, more studies shifted their focus from the stock market towards the cryptocurrency market.

Articles by Kaminski (2014) and Matta et al. (2015) get close, in terms of methodologies and research questions, to the research that has been done earlier for the stock market.

The studies use Twitter data to analyze relationships between Bitcoin market indicators and Twitter posts containing emotional signals.

The studies find significant correlations between emotional tweets and the closing price, trading volume and intra-day price spread of Bitcoin.

Additionally, the relationship of Google search queries with Bitcoin trading volumes is investigated to identify the impact of search frequencies on cryptocurrency markets.

Past research has mostly focused on classifying user comments in particular fields. Comments on online communities involve considerable use of neologisms, slang, and emoticons that transcend grammatical usage. C.J. Hutto and Eric Gilbert introduced an algorithm called VADER [44]to parse such expressions, and proposed a method to analyze social media texts by drawing on a rule-based model.

2.1.5 Time-Series Forecasting of Cryptocurrencies Using Machine Learning Techniques

Mudassir et al. (2020) conduct a comprehensive comparative analysis of multiple machine learning techniques for cryptocurrency price forecasting. The objective of this study is to identify the most effective ML models for predicting the prices of major cryptocurrencies such as Bitcoin, Ethereum, and Litecoin. The authors evaluate algorithms including Support Vector Machines (SVM), Random Forest, Gradient Boosting, and Neural Networks using historical time-series data, The studies find significant correlations between emotional tweets and the closing price, trading volume and intra-day price spread of Bitcoin.

Additionally, the relationship of Google search queries with Bitcoin trading volumes is investigated to identify the impact of search frequencies on cryptocurrency markets. By addressing latency and scalability issues, this research provides practical insights.

2.2 Source Used

The development of the CryptoHub application relies on a robust integration of external data providers and open-source software libraries. all the datasets used for analysis and forecasting are sourced exclusively from Coin Gecko, which provides comprehensive and reliable historical as well as real-time cryptocurrency data. This includes information on coin prices, market capitalization, trading volumes, and historical trends for a wide variety of cryptocurrencies. By relying solely on Coin Gecko, the project ensures consistency and accuracy in the data, which is crucial for training predictive models like Random Forest and performing meaningful market analysis. It is important to note that this project does not incorporate external news feeds or sentiment data; the focus is entirely on quantitative price and market metrics. While integrating news and sentiment could provide additional insights, restricting the dataset to Coin Gecko allows for a more controlled and precise evaluation of model performance based solely on market data.

This approach simplifies data processing and reduces noise from unstructured sources, making the forecasting models more robust against overfitting and extreme outliers. By using Coin Gecko as the single source, the project emphasizes a data-driven methodology for analyzing cryptocurrency trends and predicting price movements, ensuring that all results and visualizations in CryptoHub are grounded in verifiable, high-quality market data. In the current version of the CryptoHub and Forecasting Project, the focus is entirely on quantitative cryptocurrency data, and no live news feeds are incorporated. All historical and real-time price data, trading volumes, and market capitalization information are sourced exclusively from Coin Gecko, ensuring consistency and reliability in the datasets used for analysis and forecasting.

2.3 Live news

While live news and sentiment analysis can provide additional insights into market movements, they are not included in this project to maintain a controlled environment for evaluating the predictive performance of the Random Forest model. For future enhancements, integrating live news sources such as Crypto Panic, News API, or RSS feeds from major cryptocurrency news platforms like Coin Telegraph and Coin Desk.

This approach simplifies data processing and reduces noise from unstructured sources, making the forecasting models more robust against overfitting and extreme outliers. By using Coin Gecko as the single source, the project emphasizes a data-driven methodology for analyzing cryptocurrency trends and predicting price movements, ensuring that all results and visualizations in CryptoHub are grounded in verifiable, high-quality market data. In the current version of the CryptoHub and Forecasting Project, the focus is entirely on quantitative cryptocurrency data, and no live news feeds are incorporated. All historical and real-time price data, trading volumes, and market capitalization information are sourced exclusively from Coin Gecko, ensuring consistency and reliability in the datasets used for analysis and forecasting.

CHAPTER 3

OBJECTIVE AND METHODOLOGY

The analysis detailed later in this paper requires an understanding of where and why the data was collected, and how cryptocurrencies may vary from standard fait currencies or stocks in companies from traditional stock markets. In this section we will provide more background on these data sources and why they were chosen so that the final analysis is put in the proper context for the reader.

3.1 Blockchain Technology and Cryptocurrency

In this paper we analyze data about the world's two largest cryptocurrencies in terms of market capitalization. The largest is Bitcoin followed by Ethereum. Bitcoin was the first cryptocurrency ever created. The creation of Bitcoin is mysterious as it was created by a person or group of people using the name "Satoshi Nakamoto" and released in 20093. Along with the launch of Bitcoin "Satoshi Nakamoto" published a paper titled "Bitcoin: A Peer-to-Peer Electronic Cash System" which described a peer-to-peer payment system using electronic cash (cryptocurrencies) that could be sent directly from one party to another without the use of a third party to validate the transaction. This innovation is created by the use of the "blockchain" which is like a shared ledger on the peer-to-peer network where all transactions are veried by the network so they cannot be forged.

This innovation is created by the use of the "blockchain" which is like a shared ledger on the peer-to-peer network where all transactions are veried by the network so they cannot be forged. The applications of blockchain technology go beyond peer-to-peer payment systems. Blockchain technology provides security, privacy, and a distributed ledger which makes them applicable for internet-of-things applications, distributed storage systems, healthcare, and more. The range of applications of the blockchain has led to many more blockchains and cryptocurrencies being created (1,658 cryptocurrencies are in existence Cryptocurrencies) are tied to the blockchain because they provide the incentive for machines, and the electricity they consume. In the current version of the CryptoHub and Forecasting Project, the focus is entirely on quantitative cryptocurrency data, and no live news feeds are incorporated. The applications of blockchain technology go beyond peer-to-peer payment systems. Blockchain technology provides security, privacy, and a distributed ledger which makes them applicable for internet-of-things applications, distributed storage systems, healthcare, and more.

3.2 Objective

The primary objective of the CryptoHub project is to design, develop, and implement a robust automated system capable of predicting the future price movements of major cryptocurrencies with a high degree of statistical accuracy. In the highly volatile and decentralized landscape of digital assets, traditional financial models often fail to capture rapid market fluctuations. Therefore, this project aims to leverage advanced Machine Learning (ML) techniques to analyze market patterns and provide actionable insights for investors, traders, and financial analysts. A central component of this objective is the seamless integration of real-time market data. By utilizing the Coin Gecko API, the system seeks to establish a continuous data pipeline that retrieves live pricing, trading volume, and market capitalization metrics, alongside extensive historical data. This ensures that the predictive models are not only trained on past trends but are also responsive to current market conditions. This ensures that the predictive models are not only trained on past trends but are also responsive to current market conditions.

Specifically, the project aims to:

• **Democratize Data Access:** Create a user-friendly interface where complex market data is processed into intelligible signals, helping users navigate the complexity of the crypto market without needing deep technical expertise.

• **Enhance Predictive Capability:** Implement and fine-tune sophisticated algorithms—specifically focusing on time-series forecasting models—to identify non- linear patterns and long-term dependencies in price movements that are invisible to the naked eye, trading volume, and market capitalization metrics, alongside extensive historical data.

• **Risk Mitigation:** Provide users with data-driven forecasts that can serve as a decision-support tool, thereby reducing the emotional and speculative risks typically associated with cryptocurrency trading. By utilizing the Coin Gecko API, the system seeks to establish a continuous data pipeline that retrieves live pricing, trading volume, and market capitalization metrics, alongside extensive historical data.

• **System Scalability:** Develop a modular architecture that allows for the future inclusion of additional cryptocurrencies and sentiment analysis features, ensuring the tool remains relevant as the market evolves. By utilizing the Coin Gecko API, the system seeks to establish a continuous data pipeline that retrieves live pricing, trading volume, and market capitalization metrics, alongside extensive historical data.

3.3 METODOLOGY

The methodology employed for the CryptoHub project follows a rigorous, standard Data Science lifecycle, structured into five distinct phases to ensure reproducibility and optimal prediction accuracy. This quantitative approach relies heavily on the extraction of empirical financial data followed by sophisticated computational analysis.

3.3.1 Data Acquisition and Integration

The foundation of the system is the CoinGecko API, chosen for its reliability, extensive market coverage, and real-time data flow. The methodology involves automated Python scripts to query two primary data types: real-time "spot" prices for immediate display, and historical OHLCV (Open, High, Low, Close, Volume) data for model training. The raw JSON responses are parsed and stored efficiently in a structured format, creating a persistent, accessible dataset.

3.3.2 Data Preprocessing and Cleaning

Raw financial data often contains irregularities unsuitable for direct model training. This phase cleans the dataset by handling missing timestamps through interpolation to maintain time-series continuity. Crucially, Min-Max Normalization is applied to scale all numerical features (price, volume, etc.) between 0 and 1. This step is vital for ensuring that the chosen machine learning algorithms converge efficiently without being biased by the magnitude differences between features. The data is then transformed using a Sliding Window technique to create sequential input-output pairs. This step is vital for ensuring that the chosen machine learning algorithms converge efficiently without being biased by the magnitude differences between features

3.3.3 Model Architecture

To enhance predictive power, synthetic features are engineered from the raw data. This includes calculating and integrating technical indicators such as Moving Averages (SMA/EMA) and the Relative Strength Index (RSI). The core predictive engine utilizes a Long Short-Term Memory (LSTM) network. LSTMs are a specific type of Recurrent Neural Network (RNN) ideally suited for time-series forecasting due to their ability to recognize and 'remember' long-term dependencies within the sequential price data. This step is vital for ensuring that the chosen machine learning algorithms converge efficiently without being biased by the magnitude differences between features. This step is vital for ensuring that the chosen machine learning algorithms converge efficiently without being biased by the magnitude differences between features.

3.3.4 Training, Evaluation, and Validation

The final methodology involves splitting the prepared dataset into training (80%) and testing (20%) sets. The LSTM model is trained iteratively, adjusting its internal weights to minimize the Mean Squared Error (MSE) loss function. Performance is rigorously validated against the unseen test data using statistical metrics, primarily the Root Mean Squared Error (RMSE), to quantify the average magnitude of prediction error and confirm the model's reliability before deployment.

3.4 Related Work

This paper builds on the ideas of a wide range of research and topics. Behavioral economists like Daniel Kahneman and Amos Tversky established that decisions, even ones involving financial consequences, are impacted by emotion and not just value alone. R. J. Dolan's work in "Emotion, Cognition, and Behavior" further supports that decision making is highly impacted by emotions. The insights from these researchers opens up the possibilities to and advantages through tools like sentiment analysis as it indicates that demand for a good, and therefore price, may be impacted by more than its economic fundamentals.

Later researchers found specifically that purchase decisions people made were being impacted from information gathered online. Galen Thomas Panger found that Twitter sentiment correlated with people's general emotional state.

Galen Thomas Panger found that Twitter sentiment correlated with people's general emotional state.

Additionally, he found that social media like Twitter tended to have a calming affect on the end-user rather than amplifying their emotional state. Chen et al. performed textual analysis on a social platform aimed at investors called "Seeking Alpha" and

found that views expressed in articles posted on "Seeking Alpha" were associated with returns and could even predict earnings surprises. In a similar vein Paul Tetlock found that high levels of media pessimism of the stock market impacted trading volumes. Finally, Gartner found in a study that the majority of consumers relied on social media to influence purchase decisions. Later researchers found specifically that purchase decisions people made were being impacted from information gathered online. Later researchers found specifically that purchase decisions people made were being impacted from information gathered online. Galen Thomas Panger found that Twitter sentiment correlated with people's general emotional state.

Other researchers have specifically studied the efficacy of sentiment analysis of tweets. Kouloumpis et al. found that standard natural language processing techniques such as sentence level and document level sentiment scoring was ineffective due to the short nature of tweets and uniqueness of language used. Alexander Pak and Patrick Paroubek showed that separating tweets into positive, negative, or neutral categories could result in effective sentiment analyses. O'Connor et al. showed that the sentiment found in tweets was reactive of public opinion on various topics in national polling.

Their research identified sentiment analysis as a cost saving option versus national polling, but the implication that sentiments from tweets do accurately react the larger population's feelings on topics suggests that it could also be used to predict demand, and therefore, price changes of products. Web data beyond Twitter and social media has been a rich area of research as well. To our knowledge one of the first papers to and that web search data could be used to predict macroeconomic indicators was by Ettredge et al. where they found that searches relating to employment was associated with unemployment rates. Bordino et al.where they found that searches relating to employment was associated with unemployment rates. Bordino et al.

Specific research into Google Trends data has been done as well by Hyunyoung Choi and Hal Varian with the conclusion that simple seasonal auto-regressive models which included Google Trends data as inputs outperformed models that did not use Google Trends data by 5% to 20%. Asur et al. found that tweet volume about recently released movies accurately predicted box-office receipts.

Having established that decisions are inuenced by emotions, that social media can impact decisions, that sentiment analysis of social media can accurately react the larger population's opinions towards something, and that web search data can predict changes in macroeconomic statistics, much research built of those findings to see if they applied to the stock market. A lan Dennis and Lingyao Yuan collected valence scores on tweets about the companies in the S&P 500 and found that they correlated with stock prices. Pieter de Jong et al. analysed minute-by-minute. . Alexander Pak and Patrick Paroubek showed that separating tweets into positive, negative, or neutral categories could result in effective sentiment analyses. Minute stock price and tweet data for 30 stocks in the DOW Jones Industrial Average and found that 87% of stock returns were influenced by the tweets. However, they also looked for the inverse happening, that stock prices were influencing tweets and found little evidence for it. Bollen et al. used a self-organizing fuzzy neural network, with Twitter mood from sentiment as an input, to predict price changes in the DOW Jones Industrial average and achieved 86.7% accuracy.

With the introduction of cryptocurrencies similar work has been done to see if such methods effectively predict cryptocurrency price changes. In the paper "Predicting Bitcoin price fluctuation with Coi Gecko APIs sentiment analysis" by Evita Stenqvist and Jacob, the authors describe their process in which they collected tweets related to Bitcoin, and Bitcoin prices from May 11 to June 11 in 2017. Tweets were cleaned of non- alphanumeric symbols are used.

\#" and \@" as examples of symbols removed). Then Coin Gecko APIs which were not relevant or determined to be too influential were removed from the analysis.

The authors then used VADER (Valence Aware Dictionary and sentiment Reasoner) to analyze the sentiment of each tweet and classify it as negative, neutral, or positive. Only tweets that could be considered positive or negative were kept in the sentimental analysis. Connor Lamon et al. used sentiment of news headlines and tweets to regression performed best to classify these tweets and that they were able to correctly predict 43.9% of price increases correctly and 61.9% of price decreases. Colianni et al. collected tweets from November 15, 2015 to December 3, 2015 and used Naive Bayes and Support Vector Machines to classify tweets and achieved a 255-accuracy increase.

Finally, Shah et al. successfully established a trading strategy using historical prices and Bayesian regression analysis. Another branch of research in this area involves various applications of neural networks. Kimoto et al. used a modular neural network to create a buying and selling timing system for stocks on the Tokyo stock exchange and achieved profitability using their system with simulated stock purchases. The count of positive tweets, neutral tweets, and negative tweets are the features of the dataset.

Guresenetal compared various neural network performance in forecasting stock exchange rates and found that a multilayer perceptron (MLP) neural network performed best. Xhu et al. used stock trading volumes as a neural network input and found that they modestly improved prediction performance over the medium and long terms.

The research presented in this paper builds on of everything above, but is unique in that we solve the problem of predicting cryptocurrency prices changes by combining web search data (in the form of Google Trends) and tweet volume as inputs into a linear model. In addition, we show why sentiment analysis is less useful in its predictive capabilities of cryptocurrencies despite its potential in other areas.

Specific research into Google Trends data has been done as well by Hyunyoung Choi and Hal Varian with the conclusion that simple seasonal auto-regressive models which included Google Trends data as inputs outperformed models that did not use Google Trends data by 5% to 20%. Asur et al. found that tweet volume about recently released movies accurately predicted box-office receipts.

Having established that decisions are inuenced by emotions, that social media can impact decisions, that sentiment analysis of social media can accurately react the larger population's opinions towards something, and that web search data can predict changes in macroeconomic statistics, much research built of those findings to see if they applied to the stock market. A lan Dennis and Lingyao Yuan collected valence scores on tweets about the companies in the S&P 500 and found that they correlated with stock prices. Pieter de Jong et al. analyzed minute-by-minute.

Specific research into Google Trends data has been done as well by Hyunyoung Choi and Hal Varian with the conclusion that simple seasonal auto-regressive models which included Google Trends data as inputs outperformed models that did not use Google Trends data by 5% to 20%. Asur et al. found that tweet volume about recently released movies accurately predicted box-office receipts. With the introduction of cryptocurrencies similar work has been done to see if such methods effectively predict cryptocurrency price changes. In the paper "Predicting Bitcoin price fluctuation with Coin Gecko sentiment analysis" by Evita Stenqvist and Jacob. Resource Visualization: The visualizations should be self-explanatory which can be easily understood by the user.

Finally, Shah et al. successfully established a trading strategy using historical prices and Bayesian regression analysis. Another branch of research in this area involves various applications of neural networks. Kimoto et al. used a modular neural network to create a buying and selling timing system for stocks on the Tokyo stock exchange and achieved profitability using their system with simulated stock purchases. The count of positive tweets, neutral tweets, and negative tweets are the features of the dataset.

CHAPTER 4

REQUIREMENTS SPECIFICATION

4.1 Functional Requirements

•The system should be able to provide these functionalities efficiently.

•Resource Visualization: The visualizations should be self-explanatory which can be easily understood by the user. There will be line plots and graphs which can be used as an effective measure while devising any new program.

•ML algorithm should be able to predict the output efficiently and accurately.

•On exceeding the critical conditions, alert should be sent to the aqua farmers.

•Predict the bitcoin price in hourly manner. i.e., predict for next hour based on past data.

4.2 Non-Functional Requirements

Non-functional requirements are requirements that specifies criteria that can be used to judge the operation of a system rather than the behaviour.

•Usability: System has been made user friendly by including a readme file in the program so that any user facing difficulty can refer it and easily solve there problem.

•Scalability: If more parameters required, it can be added easily. Number of visualizations can be increased. Currently the system predicts for hourly manner this interval can be changed accordingly.

•Reliability: System should give reliable predicted results. In addition, we show why sentiment analysis is less useful in its predictive capabilities of cryptocurrencies despite its potential in other areas.

•Reliability and Availability: CryptoHub must be highly reliable, ensuring continuous access to users. Data fetching from Coin Gecko and processing pipelines should handle API downtime gracefully, possibly with caching or retry mechanisms. Scheduled maintenance should minimize user disruption.

Performance: Our LSTM model will have improved performance because of the use of datasets with lowest time intervals and has high precession. For checking the accuracy, we have shown the performance metrics using RMSE.

•**Documentation:** Coding standards are maintained throughout the project.

•**Reliability and Availability:** CryptoHub must be highly reliable, ensuring continuous access to users. Data fetching from Coin Gecko and processing pipelines should handle API downtime gracefully, possibly with caching or retry mechanisms. Scheduled maintenance should minimize user disruption.

•**Maintainability:** This project has easy maintainability of the web application, can be modifiable and integrated with advanced computational and operational technologies.

•**Accuracy and Data Integrity:** The data fetched from CoinGecko should be validated for completeness and correctness. Forecasting results must be reproducible and based on verified datasets.

•**Compliance:** The system should comply with relevant data usage policies and licensing terms of CoinGecko and other potential APIs.

4.3 Hardware Requirements

System: Core i5 or i7 Processor

Hard Disk :1 TB.

GPU: for matrix calculation

RAM: 8GB

4.4 Software Requirement

•**IDE**: vs code (with Tensor flow)

•**Programming language:** Python,Html,Css,Sql,APIs

•**Library:** Numpy, pandas, Tensorflow ,Flask, scikit-Learn,plotly,matplotlib,seaborn

This chapter gives an insight into the functional and non-functional requirements that the system provides. It also describes the hardware and software requirements that are required for building the system. Resource Visualization: The visualizations should be self-explanatory which can be easily understood by the user.

CHAPTER 5

SYSTEM ANALYSIS AND DESIGN

5.1 System Architecture

The proposed system architecture (Figure 5.1) shows the complete working of the system starting from training the model using the collected dataset to showing the predicted result and appropriate message on the web application.

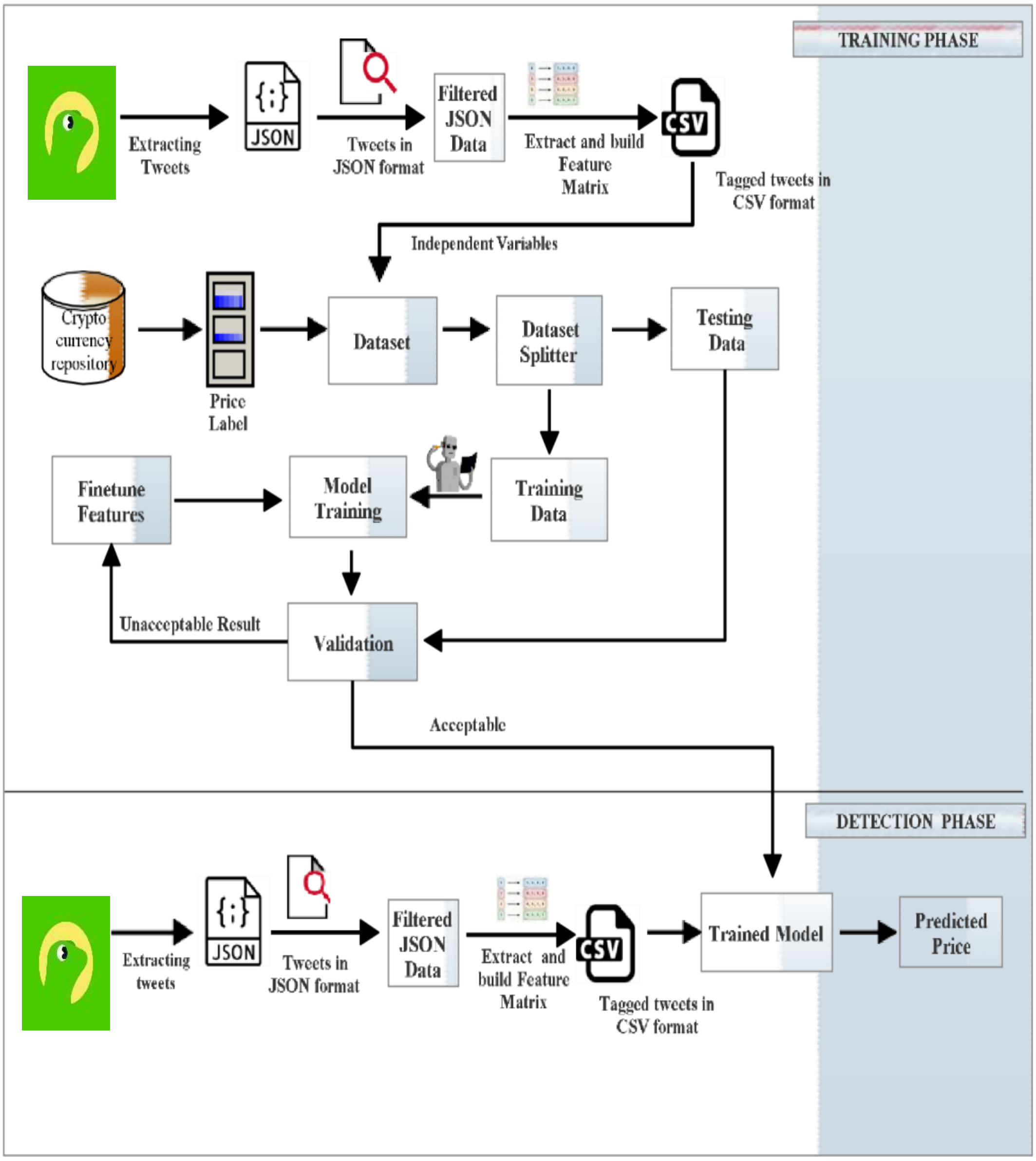


Fig 5.1 Proposed System Architecture

The proposed Architecture (Fig – 5.1) works in two phases Training phase and Detection phase. The training phase is a one-time activity. For carrying out training phase, we have collected Twitter data and the concurrent Bitcoin and Litecoin prices.

The collected Twitter data and prices data are not in same format, the former being in JSON format and the latter being in the CSV format. So, in order to make synchronization in between these two, the Twitter data is converted into CSV format.

The process of conversion of JSON file to CSV file is highlighted in Figure 3. The tweets in the data are analysed for their sentiment polarity. The tweets having polarity above 0 are tagged as positive tweets. The tweets having polarity equals to 0 are tagged as neutral tweets. The tweets having polarity less than 0 are tagged as negative tweets. All the tagged tweets are stored and the stored data is broken into chunks containing tagged tweets which are posted in the time duration of two hours. The number of positive tweets, neutral tweets and negative tweets present in one chunk, are counted.

These counted numbers are then mapped with the average of the prices that occur in corresponding two hours' time duration. The count of positive tweets, neutral tweets, and negative tweets are the features of the dataset, and the mapped average price is the label of the dataset. Model is validated with the original labels of the given dataset. If the result of validation is acceptable, then the model is ready to be used for predicting future price by analysing real-time tweets. If not, then a new model is to be formed. The number of positive tweets, neutral tweets and negative tweets present in one chunk, are counted

The training and testing process are repeated until an acceptable model is formed. Once the acceptable model is formed, the detection phase starts. In the detection phase, real- time tweets are inputted to the model, and the model predicts the average price for the duration of two hours.

These counted numbers are then mapped with the average of the prices that occur in corresponding two hours' time duration. The count of positive tweets, neutral tweets, and negative tweets are the features of the dataset, and the mapped average price is the label of the dataset.The count of positive tweets, neutral tweets, and negative tweets are the features of the dataset, and the mapped average price is the label of the dataset. Model is validated with the original labels of the given dataset.

5.2 Flow Chart

The below flow chart shows the step by step execution implemented at the backend and the frontend of the system:

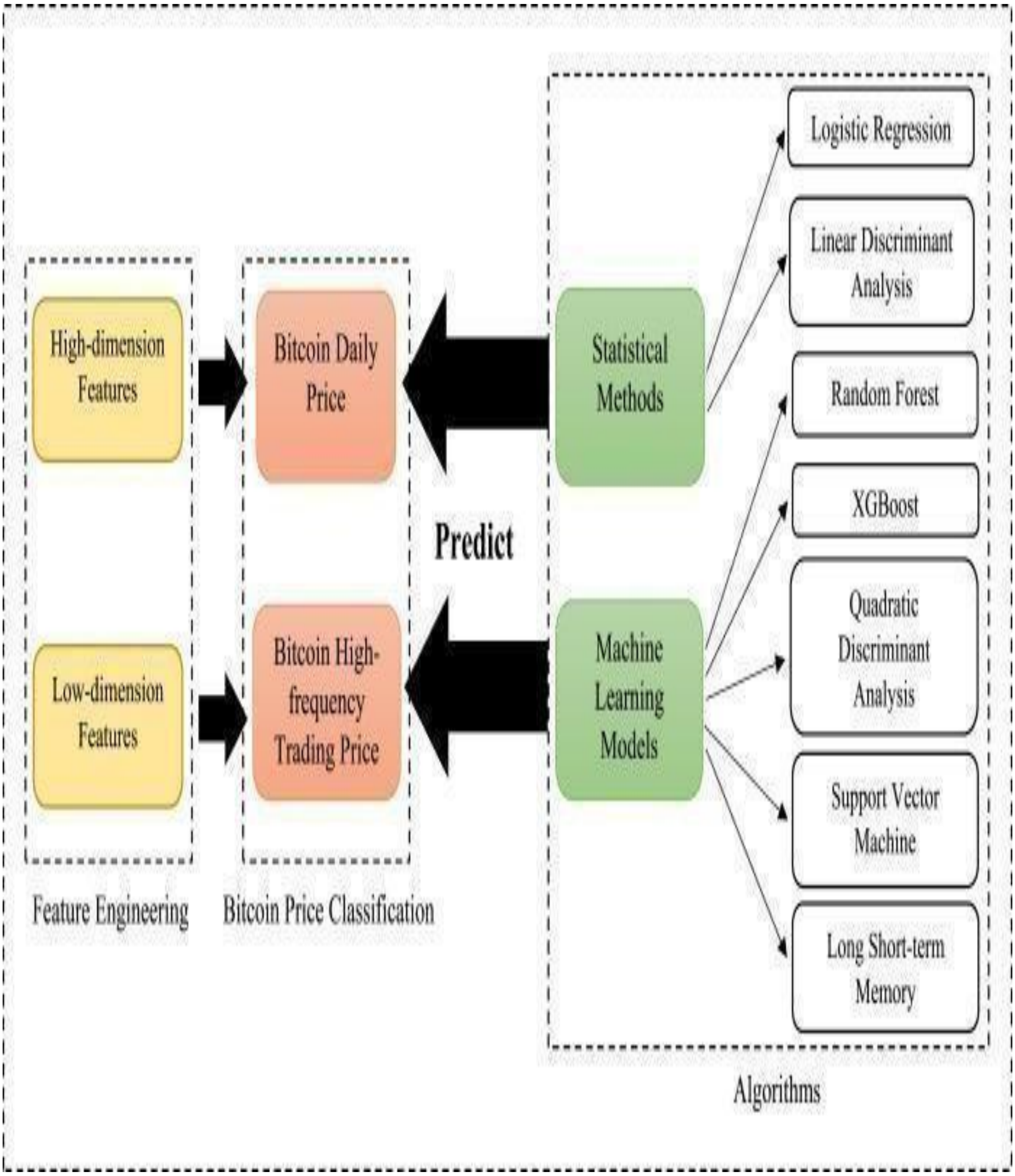


Fig 5.2 System Architecture of Cryptocurrency price prediction

5.3 State Diagram

Figure 5.3 represents the transition between various states of the prediction system. It gives an idea about the various states and the events involved from data collection to generating an alert when any parameter is not in range.

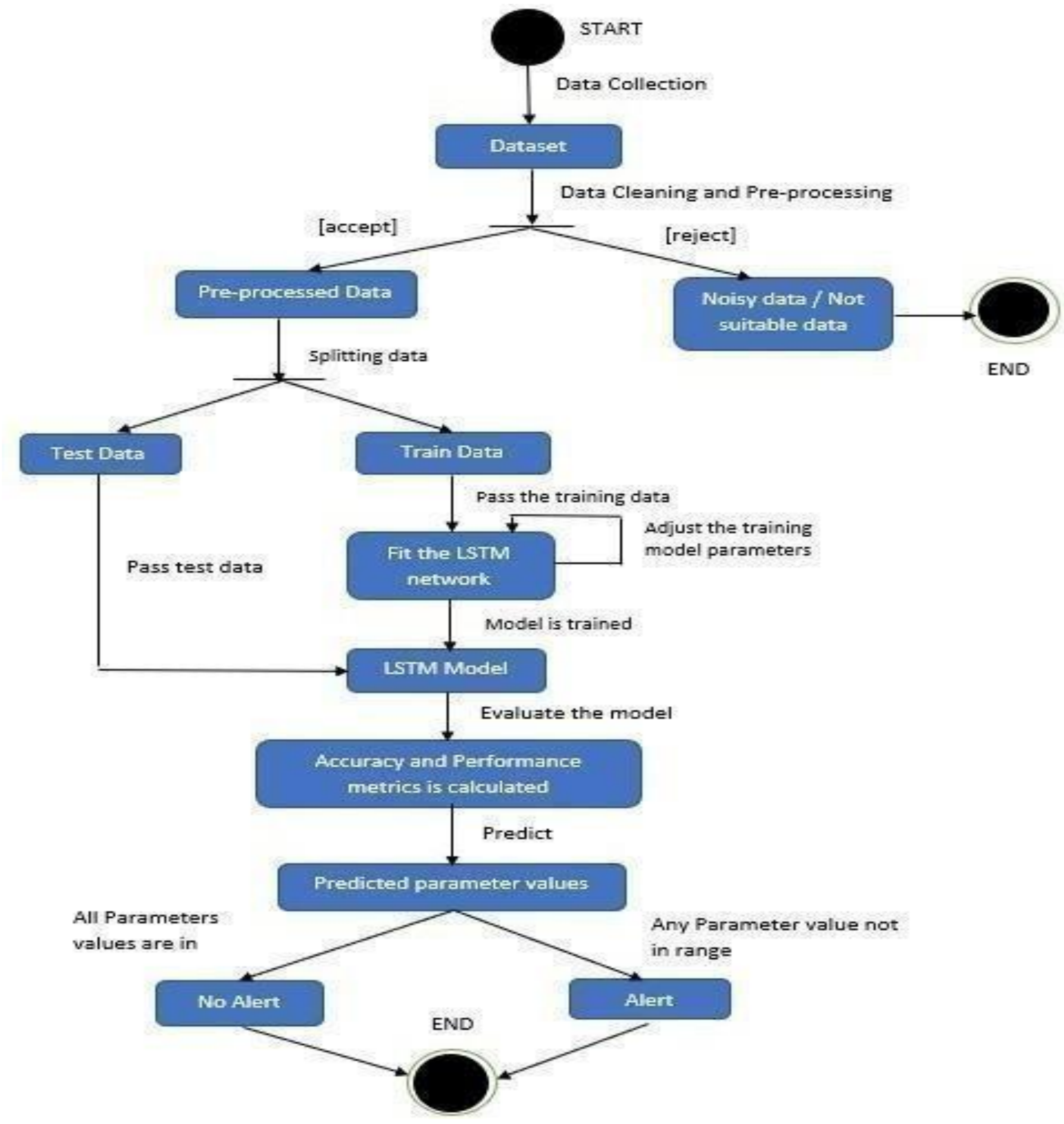


Figure 5.3 State Diagram of Cryptocurrency Prediction System

5.3 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. Here in Figure 5.3 Supervisor is involved in the use cases View prediction result, View Visualization, View Alert and Managing the application

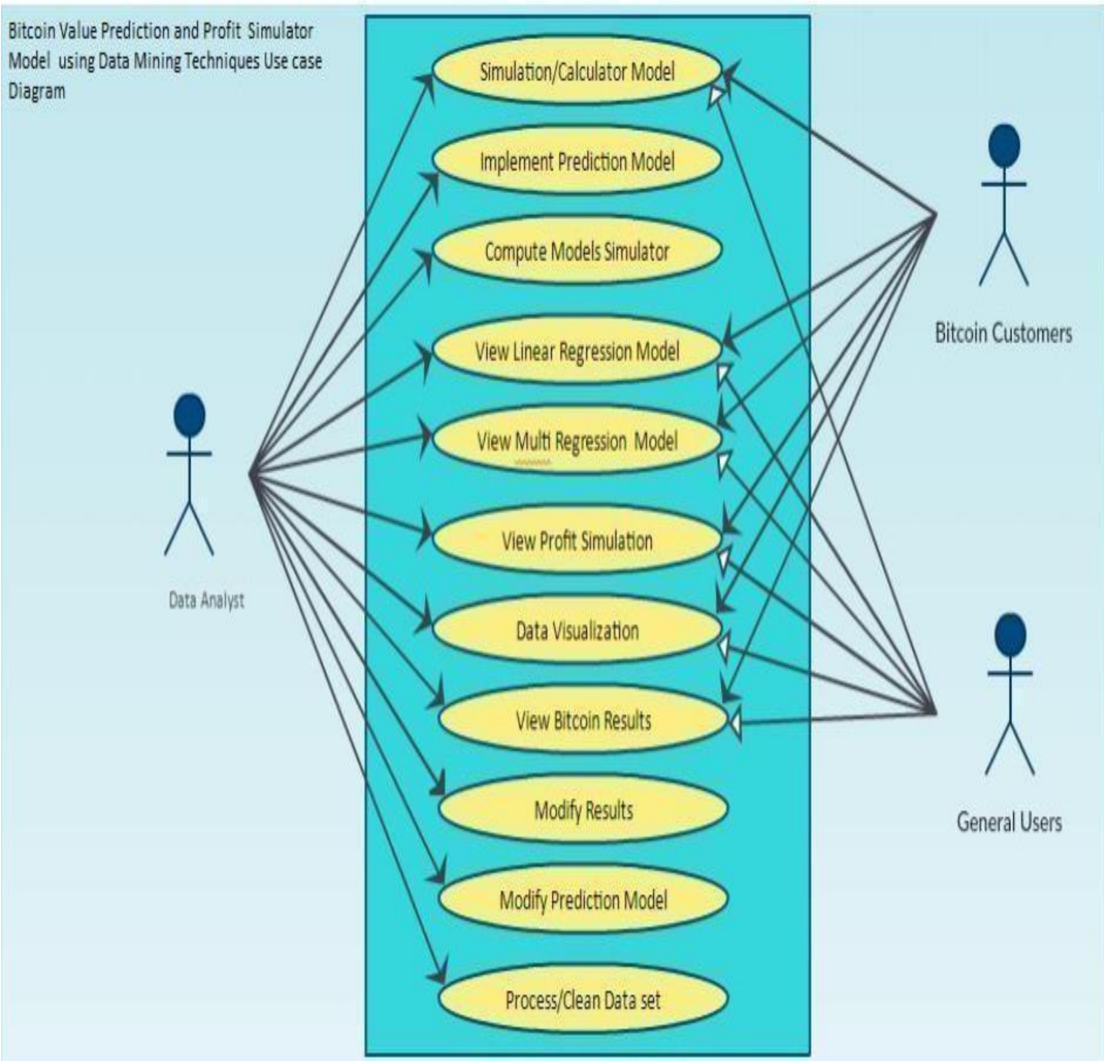


Figure 5.4 Use-Case Diagram of Cryptocurrency Prediction System

This chapter discusses the working of the system through proposed system architecture. The flow diagram shows the working of ML algorithm. The use-case diagram shows interaction between actors and the system. The sequence diagram is shown.

CHAPTER 6

ALGORITHAM

In this section we will discuss about the different algorithms we have used while developing our search engine.

6.1 RNN

Recurrent Neural Network is a generalization of feedforward neural network that has an internal memory. RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation. After producing the output, it is copied and sent back into the recurrent network. For making a decision, it considers the current input and the output that it has learned from the previous input. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. In other neural networks, all the inputs are independent of each other. But in RNN, all the inputs

are related to each other.

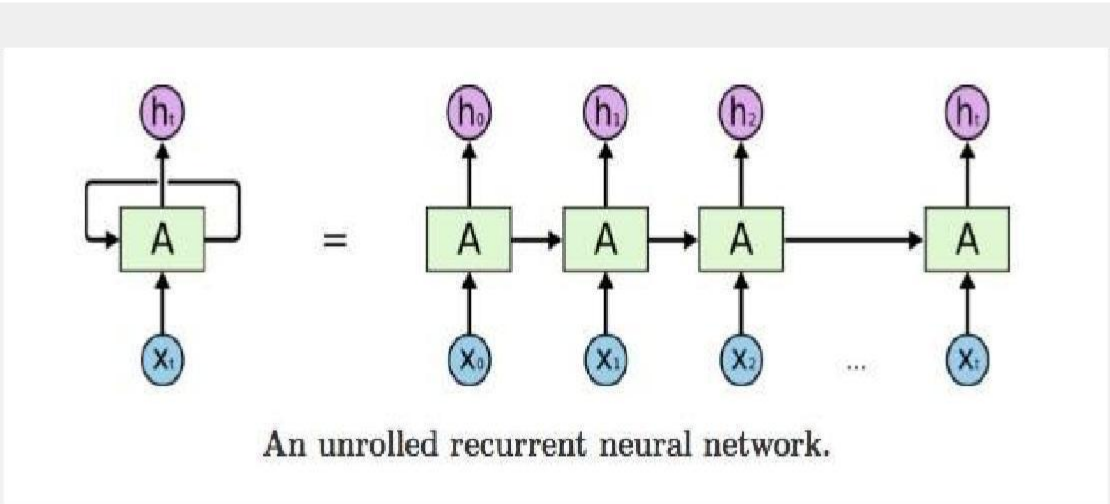


Fig 6.1 Unrolled recurrent neural network

First, it takes the X(0) from the sequence of input and then it outputs h(0) which together with X(1) is the input for the next step. So, the h(0) and X(1) is the input for the next step. Similarly, h(1) from the next is the input with X(2) for the next step and so on. This way, it keeps remembering the context while training.

The formula for the current state is

$$h_t = f(h_{t-1}, x_t)$$

Applying Activation Function:

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

W is weight, h is the single hidden vector, Whh is the weight at previous hidden state, Whx is the weight at current input state, tanh is the Activation function, that implements a Non-linearity that squashes the activations to the range[-1.1]

Output:

$$y_t = W_{hy}h_t$$

Yt is the output state. Why is the weight at the output state.

6.2 LSTM

Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory. The vanishing gradient problem of RNN is resolved here. LSTM is well-suited to classify, process and predict time series given time lags of unknown duration. It trains the model by using back-propagation. In an LSTM network, three gates are present: This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. In other neural networks, all the inputs are independent of each other.

This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. In other neural networks, all the inputs are independent of each other. But in RNN, all the inputs

It trains the model by using back-propagation. In an LSTM network, three gates are present: This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. In other neural networks, all the inputs are independent of each other. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. In other neural networks, all the inputs are independent of each other. But in RNN, all the inputs

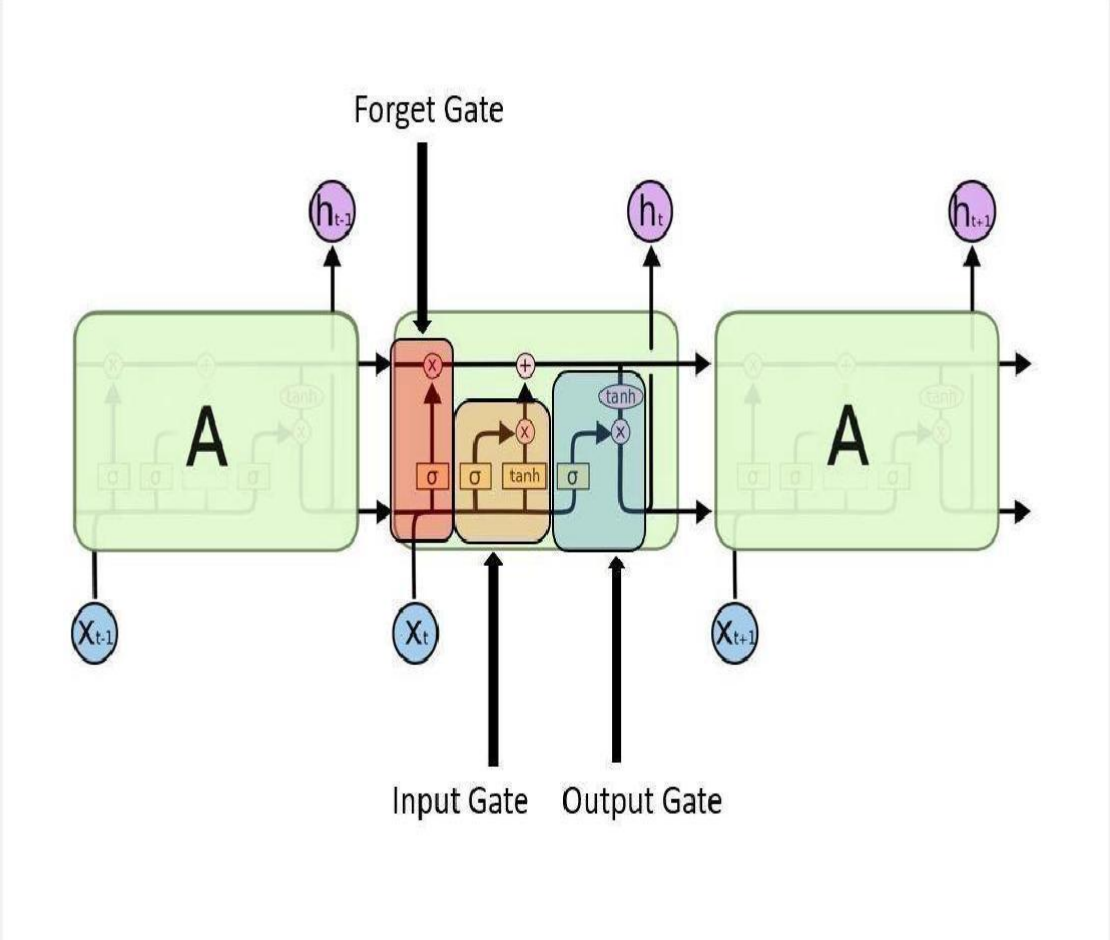


Fig 6.2 - LSTM gates

1. Input gate — discover which value from input should be used to modify the memory. Sigmoid function decides which values to let through 0,1. and tanh function gives weightage to the values which are passed deciding their level of importance ranging from-1 to 1.

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

(6.4) input gate

2. Forget gate — discover what details to be discarded from the block. It is decided by the sigmoid function. it looks at the previous state(ht-1) and the content

input (Xt) and outputs a number between 0(omit this) and 1(keep this) for each number the cell state

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \;+\; b_f\right)$$

Ct−1 Forget gate

Output gate — the input and the memory of the block is used to decide the output. Sigmoid function decides which values to let through 0,1. and tanh function gives weightage to the values which are passed deciding their level of importance

ranging from-1 to 1 and

$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t] \;+\; b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Multiplied with output of Sigmoid.

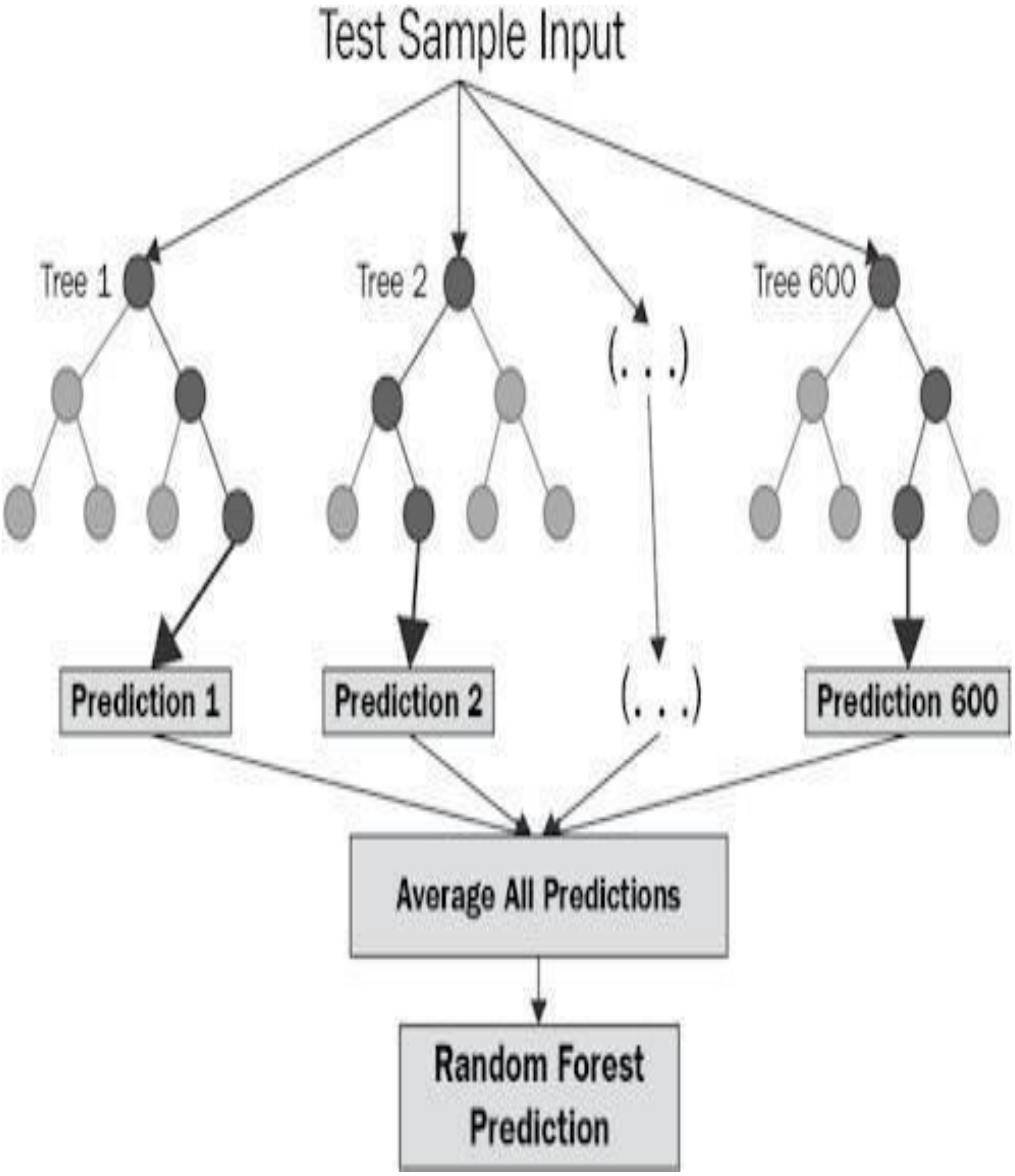
In this section we will discuss about the different algorithms we have used while developing our search engine.

Random Forest

Forest is commonly used in crypto price prediction as a supervised machine learning model that captures complex, non-linear relationships in highly volatile market data. It works by building an ensemble of decision trees trained on different subsets of historical cryptocurrency data, such as prices, trading volume, technical indicators (e.g., RSI, MACD, moving averages)

sometimes sentiment or on-chain metrics. Each tree makes its own prediction, and the final output is obtained by averaging (for price regression) or majority voting (for direction classification), which helps reduce overfitting and noise—both major issues in crypto markets. Random Forest is valued for its robustness, ability to handle large feature sets, and resistance to outliers, making it suitable for short-term price movement or trend prediction.

However, while it often outperforms simple linear models, its predictions are still limited by sudden market shocks, news events, and regime changes common in cryptocurrency markets. It works by building an ensemble of decision trees trained on different subsets of historical cryptocurrency data.



Core Logic of Random Forest

- **Bootstrap Sampling (Bagging)**
- Randomly select multiple subsets of data from the training dataset
- At each split in a decision tree, only a random subset of features is considered.
- This reduces correlation between trees and improves model diversity.
- Trees learn different patterns from different data samples
- Each subset is used to train a separate decision tree.

• **Random Feature Selection**

• At each split in a decision tree, only a random subset of features is considered.

• This reduces correlation between trees and improves model diversity.

• **Tree Construction**

• Each decision tree grows independently and usually to full depth (no pruning).

• Trees learn different patterns from different data samples.

• **Prediction Aggregation**

• **Classification:** Final output is decided by majority voting.

• **Regression:** Final output is the average of all tree predictions.

• **Classification:** Final output is decided by majority voting.

• **Regression:** Final output is the average of all tree predictions.

Simple Example

If 100 trees predict:

• Class A → 60 times

• Class B → 40 times Final prediction = Class A

Random Forest is an ensemble machine learning algorithm widely used for both regression and classification tasks, including cryptocurrency price prediction. It operates by constructing a large number of decision trees during training, where each tree is built on a random subset of the dataset and considers a random subset of features at each splitFor regression tasks, such as predicting future prices, it uses majority voting to determine the predicted class. Random Forest is valued for its robustness, ability to handle large feature sets, and resistance to outliers, making it suitable for short-term price movement or trend prediction.

CHAPTER 7

IMPLEMENTATION

7.1 Data Representation

The used data set is a history of Bitcoin prices per minute from March 1st, 2023 to April 1st, 2023. That is to say, there are 129316 data samples. Each of them has the associated timestamp and Bitcoin price information. For this work, several fields are ignored, starting with the timestamp, since the interval is constant and it is enough knowing the order of the data, being redundant. The lowest and highest value of the minute are not considered for simplicity; there are close in value and somewhat redundant to the weighted price, which is the prediction target. The opening and closing values are also ignored for the same reason as the previous fields. That is, the only value to consider is the weighted price, which can be conceptually seen as the average price of the Bitcoin (inUnited States dollars, USD), for each minute.

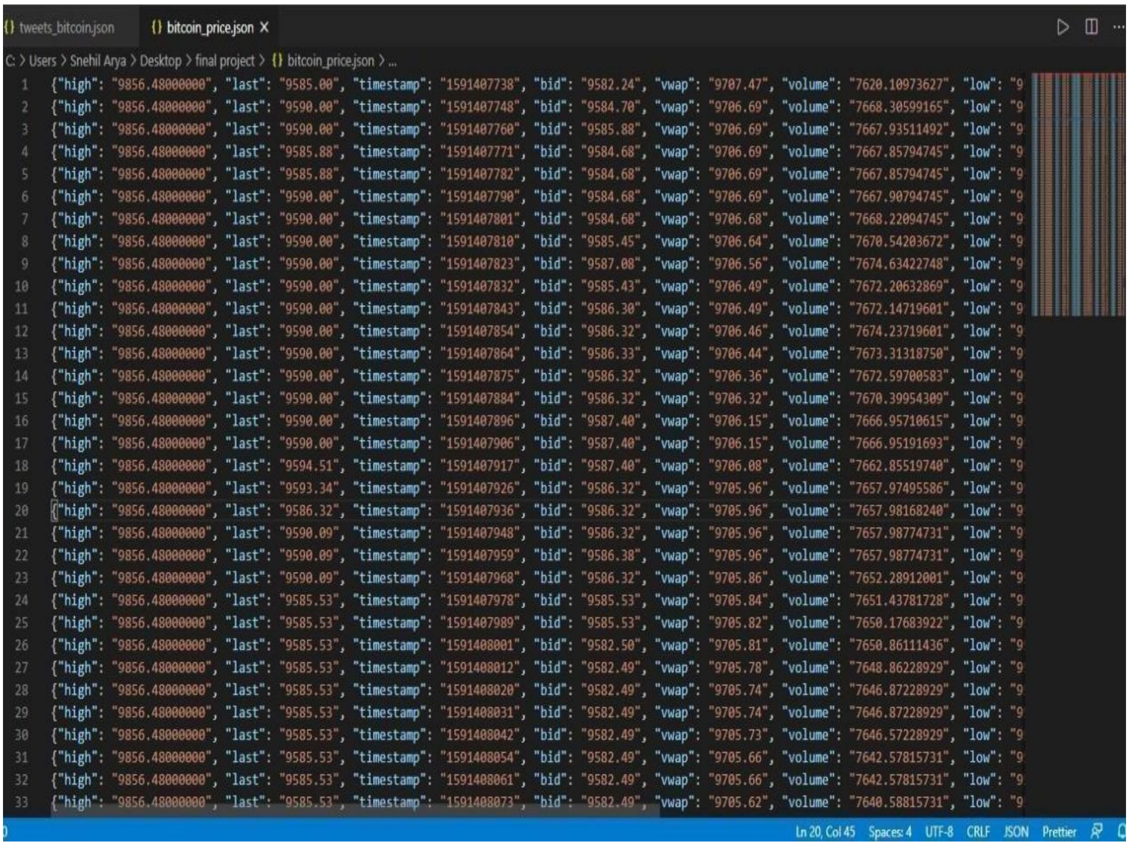


Fig 7.1 Live Bitcoin Price

7.2 Data Pre-processing

The data has been normalized using a minimum-maximum scaler, that is, translating the whole set of price values to the range 0-1, by assigning 0 to the lowest original value, 1 to the highest, and a linear equivalent to the rest of values, lying in between the extremes. Once the scaler has been fitted to the data, it can be used after producing the model to invert the transformation on the predicted values, to recover the original ranges of values for the predictions. The reason to normalize the data is to help the Random Forest and specifically back-propagation and gradient descent learn faster by reducing the magnitude of the value search space.

7.3 Data Split

As previously mentioned, the data is composed of 1293167 instances. The decision of how to split the data was taken trying to both have a large percentage of the data to learn and to keep a reasonably long and heterogeneous sample as the test one. Therefore, the model was trained with the first 120000 instances (92.2% of the data), while the remaining 9316 (7.8%) were used as a test. They correspond to a period of time, during the first half of 2020, with large, fast changes in the value of the Bitcoin, which make the prediction task really challenging. , which make the prediction task really challenging.

7.4 Collection from coin Gecko

For this research work, data is extracted as tweets with the name of cryptocurrencies- Bitcoin, concurrent price data which is extracted from Coin Gecko. For Bitcoin, data is collected from (30 days) March 2020 using the REST API5 of Coin Gecko. For Litecoin, data is collected from March–April 2020. At the same time, per minute price of the Bitcoin is also collected. The collected tweets are obtained in JSON format, and the prices are obtained in .csv format. We tagged tweets as positive, neutral, and negative. For this, Textblob sentiment polarity is used for knowing tweet’s sentiments.

The value returned by “Textblob.sentiment.polarity” is in between -1. Random Forest is valued for its robustness, ability to handle large feature sets, and resistance to outliers, making it suitable for short-term price movement or trend prediction. The decision of how to split the data was taken trying to both have a large percentage of the data to learn and to keep a reasonably long and heterogeneous sample as the test one. We have considered the count of the tagged tweets for two hours because the efficiency of the model increases up to the second hour and after that, it starts decreasing, making two-hour duration an ideal for the consideration.

The information from datasets whose polarity value is 0 are tagged as neutral. The tweets whose polarity value is in between -1 and 0 are tagged as negative. The tweets whose polarity value is in between 0 and 1 are tagged as positive. After collection of tweets and prices, we have counted the number of tagged tweets in two hours duration and the final dataset comprises of the total count of positive, negative and neutral tagged tweets at the end of every two hours. We have considered the count of the tagged tweets for two hours because the efficiency of the model increases up to the second hour and after that, it starts decreasing, making two-hour duration an ideal for the consideration.

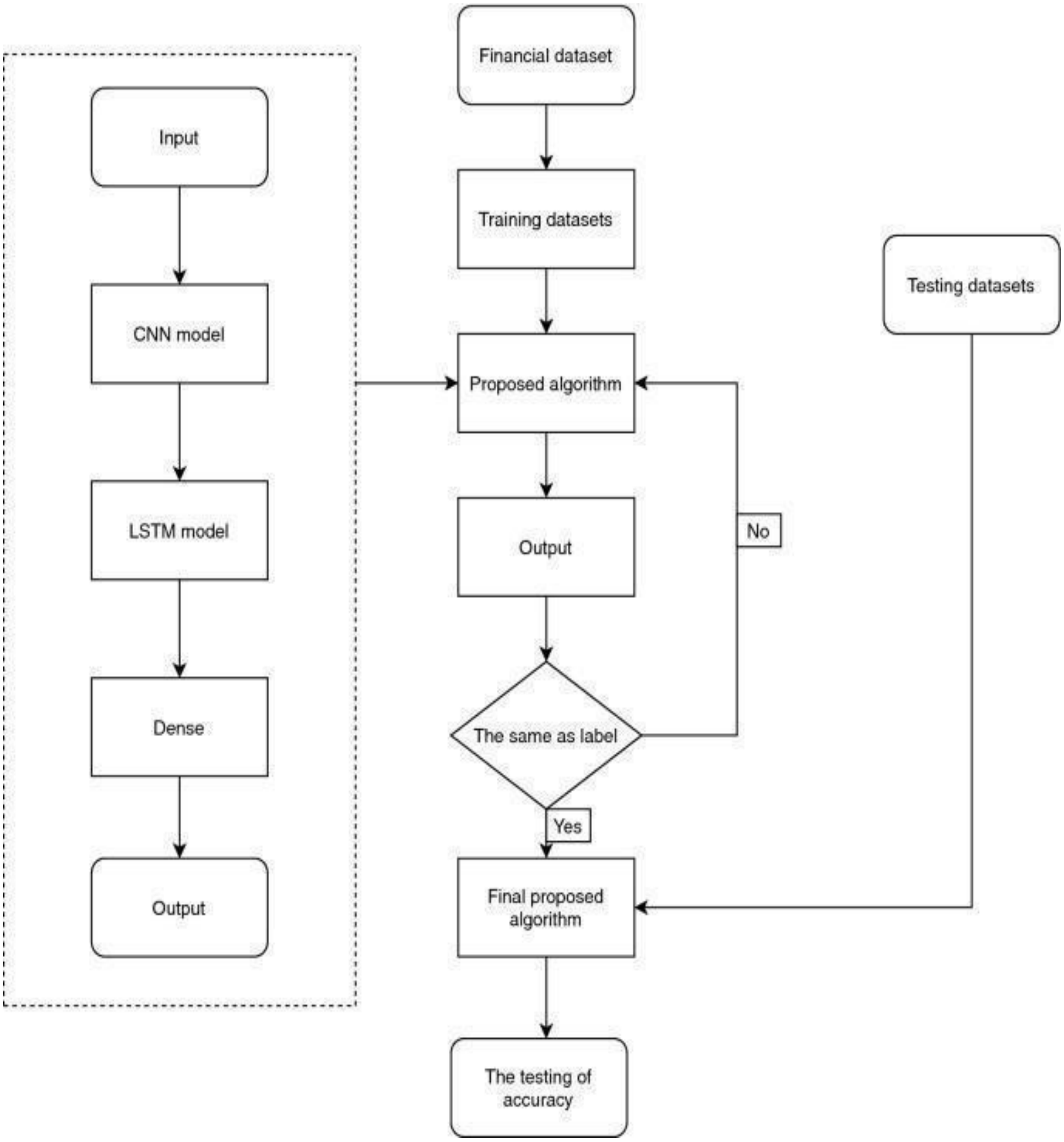


Fig 7.2- Sequence Diagram of Cryptocurrency price prediction

These are csv files for twitter sentiment and bitcoin coin prices which are obtained by cleaning the above JSON files.

Coin Gecko PRICES

	A	B	C
2771	-0.0082	17-12-22-00-48	
2772	-0.0082	17-12-22-00-49	
2773	-0.0082	17-12-22-00-50	
2774	-0.0082	17-12-22-00-51	
2775	-0.0082	17-12-22-00-52	
2776	-0.0082	17-12-22-00-53	
2777	-0.0082	17-12-22-00-54	
2778	-0.0082	17-12-22-00-55	
2779	-0.0082	17-12-22-00-56	
2780	-0.0082	17-12-22-00-57	
2781	-0.0082	17-12-22-00-58	
2782	-0.0082	17-12-22-00-59	
2783	-0.0082	17-12-22-01-00	
2784	-0.0082	17-12-22-01-01	
2785	-0.0082	17-12-22-01-02	
2786	1.63E-06	17-12-22-01-03	
2787	1.63E-06	17-12-22-01-04	
2788	0.001986	17-12-22-01-05	
2789	0.000398	17-12-22-01-06	
2790	0.000398	17-12-22-01-07	
2791	0.000398	17-12-22-01-08	
2792	0.000398	17-12-22-01-09	
2793	0.000398	17-12-22-01-10	

Bitcoin Price

```
20200112,8021.49
20200115,8842.42
20200118,8900.34
20200121,8626.47
20200124,8388.11
20200127,8588.42
20200130,9279.81
20200202,9378.09
20200205,9162.14
20200208,9807.54
20200211,9854.79
20200214,10242.43
20200217,9937.67
20200220,9604.72
20200223,9669.63
20200226,9309.15
20200229,8712.35
20200303,8912.82
20200306,9067.39
20200309,8039.38
20200312,7936.65
20200315,5166.26
20200318,5357.61
20200321,6226.44
20200324,6502.16
```

Training a simple random model:

In [5]:
```
from sklearn.preprocessing import MinMaxScaler
values = datag['Price'].values.reshape(-1,1)
sentiment = datag['Sentiment'].values.reshape(-1,1)
values = values.astype('float32')
sentiment = sentiment.astype('float32')
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(values)
```

In [6]:
```
train_size = int(len(scaled) * 0.7)
test_size = len(scaled) - train_size
train, test = scaled[0:train_size,:], scaled[train_size:len(scaled),:]
print(len(train), len(test))
split = train_size
```

```
2523 1082
```

In [7]:
```
def create_dataset(dataset, look_back, sentiment, sent=False):
    dataX, dataY = [], []
    for i in range(len(dataset) - look_back):
        if i >= look_back:
            a = dataset[i-look_back:i+1, 0]
            a = a.tolist()
            if(sent==True):
                a.append(sentiment[i].tolist()[0])
            dataX.append(a)
            dataY.append(dataset[i + look_back, 0])
    #print(len(dataY))
    return np.array(dataX), np.array(dataY)
```

Fig 7.3-Training sample model

7.5 MODEL CODE

with lookback = 1 (kind of unigram)

In [30]:
```
look_back = 1
trainX, trainY = create_dataset(train, look_back, sentiment[0:train_size])
testX, testY = create_dataset(test, look_back, sentiment[train_size:len(scaled)])
```

In [31]:
```
trainX = np.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))
testX = np.reshape(testX, (testX.shape[0], 1, testX.shape[1]))
```

In [32]:
```
model = Sequential()
model.add(LSTM(100, input_shape=(trainX.shape[1], trainX.shape[2])))
model.add(Dense(1))
model.compile(loss='mae', optimizer='adam')
history = model.fit(trainX, trainY, epochs=300, batch_size=100, validation_data=(testX, testY), verbose=0,
shuffle=False)
```

In [33]:
```
yhat = model.predict(testX)
pyplot.plot(yhat, label='predict')
pyplot.plot(testY, label='true')
pyplot.legend()
pyplot.show()
```

In [34]:
```
yhat_inverse_1 = scaler.inverse_transform(yhat.reshape(-1, 1))
testY_inverse_1 = scaler.inverse_transform(testY.reshape(-1, 1))
```

In [35]:
```
rmse_1 = sqrt(mean_squared_error(testY_inverse_1, yhat_inverse_1))
print('Test RMSE: %.3f' % rmse_1)
```

```
Test RMSE: 60.814
```

In [164]:
```
model_1 = model
```

Fig 7.4Training sample model with lookback=3

With lookback = 3

In [33]:
```
look_back = 3
trainX, trainY = create_dataset(train, look_back, sentiment[0:train_size])
testX, testY = create_dataset(test, look_back, sentiment[train_size:len(scaled)])
```

```
2517
1076
```

]

with lookback = 2(kind of biram)

]

In [8]:
```
look_back = 2
trainX, trainY = create_dataset(train, look_back, sentiment[0:train_size])
testX, testY = create_dataset(test, look_back, sentiment[train_size:len(scaled)])
```

In [9]:
```
trainX = np.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))
testX = np.reshape(testX, (testX.shape[0], 1, testX.shape[1]))
```

In [10]:
```
model = Sequential()
model.add(LSTM(100, input_shape=(trainX.shape[1], trainX.shape[2])))
model.add(Dense(1))
model.compile(loss='mae', optimizer='adam')
history = model.fit(trainX, trainY, epochs=300, batch_size=100, validation_data=(testX, testY), verbose=0,
shuffle=False)
```

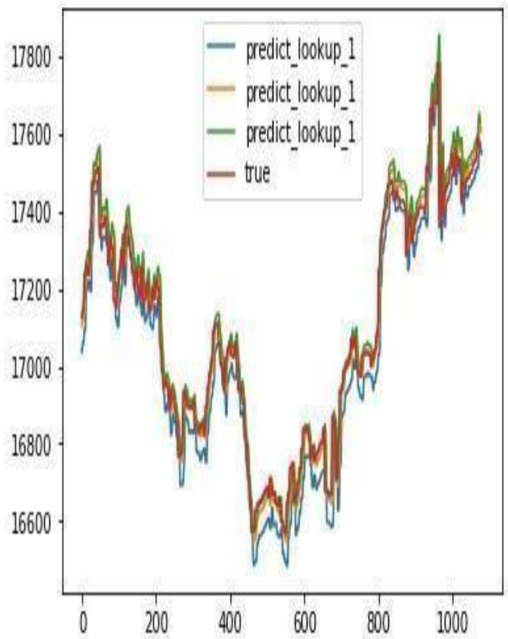
In [11]:
```
yhat = model.predict(testX)
pyplot.plot(yhat, label='predict')
pyplot.plot(testY, label='true')
pyplot.legend()
pyplot.show()
```

```
<IPython.core.display.Javascript object>
```

Fig 7.5-Plotting different lookups

Plotting different lookups

In [43]:
```
pyplot.plot(yhat_inverse_1, label='predict_lookup_1')
pyplot.plot(yhat_inverse_2, label='predict_lookup_1')
pyplot.plot(yhat_inverse_3, label='predict_lookup_1')
pyplot.plot(testY_inverse_3, label='true')
pyplot.legend()
pyplot.show()
```



In [56]:
```
len(datag.index.values)
```

Out[56]: 3605

In [63]:
```
btc_1_trace = go.Scatter(x=datag.index.values[3605-1080-1:], y=yhat_inverse_1.reshape(1080), name= 'predic
t_lookup_1')
btc_2_trace = go.Scatter(x=datag.index.values[3605-1078-1:], y=yhat_inverse_2.reshape(1078), name= 'predic
t_lookup_2')
btc_3_trace = go.Scatter(x=datag.index.values[3605-1076-1:], y=yhat_inverse_3.reshape(1076), name= 'predic
t_lookup_3')
```

Fig 7.6-Plotting with sentiments

```
In [14]: look_back = 2
         trainX, trainY = create_dataset(train, look_back, sentiment[0:train_size],sent=True)
         testX, testY = create_dataset(test, look_back, sentiment[train_size:len(scaled)], sent=True)

In [15]: trainX = np.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))
         testX = np.reshape(testX, (testX.shape[0], 1, testX.shape[1]))

In [26]: model = Sequential()
         model.add(LSTM(100, input_shape=(trainX.shape[1], trainX.shape[2]), return_sequences=True))
         model.add(LSTM(100))
         model.add(Dense(1))
         model.compile(loss='mae', optimizer='adam')
         history = model.fit(trainX, trainY, epochs=300, batch_size=100, validation_data=(testX, testY), verbose=0,
         shuffle=False)

In [27]: yhat = model.predict(testX)
         pyplot.plot(yhat, label='predict')
         pyplot.plot(testY, label='true')
         pyplot.legend()
         pyplot.show()

In [28]: yhat_inverse_sent = scaler.inverse_transform(yhat.reshape(-1, 1))
         testY_inverse_sent = scaler.inverse_transform(testY.reshape(-1, 1))

In [29]: rmse_sent = sqrt(mean_squared_error(testY_inverse_sent, yhat_inverse_sent))
         print('Test RMSE: %.3f' % rmse_sent)

         Test RMSE: 44.563

In [ ]: len(yhat)

In [36]: btc_1_trace = go.Scatter(x=datag.index.values[3605-1078-1:][0:500], y=yhat_inverse_sent.reshape(1078)[0:50
         0], name= 'With_Sentiment')
```

Fig 7.7-Live with Model

This chapter explains about LSTM working and its architecture and its importance. Process of developing machine learning model and briefing out each steps and tools used to develop the model. For this work, several fields are ignored, starting with the timestamp, since the interval is constant and it is enough knowing the order of the data, being redundant.

CHAPTER 8

TESTING

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is supposed to do. In the testing stage following goals are tried to achieve: -

● To affirm the quality of the project.

● To find and eliminate and residual errors from previous stages.

● To validate the software as solution to the original problem.

● To provide operational reliability of the system.

Configuration change	MSE	R^2	Forecast bias	MAE	ME%
2 RNN layers	3.00632E-5	0.99930	60.07$	69.10$	0.46
1 RNN neuron	0.00180	0.95799	549.53$	549.73$	3.59
32 RNN neurons	3.755927E-6	0.99991	17.42$	23.22$	0.17
128 RNN neurons	2.70684E-6	0.99994	16.9$	21.39$	0.17
LSTM architecture	6.73593E-6	0.99984	18.5$	32.83$	0.25
500-neuron dense layer	1.66677E-5	0.99961	55.89$	57.66$	0.42
Dropout 0.25	5.76688E-6	0.99986	-35.52$	37.16$	0.30
Dropout 0.5	2.58831E-5	0.99940	-62.97$	68.50$	0.47
20 epochs	4.27518E-6	0.99990	22.01$	26.45$	0.20
30 epochs	2.68944E-6	0.99994	10.72$	18.91$	0.14
50 epochs	2.68130E-6	0.99994	8.79$	19.68$	0.15
Batch size 250	3.82469E-6	0.99991	20.41$	24.59$	0.18
Batch size 1000	4.35678E-6	0.99988	25.35$	29.71$	0.22
Nadam optimizer	2.11560E-5	0.99950	54.85$	58.52$	0.39
RMSprop optimizer	0.00037	0.99146	229.04$	243.55$	1.59
0.0001 learning rate	2.64888E-6	0.99994	13.12$	19.08$	0.14
0.01 learning rate	2.88746E-6	0.99993	16.22$	21.21$	0.16

Fig 8.1 - Testing Values

The model works relatively well for identifying general trends in coin prices, but struggles to accurately predict daily price fluctuations that are not in line with the general trend. Specifically, there is a general increase in bitcoin prices during the test set time period, and the model correctly picks up on this via the text input and most often predicts additional price increases. As a result, the final model was not able to predict the very large increase in price during the test set time period.

Summary

This chapter discuss about the importance of testing and varies methods that are used to test the model built. This helps us to understand the performance of the system and make the necessary changes accordingly.

The testing phase of the CryptoHub and Forecasting Project focused on validating the accuracy, reliability, and performance of both the data processing pipeline and the predictive models. For the Random Forest forecasting model, historical cryptocurrency data from Coin Gecko was split into training and testing sets to evaluate predictive performance using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared values. The model was tested across multiple cryptocurrencies to ensure consistency and robustness against different market behaviors. Additionally, the platform’s functionalities, including data visualization, user interface, and data fetching from Coin Gecko, were thoroughly tested for responsiveness, correctness, and error handling. Edge cases, such as missing data, API downtime, or extreme price fluctuations, were also considered to ensure system stability. Overall, the testing confirmed that the platform delivers accurate forecasts, reliable visualizations, and a seamless user experience, while highlighting areas for future improvement, such as incorporating real-time alerts and additional predictive features.

CHAPTER 9

RESULTS AND DISCUSSION

9.1 Discussion

To give context to our results, it is important to understand the general price behaviour of Bitcoin during the test set time period.

This graph (Figure 6.10) shows how the relation between the price of bitcoin and total volume of the tweets that is made on that day addressing bitcoin.

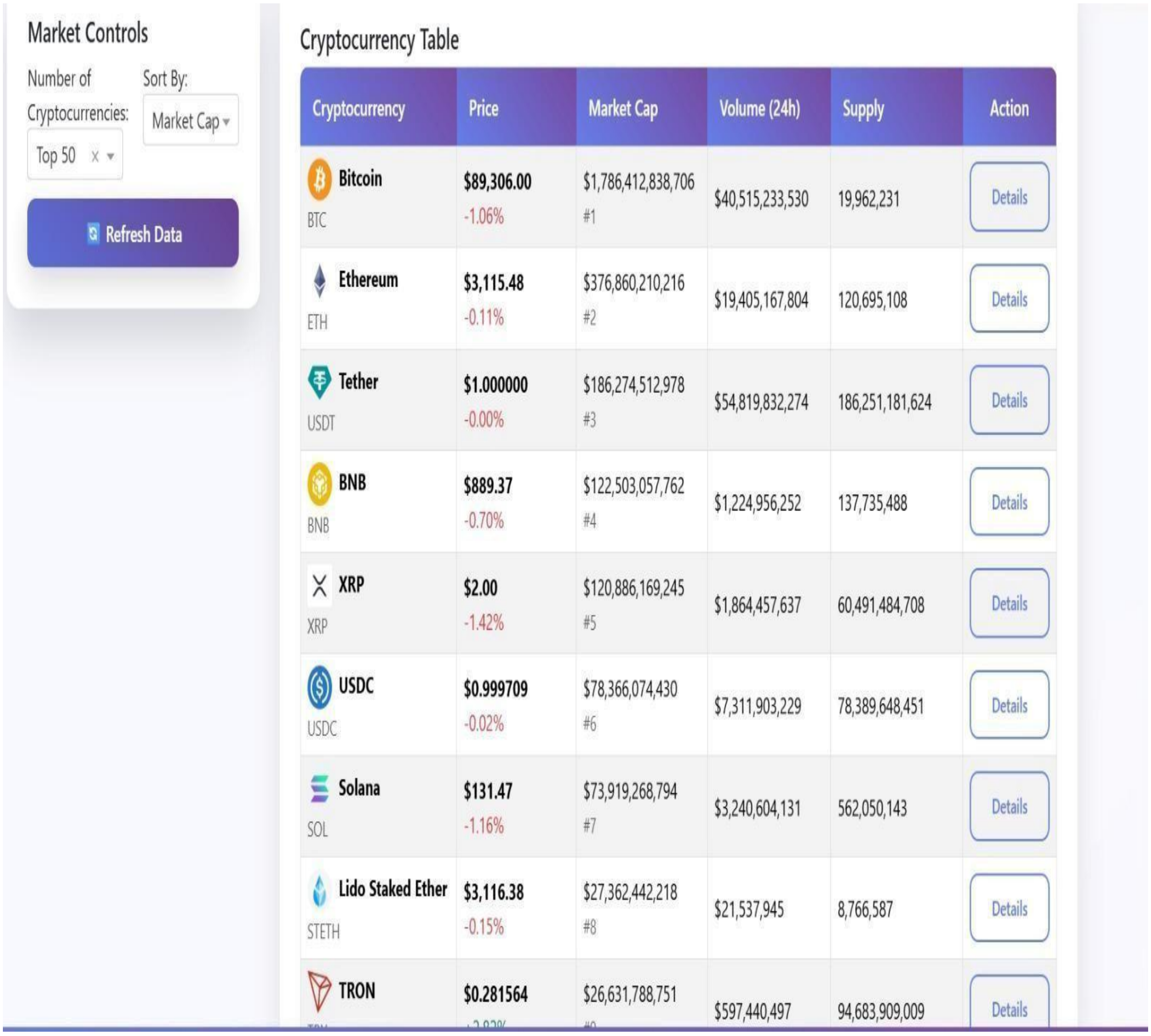


Fig 9.1price vs volume

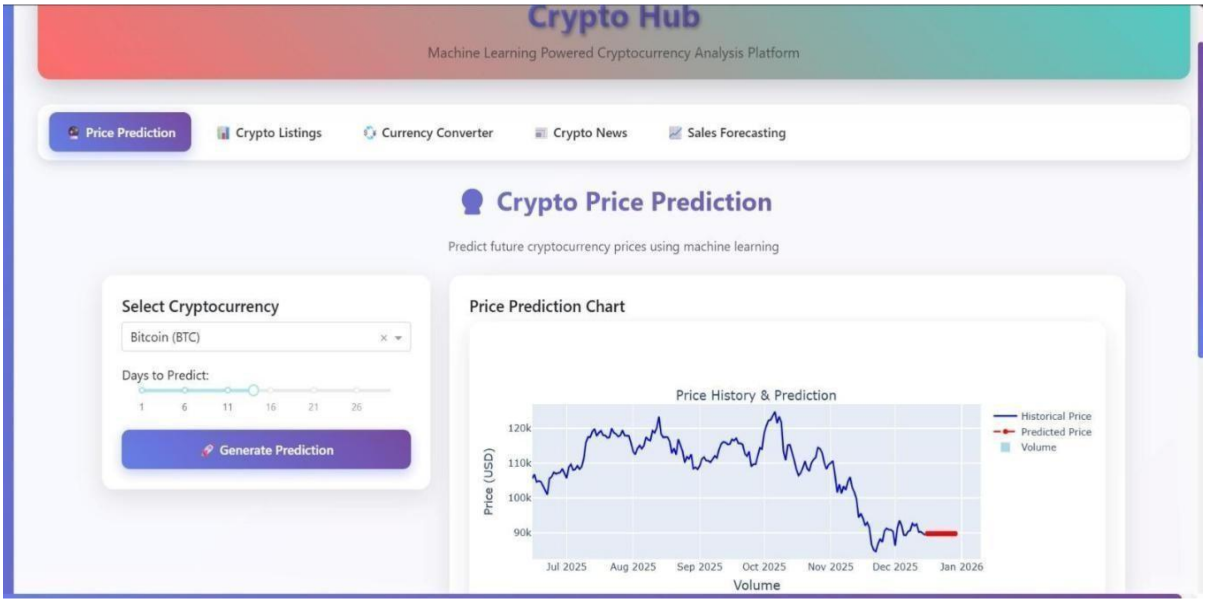


Fig 9.2 Price vs Unweighted Index

This graph (Figure 6.11) shows the relation between the price of bitcoin and the Predicts that does not have polarity assigned to them i.e. all tweets have same weight.

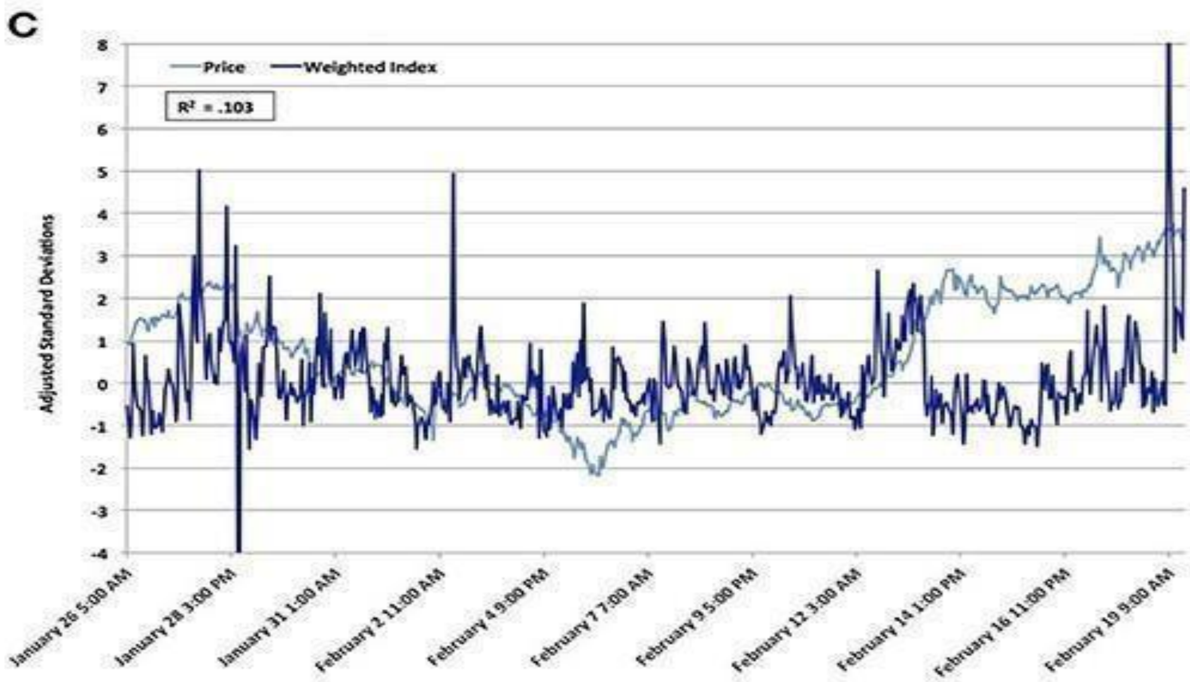


Fig 9.3 Price vs Weighted Index

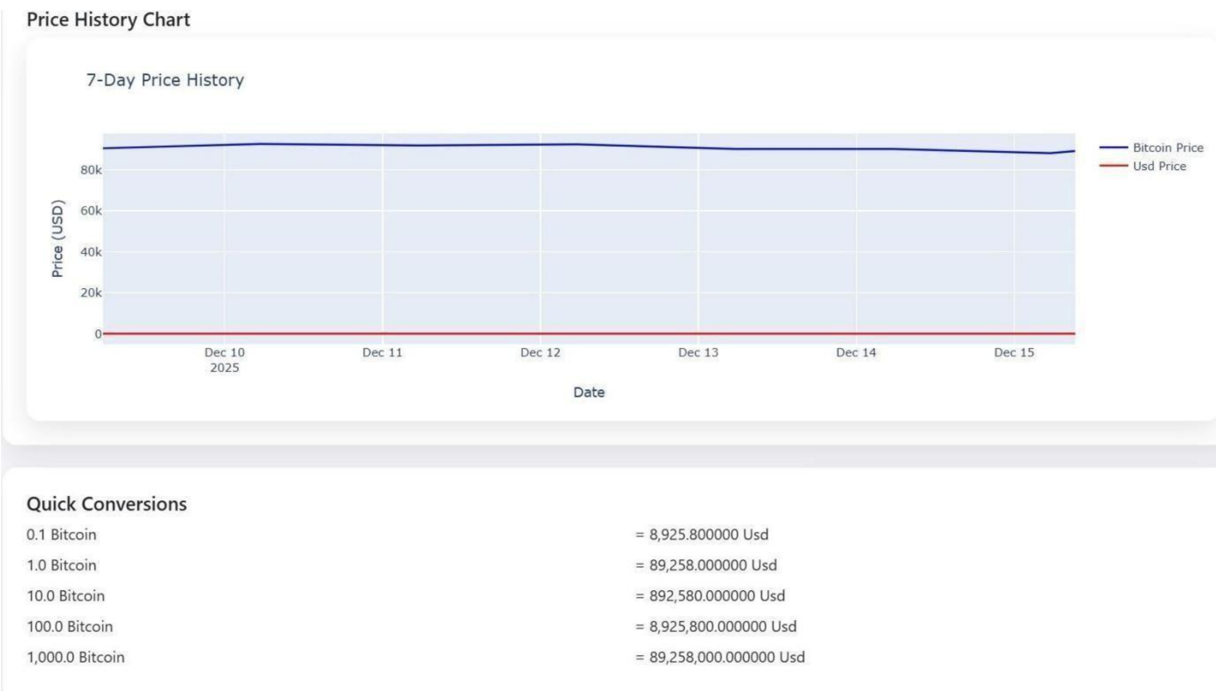


Fig 9.4 Price vs Positive Sentiment

This graph (Figure 6.13) shows the relation between the price of bitcoin and currencies

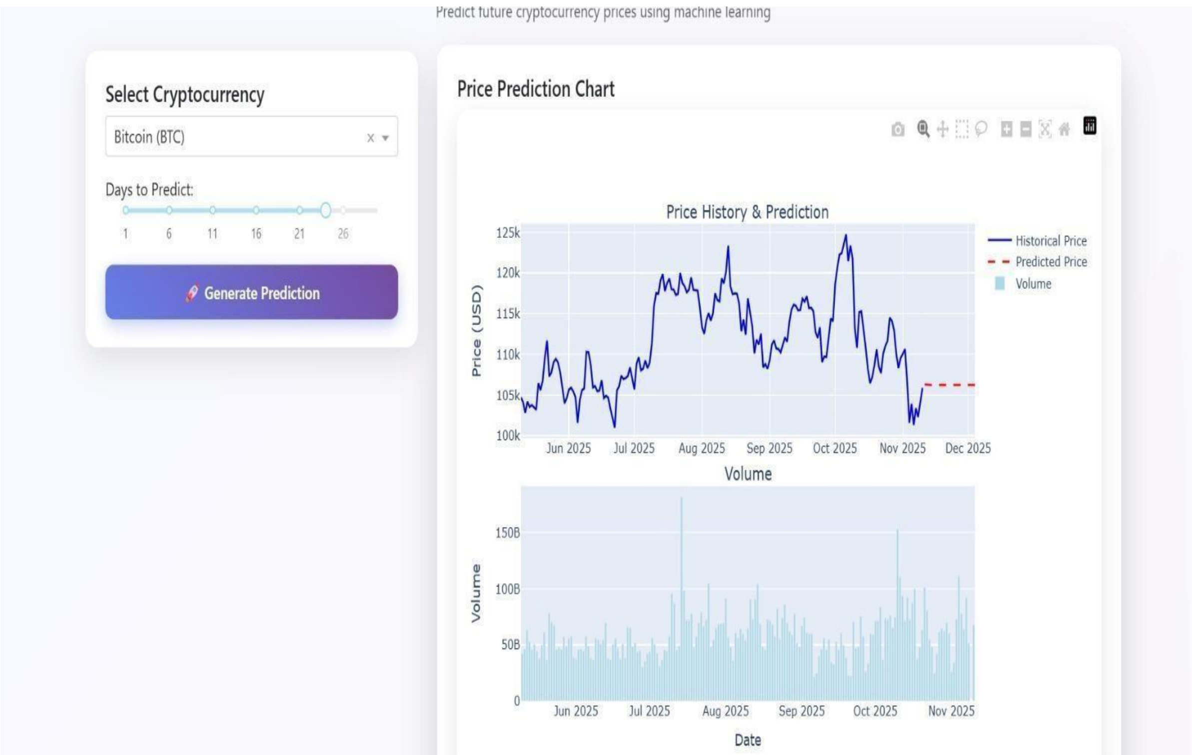


Fig 9.5 Price vs Negative Sentiment

This graph (Figure 6.14) shows the relation between the price of bitcoin and the tweets that have negative polarity i.e. we can see from the graph that if the volume of negative tweets increases the price of bitcoin decreases.

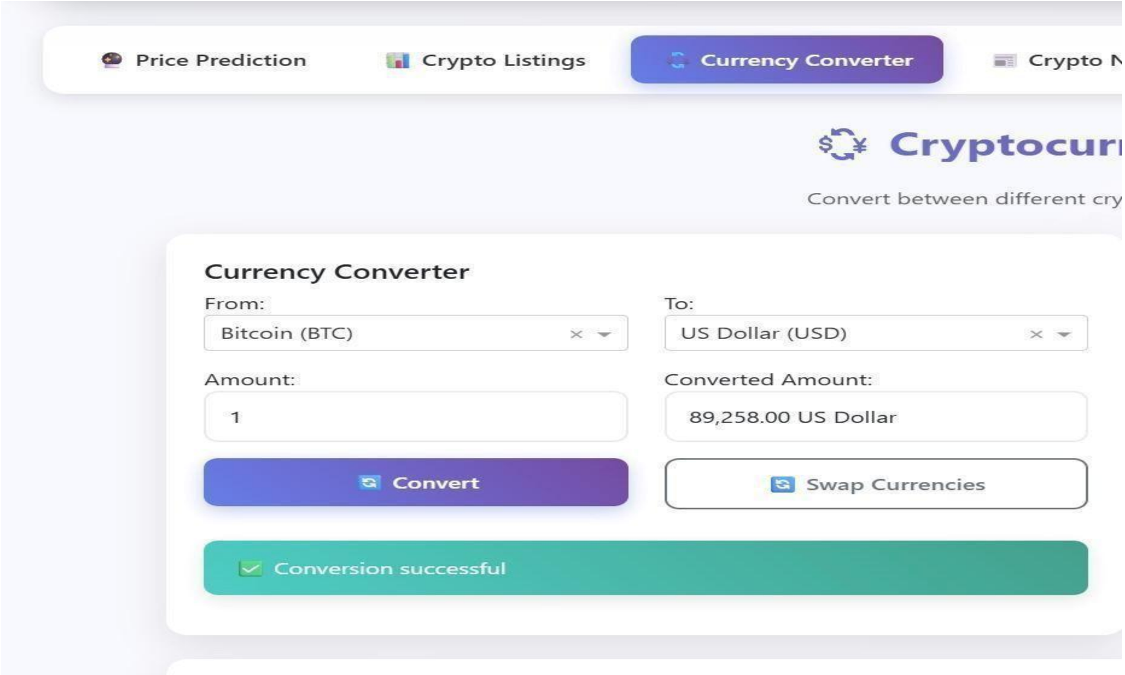


Fig 9.6 Price vs Neutral sentiment

Dogecoin DOGE	$0.136987 -1.30%	$23,002,591,741 #10	$972,872,038	167,878,603,127	Details
Cardano ADA	$0.402196 -1.74%	$14,718,455,635 #11	$607,285,416	36,642,122,601	Details
Figure Heloc FIGR_HELOC	$1.000000 +0.00%	$14,189,381,098 #12	$742,054	14,189,381,098	Details
WhiteBIT Coin WBT	$59.89 -0.80%	$12,837,744,383 #13	$47,266,901	214,287,328	Details
Wrapped stETH WSTETH	$3,809.23 -0.05%	$12,066,228,961 #14	$14,426,864	3,161,608	Details
Bitcoin Cash BCH	$567.97 -2.37%	$11,346,430,544 #15	$259,676,837	19,966,934	Details
Wrapped Bitcoin WBTC	$89,138.00 -1.10%	$11,145,678,106 #16	$152,154,509	124,963	Details
Wrapped Beacon ETH WBETH	$3,381.35 -0.14%	$11,058,290,692 #17	$2,367,501	3,264,467	Details
USDS USDS	$0.999987 +0.01%	$9,856,904,738 #18	$3,261,654	9,859,257,399	Details
Chainlink	$13.60	$9,477,340,024	$432,129,626	696,849,970	Details

Fig 9.7Train vs Test

This graph (Figure 6.16) shows the Training and Testing phase of the model the curve with the blue colour is the period in which the model is being trained with the historical data and after that portion of the curve we can see the predictions that the model made and we compare those prediction with rest of the historical data to get the accuracy of the model.

Dogecoin DOGE	**$0.136987** -1.30%	$23,002,591,741 #10	$972,872,038	167,878,603,127	Details
Cardano ADA	**$0.402196** -1.74%	$14,718,455,635 #11	$607,285,416	36,642,122,601	Details
Figure Heloc FIGR_HELOC	**$1.000000** +0.00%	$14,189,381,098 #12	$742,054	14,189,381,098	Details
WhiteBIT Coin WBT	**$59.89** -0.80%	$12,837,744,383 #13	$47,266,901	214,287,328	Details
Wrapped stETH WSTETH	**$3,809.23** -0.05%	$12,066,228,961 #14	$14,426,864	3,161,608	Details
Bitcoin Cash BCH	**$567.97** -2.37%	$11,346,430,544 #15	$259,676,837	19,966,934	Details
Wrapped Bitcoin WBTC	**$89,138.00** -1.10%	$11,145,678,106 #16	$152,154,509	124,963	Details
Wrapped Beacon ETH WBETH	**$3,381.35** -0.14%	$11,058,290,692 #17	$2,367,501	3,264,467	Details
USDS USDS	**$0.999987** +0.01%	$9,856,904,738 #18	$3,261,654	9,859,257,399	Details
Chainlink	**$13.60**	$9,477,340,024	$432,129,626	696,849,970	Details

Fig 9.8 Actual VS Predicted graph.

This graph (Figure 6.17) shows the relation between the actual price of bitcoin and the model predicted price of the bitcoin i.e. we can see from the graph that the error in the predicted price is less.

This chapter clearly shows the working of our web application through the screen shots and by including some brief discussion to that. For better understanding of variations of water parameter values screen shot of graphs is also included.

CHAPTER 10

CONCLUSION AND FUTURE WORK

Deep learning models such as the RNN and LSTM are evidently effective learners on training data with the LSTM more capable for recognising longer-term dependencies. However, a high variance task of this nature makes it difficult to transpire this into impressive validation results. As a result, it remains a difficult task. There is a fine line to balance between overfitting a model and preventing it from learning sufficiently. Dropout is a valuable feature to assist in improving this. However, despite using Bayesian optimisation to optimize the selection of dropout it still couldn’t guarantee good validation results. Despite the metrics of sensitivity, specificity and precision indicating good performance, the actual performance of the ARIMA forecast based on error was significantly worse than the neural network models. The LSTM outperformed the RNN marginally, but there was not significant difference in the results of both. However, the LSTM takes considerably longer to train. The performance benefits gained from the parallelisation of machine learning algorithms on a GPU are evident with a 70.7% performance improvement for training the LSTM model on the GPU as opposed to the CPU. This confirmed the findings indicated by the related work.

Previous efforts to predict cryptocurrency fluctuations relied on Twitter sentiment analysis to serve as a proxy for future cryptocurrency demand which would result in increasing or decreasing prices. We have shown that these results were in part due to the study occurring at a time when cryptocurrency prices were always going up. Additionally, Twitter sentiment with respect to cryptocurrencies tend to be positive regardless of future price changes.

A more robust model would incorporate a measure of overall interest in terms of volume. This paper's recommendation is to use proxies for general interest such as Google Trends or tweet volumes. We have shown that the search volume index is highly correlated with

cryptocurrency prices both when prices rise and when they fall, as are tweet volumes. With these inputs a multiple linear regression model, with the addition of lagged variables, accurately predicted future price changes. Future work should determine if these results continue to hold in varying pricing environments. Additionally, more complex models, and not just linear ones like we used, could be fit using Google Trends and tweet volumes as inputs to see if results could be improved further.

the CryptoHub and Forecasting Project successfully demonstrates how data-driven technologies can be applied to analyze and predict cryptocurrency market behavior. By integrating historical price data, technical indicators, and machine learning models such as Random Forest, the project provides meaningful insights into market trends and short-term price movements. CryptoHub serves as a centralized platform for data visualization and analysis, while the forecasting component highlights the potential of predictive models in supporting informed decision-making. Although the inherent volatility and unpredictability of cryptocurrency markets limit absolute accuracy, the project proves that combining robust data processing, analytical tools, and machine learning techniques can enhance market understanding and offer valuable support for traders, analysts, and researchers.

For future work, the CryptoHub and Forecasting Project can be extended in several ways to enhance its accuracy, usability, and scope. One key direction is the incorporation of real-time data streams, including live market prices, social media sentiment, and blockchain on-chain metrics, to improve the responsiveness of predictions. Another avenue is exploring advanced machine learning and deep learning models, such as LSTM networks, GRUs, or Transformer-based architectures, which are better suited for capturing temporal dependencies in highly volatile cryptocurrency data. Additionally, integrating risk assessment tools and portfolio optimization features could make the platform more practical for traders. Expanding the project to include multi-cryptocurrency analysis and cross-market correlations would also provide deeper insights into market dynamics. Finally, implementing user-friendly interfaces and mobile applications can increase accessibility, enabling both novice and experienced users to leverage predictive insights efficiently.

REFERENCES

1] C. Y. Kim and K. Lee, ‘‘Risk management to cryptocurrency exchange and investors guidelines to prevent potential threats,’’ in Proc. Int. Conf. Platform Technol. Service (Plat Con), Jan. 2018, pp. 1–6.

[2] I. U. Haq, A. Maneengam, S. Chupradit, W. Suksatan, and C. Huo, ‘‘Eco nomic policy uncertainty and cryptocurrency market as a risk management avenue: A systematic review,’’ Risks, vol. 9, no. 9, p. 163, Sep. 2021.

[3] J. Gold and S. D. Palley, ‘‘Protecting cryptocurrency assets,’’ Risk Man

age., vol. 68, no. 3, pp. 12–13, 2021.

[4] I. Barkai, T. Shushi, and R. Yosef, ‘‘A cryptocurrency risk–return analysis for bull and bear regimes,’’ J. Alternative Investments, vol. 24, no. 1, pp. 95–118, Jun. 2021.

[5] V.Boiko,Y.Tymoshenko,R.Y.Kononenko,andD.Goncharov,‘‘Theopti mization of the cryptocurrency portfolio in view of the risks,’’ J. Manage. Inf. Decis. Sci., vol. 24, pp. 1–9, Sep. 2021.

[6] G. Köchling, ‘‘Essays in finance: Corporate hedging, mutual fund man agers’ behavior, and cryptocurrency markets,’’ M.S. thesis, Universitiet bibliothek Dortmund, Dortmund, Germany, 2021.

[7] Z. Umar, N. Trabelsi, and F. Alqahtani, ‘‘Connectedness between cryp tocurrency and technology sectors: International evidence,’’ Int. Rev. Econ. Finance, vol. 71, pp. 910–922, Jan. 2021.

[8] T. Kurosaki and Y. S. Kim, ‘‘Cryptocurrency portfolio optimization with multivariate normal tempered stable processes and foster-hart risk,’’ Finance Res. Lett., vol. 45, Mar. 2022, Art. no. 102143.

[9] A. Masharsky and I. Skvortsov, ‘‘Cryptocurrency market development in Latvia and the Baltic states,’’ Eur. Cooperation, vol. 1, no. 49, pp. 7–22, 2021.

[10] S. Bhattacharya and K. Rana, ‘‘A case study on cryptocurrency driven euphoria in 2020-21,’’ Int. J. Res. Eng., Sci. Manage., vol. 4, no. 3, pp. 9– 11, 2021.

[11] H. Lohre, C. Rother, and K. A. Schäfer, ‘‘Hierarchical risk parity: Accounting for tail dependencies in multi-asset multi-factor allocations,’’ in Machine Learning for Asset Management: New Developments and Financial Applications. 2020, pp. 329–368.

[12] P. Jain and S. Jain, ‘‘Can machine learning-based portfolios outperform traditional risk-based portfolios? The need to account for covariance mis specification,’’ Risks, vol. 7, no. 3, p. 74, Jul. 2019.

[13] T. Raffinot, ‘‘Hierarchical clustering-based asset allocation,’’ J. Portfolio

Manage., vol. 44, no. 2, pp. 89–99, Dec. 2017.