CRYPTOCURRENCY PRICE PREDICTION

Mini Project submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Under the esteemed guidance of

Mr. G Praveen kumar
Assistant Professor

By
M. Sathvika(20R11A05M0)
M. Deepika(20R11A05L9)
G. Haripriya(20R11A05L0)

Department of Computer Science and Engineering
Accredited by NBA

Geethanjali College of Engineering and Technology
(UGC Autonomous)
(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi)
Cheeryal (V), Keesara (M), Medchal.Dist.-501 301.

October-2023

Geethanjali College of Engineering & Technology

(UGC Autonomous)

(Affiliated to JNTUH, Approved by AICTE, New Delhi)
Cheeryal (V), Keesara(M), Medchal Dist.-501 301.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Accredited by NBA

CERTIFICATE

This is to certify that the B.Tech Mini Project report entitled **"CRYPTOCURRENCY PRICE PREDICTION"** is a bonafide work done by **M. Sathvika(20R11A05M0), M.Deepika(20R11A05L9), Haripriya(20R11A05L0)**, in partial fulfillment of the requirement of the award for the degree of Bachelor of Technology in "**Computer Science and Engineering**" from Jawaharlal Nehru Technological University, Hyderabad during the year 2023-2024.

Internal Guide
G PRAVEEN KUMAR
Assistant Professor

HOD - CSE
Dr A SreeLakshmi
Professor

External Examiner

Geethanjali College of Engineering & Technology

(UGC Autonomous)

(Affiliated to JNTUH Approved by AICTE, New Delhi)
Cheeryal (V), Keesara(M), Medchal Dist.-501 301.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Accredited by NBA

DECLARATION BY THE CANDIDATE

We, **M. Sathvika, M. Deepika, G. HariPriya,** bearing Roll Nos. **20R11A05M0, 20R11A05L9, 21R15A0L0,** hereby declare that the project report entitled **"CRYPTOCURRENCY PRICE PREDICTION"** is done under the guidance of **Mr. G PRAVEEN KUMAR**, **Assistant Professor**, Department of Computer Science and Engineering, Geethanjali College of Engineering and Technology, is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**.

This is a record of bonafide work carried out by me/us in GEETHANJALI COLLEGE OF ENGNEERING AND TECHNOLOGY_ and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

M. Sathvika(20R11A05M0)
M. Deepika(20R11A05L9)
G. Haripriya(20R11A05L0)
Department of CSE
Geethanjali College of Engineering and Technology,
Cheeryal.

ACKNOWLEDGEMENT

We are greatly indebted to the Management of Geethanjali College of Engineering and Technology, Cheeryal, Hyderabad, for providing us the necessary facilities to successfully carry out this mini project work titled "CRYPTOCURRENCY PRICE PREDICTION". Firstly, we thank and express our solicit gratitude to Professor Dr. A. SREE LAKSHMI, CSE department, Geethanjali College of Engineering and Technology for her invaluable help and support which helped us a lot in successfully completing our project.

Moreover, we also express our gratitude to G PRAVEEN KUMAR, Assistant Professor, Geethanjali College of Engineering and Technology, our guide and patron, for his continued support throughout our endeavor to make our project successfully done.

We would like to express our sincere gratitude to our Principal Dr. S. UDAYA KUMAR for providing the necessary infrastructure to completeour project.

We convey our gratitude to our Chairman, Mr. G. RAVINDER REDDY, for his invaluable support and encouragement for propelling our innovation forward. Finally, we would like to express our heartfelt gratitude to our parents and all our peers who were very supportive and for their encouragement to achieve our goals.

M. Sathvika (20R11A05M0),

M. Deepika(20R11A05L9),

G. Haripriya(20R11A05L0)

ABSTRACT

This project develops an end-to-end cryptocurrency price forecasting system using deep learning models. Historical price data is collected from exchanges and stored in a SQL database after prepossessing. For price prediction, LSTM and GRU models are built to capture temporal dependencies and 1D CNNs are used for feature learning from raw price data.

These models are ensemble for improved performance. Technical indicators are engineered as additional predictive features. Trading strategies are formed based on model price forecasts, and optimized through back-testing frameworks. The system is containerized for deployment on cloud platforms along with model monitoring and periodic retraining pipelines. This systematic process leverages recent advances in deep neural networks to create an accurate and robust cryptocurrency price prediction system.

LIST OF FIGURES

LIST OF SCREENSHOTS

LIST OF TABLES

TABLE OF CONTENTS

1.INTRODUCTION

1.1 ABOUT THE PROJECT

The project aims to develop a comprehensive end-to-end cryptocurrency price forecasting system using state-of-the-art deep learning models and data engineering techniques. This system is designed to leverage historical price data obtained from cryptocurrency exchanges, which is then preprocessed and stored within a SQL database for subsequent analysis and prediction. The core of the project revolves around the utilization of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models, two powerful recurrent neural network architectures. These models are employed to capture the intricate temporal dependencies inherent in cryptocurrency price data. Additionally, 1D Convolutional Neural Networks (CNNs) are employed to extract meaningful features from the raw price data, further enhancingthe accuracy of the price predictions.

To further enhance the prediction capabilities, technical indicators are engineered as additional predictive features. These indicators are likely derived from factors such as trading volume, volatility, and various other market-specific metrics. Incorporating these indicators adds a layer of sophistication to the prediction models, allowingthem to account for a broader range of factors that influence cryptocurrency prices.

Beyond merely forecasting cryptocurrency prices, the project takes a holistic approach by incorporating trading strategies based on the model-generated price forecasts. These strategies are meticulously designed and optimized through back-testing frameworks, ensuring that they are not only effective but also robust in different market conditions. This aspect of the project demonstrates a practical application of the price forecasts, potentially leading to profitable trading opportunities. Furthermore, the system is containerized, making it easily deployable oncloud platforms. This containerization facilitates scalability and accessibility, enabling users toaccess the prediction system remotely. Model monitoring and periodic retraining pipelines arealso incorporated into the deployment process, ensuring that the system remains up-to-date andaccurate over time.

In summary, this project leverages recent advancements in deep neural networks to create a sophisticated and reliable cryptocurrency price prediction system. By combining LSTM and GRU models, 1D CNNs, technical indicators, and trading strategies, the system aims to provide accurate forecasts and actionable insights for cryptocurrency traders and investors

1.2 OBJECTIVE

The primary objective of this project is to develop a comprehensive cryptocurrency price forecasting system that utilizes cutting-edge deep learning models and data-driven techniques to provide accurate and actionable predictions. The key objectives of the project can be outlined as follows:

1.Data Collection and Preprocessing: Gather historical cryptocurrency price data from exchanges, preprocess it to remove noise and anomalies, and store it in a structured SQL database.

2.Model Development: Build LSTM and GRU models to capture temporal dependencies in the data and 1D CNNs to extract relevant features from raw price data. These models should be capable of generating accurate price forecasts.

3.Ensemble Modeling: Combine multiple models for improved forecasting accuracy. Ensemble methods can enhance prediction robustness by aggregating the outputs of individual models.

4.Feature Engineering: Engineer technical indicators from the cryptocurrency data to serve as additional predictive features, augmenting the model's understanding of market dynamics.

5.Trading Strategy Formulation: Develop trading strategies based on the model-generated price forecasts. These strategies should be optimized through back-testing to ensure they are effective and reliable.

6. Containerization and Deployment: Containerize the entire system for easy deployment on cloud platforms, making it accessible to users. Ensure that deployment includes model monitoring and periodic retraining pipelines to maintain accuracy over time.

7. Accuracy and Robustness: Strive to create a system that delivers accurate price predictions and trading strategies that perform well in various market conditions, ultimately benefiting cryptocurrency traders and investors.

By achieving these objectives, the project aims to contribute to the development of a robust and practical cryptocurrency price prediction system that leverages deep learning and advanced data analytics techniques.

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The Random Forest will make use of decision trees to understand how these factors affected bitcoin prices in the past. Simply put, a decision tree is a flowchart-like mapping of inputs and outputs.

The patterns from these decision trees are what would dictate our price prediction model. The more data we can feed the Random Forest algorithm, the more opportunities it would have in finding new patterns and verifying existing patterns. It will then average the predictions of each tree to create a more reliable prediction. We will test how the predicted values fare out against the actual values. This means, we will save some data points from our dataset to test

2.2 PRPOSED SYSTEM

The proposed approach is based on the idea of not possessing all cryptocurrency data, simultaneously. In contrast, each cryptocurrency data is processed and handled independently and then the processed data from each cryptocurrency are merged and further processed for estimating the final prediction.

The rationale behind the proposed approach is to develop a learning model which is able to independently extract useful information from various cryptocurrency data and subsequently process this information for achieving accurate and reliable predictions. Each cryptocurrency data is utilized as input in a unique convolutional layer, which is followed by a pooling layer and a LSTM layer.

The proposed approach focuses on exploiting the ability of convolutional layers for extracting useful knowledge by learning the internal representation of each cryptocurrency, independently, as well as the effectiveness of LSTM layers for identifying short-term and long-term dependencies. Then, the output vectors of all LSTM layers are merged by a concatenate layer.

This layer is followed by a series of layers, which constitute the classical structure of a deep learning neural network i.e., a dense layer, a batch normalization layer, a dropout layer, a dense layer, a batch normalization layer, a dropout layer, and a final output layer of one neuro

2.3 FEASIBILITY STUDY

2.3.1 Details

1. Data Availability:

- Ensure that historical cryptocurrency price data is obtainable from various exchanges,including a sufficient historical data range for accurate forecasting.
- Verify the quality of data, including completeness and absence of significant data gaps orinaccuracies.
- Establish data preprocessing procedures, such as data cleaning, normalization, and featureextraction, in detail.

2. Model Development:

- Specify the architecture and hyperparameters of the LSTM, GRU, and 1D CNN models, including the number of layers, neurons, activation functions, and dropout rates.
- Define the training process, including the optimizer, learning rate, batch size, and convergence criteria.
- Detail the expected computational resources required for model training, such as GPUs or TPUs.

3. Ensemble Modeling:

- Explain the ensemble techniques to be used (e.g., stacking, averaging, or boosting).
- Clarify the process of combining individual model forecasts and how their weights are determined.
- Discuss any ensemble-specific hyperparameters and how they will be optimized.

4. Feature Engineering:

- List the specific technical indicators that will be engineered from the cryptocurrencyprice data (e.g., Moving Averages, RSI, MACD).
- Describe the mathematical formulas or algorithms used to compute these indicators.
- Address the potential challenges of incorporating external data sources or sentimentanalysis into the feature engineering process.

5. Containerization and Deployment:
    - Specify the containerization technology (e.g., Docker) and cloud platform (e.g., AWS, Azure, Google Cloud) to be used for deployment.
    - Detail the deployment process, including setting up scalable infrastructure, container orchestration, and monitoring systems.
    - Outline the frequency and process of model retraining and updates.

2.3.2. Impact on environment

1. Energy Efficiency:
    - The computational demands of training deep learning models can be significant. Efforts should be made to optimize model training to reduce energy consumption and carbon footprint.
    - Consider adopting energy-efficient hardware or cloud computing services that prioritize renewable energy sources.

2. Sustainability:
    - Emphasize sustainability in the choice of data centers and cloud providers, selecting those with a commitment to renewable energy usage.
    - Promote responsible data management practices to minimize data storage and processing requirements.

3. Remote Accessibility:
    - By deploying the system on the cloud, it enables remote access, reducing the need for physical infrastructure and commuting, which can contribute to reduced greenhouse gas emissions.

4. User Experience:
    - Design a user-friendly interface and provide educational resources to ensure that users can navigate the system easily.
    - A simpler user interface can reduce user errors and minimize the environmental impact of incorrect trading decisions.

5. Time Efficiency:
    - The system's accuracy and efficiency can lead to time savings for cryptocurrency traders, reducing the need for prolonged monitoring and decision-making. This can indirectly contribute to reduced stress and energy consumption associated with continuous monitoring.

6. Reduction in Physical Transactions:
    - If the system's trading strategies lead to more informed and less impulsive trading decisions, it may reduce the number of physical transactions, which often involve paperwork and transportation.

2.3.3. Safety

1. Data Security:
    - Impact: Ensuring the security of historical price data is crucial. Unauthorized access or data breaches could lead to data manipulation or theft, potentially compromising the accuracy and integrity of the forecasting system.

2. Network Security:
    - Impact:Protecting the network infrastructure used for data collection, model training, and system deployment is essential. Breaches or attacks could disrupt operations or lead to data leaks.

3. Information Security:
    - Impact:Safeguarding sensitive information related to trading strategies, user data, and model architecture is vital. Unauthorized access could expose proprietary information or trading strategies to competitors.

4. Privacy:
    - Impact: Privacy concerns arise when handling user data or personal information. Collecting, storing, and using data must comply with applicable privacy regulations to protect user rights and maintain trust.

5. Ethical Considerations:

- Impact:Ethical considerations regarding data usage and trading strategies may affect the project's reputation. Unethical practices could lead to public backlash and damage the project's credibility.

2.3.4. Impact on Various Areas

1. Data Management:

- Impact-Security measures must be implemented to protect historical price data. Access controls, encryption, and regular audits should be in place to safeguard data integrity.

2. Model Development:

- Impact-Ensuring the privacy of model architecture and training data is critical. Secure development practices should be followed to prevent model vulnerabilities.

3. Deployment and Maintenance:

- Impact-Network security should be maintained during system deployment. Regular security updates and patches are necessary to address vulnerabilities.

4. User Trust

- Impact - Privacy concerns and data security directly affect user trust. Transparent data usage policies and robust security measures can help build and maintain trust with users.

5. Legal and Regulatory Compliance:

- Impact:Failure to comply with data protection and privacy laws can lead to legal consequences. Adequate safeguards and compliance mechanisms should be in place.

6. Risk Management:

- Impact:Mitigating data and security risks is essential for the project's success. Risk assessment and contingency plans should be part of the project strategy.

7. Ethical Responsibility:

- Impact:Ethical considerations affect the project's reputation and public perception. Upholding ethical standards in data usage and trading strategies is important.

8. User Experience:

- Impact:Ensuring the safety and privacy of user data contributes to a positive user experience. Clear communication about data usage practices can enhance user confidence.

9. Competitive Advantage:

- Impact:Demonstrating a commitment to data security and ethical practices can be a competitive advantage, attracting users who prioritize safety and privacy.

2.3.5. Ethics

1. Data Privacy and Security

- User Data: Ensure that user data, including login credentials and personal information, is securely managed and not exposed in any form. Implement robust encryption protocols to protect user privacy.

- Data Storage: Safeguard the historical cryptocurrency price data stored in the SQL database. Protect it from unauthorized access and data breaches.

2. Fair and Unbiased Predictions

- Model Fairness: Ensure that the forecasting models are trained and validated to be as fair and unbiased as possible. Avoid any discriminatory biases that may result in unfair treatment of users or market participants.

3. Transparency

- Model Transparency: Provide transparency into how the forecasting models work, including the data sources, model architecture, and feature engineering techniques used. Users should have a clear understanding of how predictions are generated.

4. Avoiding Harm

- Financial Harm: Acknowledge that cryptocurrency trading carries financial risks. Ensure that users are informed about the inherent risks and do not make trading decisions solely based on model predictions. Emphasize responsible trading practices.
- Security: Prioritize the security of the application to prevent any vulnerabilities that could be exploited by malicious actors to harm users or the system.

5. Legal and Regulatory Compliance

- Data Regulations: Comply with data protection regulations, such as GDPR, to protect user data and ensure that data is collected and processed legally and ethically.
- Market Regulations: Be aware of and comply with any legal regulations and requirements related to cryptocurrency trading and financial markets.

6. User Consent

- Informed Consent: Ensure that users provide informed consent for using the application and agree to its terms and conditions, including data usage and potential financial risks.

7. Ethical Trading Strategies

- Trading Fairly: If the system incorporates trading strategies, ensure that these strategies are ethical and do not engage in any manipulative or fraudulent trading practices.

8. Continuous Monitoring and Responsiveness

- Ethical Oversight: Establish a mechanism for continuous monitoring and ethical oversight of the system's operations. Address any ethical concerns or issues promptly.

9. Accessibility and Inclusivity

- Equal Access: Ensure that the application is designed to be accessible to all users, regardless of their background or abilities.

2.3.6. COST

1. Cost of Development

- Data Acquisition: The cost associated with obtaining historical cryptocurrency price data from exchanges, which may involve data purchase or subscription fees.
- Infrastructure: Investment in hardware, software, and cloud computing resources for model development and testing. This includes the cost of high-performance GPUs if needed for training deep learning models.
- Personnel: The salaries and benefits of data scientists, machine learning engineers, and developers working on the project.
- Software Tools: Licensing fees for specialized software tools and libraries used in model development and data preprocessing.
- Training and Education: Costs related to training team members in deep learning techniques and cryptocurrency market dynamics.

2. Cost of Usage

- Cloud Hosting: Ongoing expenses for hosting the system on cloud platforms, which can include charges for computing resources, storage, and bandwidth.
- Data Maintenance: Costs associated with maintaining the SQL database, ensuring data integrity, and updating historical price data.
- Model Monitoring: Expenses for real-time or periodic monitoring of model performance and system health.

- User Support: Budget allocated for providing customer support and assistance to users of the system.

3. Cost of Maintenance

- Model Retraining: Costs related to periodically retraining machine learning models to keep them up-to-date and accurate.
- Bug Fixes and Updates: Funds allocated for addressing bugs, security vulnerabilities, and implementing system updates and improvements.
- Data Quality Assurance: Investment in data quality checks and ensuring the reliability of historical price data.

4. Cost Reduction Due to Implementation

- Increased Efficiency: Once the system is in production, it can streamline and automate various tasks, reducing the need for manual intervention and potentially lowering operational costs.
- Improved Decision-Making: More accurate price forecasts and trading strategies may lead to better investment decisions, potentially reducing the risk of financial losses.
- Reduced Data Costs: Over time, the cost of acquiring and maintaining cryptocurrency price data may decrease as the project establishes data sources and partnerships.
- Scalability: As the system becomes more efficient, it can handle a larger user base without proportionally increasing costs, resulting in economies of scale.
- Revenue Generation: If the project is monetized through subscription fees, licensing, or other revenue streams, it has the potential to offset development and operational costs, leading to cost reduction.

2.3.7. Type

1. Application (Web and/or Mobile App)

This project could result in the development of a web-based application and potentially a mobile app. Users would access the cryptocurrency price forecasts, trading strategies, and other relevant information through a user-friendly interface. This application could serve traders and investors in the cryptocurrency market, providing them with actionable insights.

2. Product

The project aims to create a marketable product, which is a cryptocurrency price forecasting and trading strategy system. This product could be licensed or offered on a subscription basis to users interested in making informed cryptocurrency trading decisions.

3. Research

While the primary goal is to develop a practical application, the project involves significant research aspects. This includes the research and implementation of advanced deep learning models (LSTM, GRU, 1D CNN), data engineering techniques, and the evaluation of trading strategies. The research component ensures that the system is built on the latest advancements in the field of deep learning and cryptocurrency market analysis.

4. Review

Although not explicitly mentioned, the project may also include a review or evaluation phase where the performance of the forecasting models and trading strategies is assessed. This review could involve analyzing historical predictions against actual market data to measure the accuracy and effectiveness of the system.

2.4. Scope

1. Technical Scope:

- Data Acquisition and Preprocessing: The project will encompass the collection of historical cryptocurrency price data from exchanges and the development of robust preprocessing techniques to clean and structure the data. This will ensure that high-quality data is available for analysis and modeling.

- Model Development: The project will include the implementation of advanced deep learning models, specifically LSTM, GRU, and 1D CNN models, to capture temporal dependencies in the cryptocurrency price data and extract meaningful features. These models will serve as the core components for price forecasting.

- Ensemble Modeling: Ensemble techniques will be applied to combine the outputs of the individual models, enhancing forecasting accuracy. The feasibility study has determined that this approach is technically viable.
- Feature Engineering: Technical indicators will be engineered from the cryptocurrency data to provide additional predictive features for the models, expanding their understanding of market dynamics.
- Containerization and Deployment: The project will involve containerizing the entire system for deployment on cloud platforms. This includes setting up model monitoring and periodic retraining pipelines to maintain system accuracy and relevance over time.

2. Economic Scope:

- Cost Analysis: The project will conduct a detailed cost analysis, including expenses related to data acquisition, model development, infrastructure, and ongoing maintenance. This analysis will help determine the economic viability of the project.
- Revenue Potential: The system's revenue potential will be explored, considering potential monetization strategies such as subscription fees or licensing agreements with users.
- Return on Investment (ROI): The ROI will be calculated based on the projected costs and revenues, providing insights into the economic feasibility of the project.

3. Operational Scope:

- Human Resources: The project will require skilled data scientists and machine learning engineers to work on data preprocessing, model development, and system deployment and maintenance.

- Data Management: Efficient data management practices will be implemented to handle the cryptocurrency price data, ensuring its availability and accuracy within the SQL database.
- Deployment and Maintenance: The project will encompass the deployment of the system on cloud platforms and the establishment of maintenance procedures, including automated monitoring and periodic retraining of models.
- User Adoption: Consideration will be given to user experience and usability to ensure that potential users, such as cryptocurrency traders and investors, can readily adopt and benefit from the system.

4. Legal and Ethical Scope:

- Data Usage and Privacy: The project will adhere to data usage regulations and address privacy concerns related to user data and trading strategies if they are incorporated into the system.

5. Risk Assessment Scope:

- Market Volatility: The project will acknowledge the inherent volatility of the cryptocurrency market, understanding that it can impact the accuracy of predictions and the effectiveness of trading strategies.
- Model Accuracy: The scope will encompass the need to communicate the limitations of deep learning models and the potential risks associated with making trading decisions solely based on model forecasts.
- Competition: The project will analyze the competitive landscape for cryptocurrency price prediction systems and define strategies for differentiation and value proposition.

2.5. SYSTEM CONFIGURATION

2.5.1. Hardware Requirements

● Operating system	:	Linux/windows/mac
● RAM	:	4 GB(min)
● Hard Disk	:	100 GB

2.5.2. Software Requirements

● Integrated IDE	:	Jupyter notebook
● Programming language	:	Python
● Machine learning libraries	:	scikit-learn, Tensorflow, keras, PyTorch
● Data Manipulation libraries	:	Numpy, Matplotlib, Pandas
● Database management	:	MySQL

3. LITERATURE OVERVIEW

In the rapidly evolving landscape of cryptocurrencies, this literature survey provides an overview of the research pertaining to the prediction of cryptocurrency prices, with a particular focus on Bitcoin. We begin by introducing the concept of cryptocurrency and Bitcoin's pivotal role within this digital financial realm, emphasizing the inherent price volatility and the significant implications for investors and traders. The survey highlights the increasing adoption of machine learning techniques in financial markets, particularly in predicting cryptocurrency prices, which often exhibit extreme fluctuations. The objectives of this literature review are to explore the evolution of research in cryptocurrency price prediction, understand the methodologies employed, and identify key findings.

Emphasize the global impact of cryptocurrencies and how they have disrupted traditional financial systems. Mention the diverse range of investors, from retail traders to institutional players, who are actively engaging in cryptocurrency markets. Highlight the significance of accurate price prediction for various stakeholders, including traders, investors, and policymakers. Additionally, discuss the emergence of Initial Coin Offerings (ICOs) and the proliferation of alternative cryptocurrencies beyond Bitcoin, and how they add complexity to the prediction landscape.

Our review delves into the historical context and the evolution of cryptocurrency price prediction research. We discuss early studies that laid the foundation for subsequent investigations in this domain. We also examine the deployment of machine learning techniques, encompassing neural networks, support vector machines, and regression models, to forecast cryptocurrency prices. The advantages and limitations of these methods are critically assessed. Furthermore, we explore the realm of sentiment analysis and its impact on cryptocurrency price dynamics, taking into account the methodologies employed to gauge public sentiment from various social media and news sources. Additionally, we consider research that leverages technical indicators and blockchain data, highlighting the role of blockchain technology in shaping cryptocurrency pricing.

When discussing machine learning techniques, you can delve deeper into specific algorithms used in cryptocurrency price prediction, such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), and long short-term memory (LSTM) networks. Explore any notable advancements in these methods. When covering sentiment analysis, elaborate on the use of Natural Language Processing (NLP) and text mining techniques for extracting sentiment from unstructured data. Discuss the challenges related to sentiment data quality, bias, and the evolving nature of social media. For technical indicators, provide examples of commonly used indicators, like Moving Averages, Relative Strength Index (RSI), and Bollinger Bands, and their significance in price prediction.

The survey focuses on recent advancements and the persistent challenges in cryptocurrency price prediction research. We provide insights into the application of deep learning models, particularly recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, which have demonstrated promise in enhancing prediction accuracy. The survey also sheds light on the difficulties researchers face, such as data quality issues, the prevalence of market manipulation, and the impacts of regulatory changes on cryptocurrency prices. Evaluation metrics commonly used to assess prediction model performance are introduced, including mean squared error (MSE) and root mean squared error (RMSE). Additionally, we explore the efficiency of the cryptocurrency market, evaluating the presence of arbitrage opportunities and market anomalies.

Recent Advancements and Challenges

In the section on deep learning, elaborate on the role of deep neural networks and how they handle complex patterns and dependencies in cryptocurrency price data. Mention specific deep learning architectures that have shown promise, such as Gated Recurrent Units (GRUs) or Transformer models like BERT (Bidirectional Encoder Representations from Transformers). When discussing challenges, expand on the issues of data scarcity and the presence of "noise" in cryptocurrency markets, where market manipulation and misinformation are common. Explore the potential impacts of regulatory developments and news events on cryptocurrency price volatility.

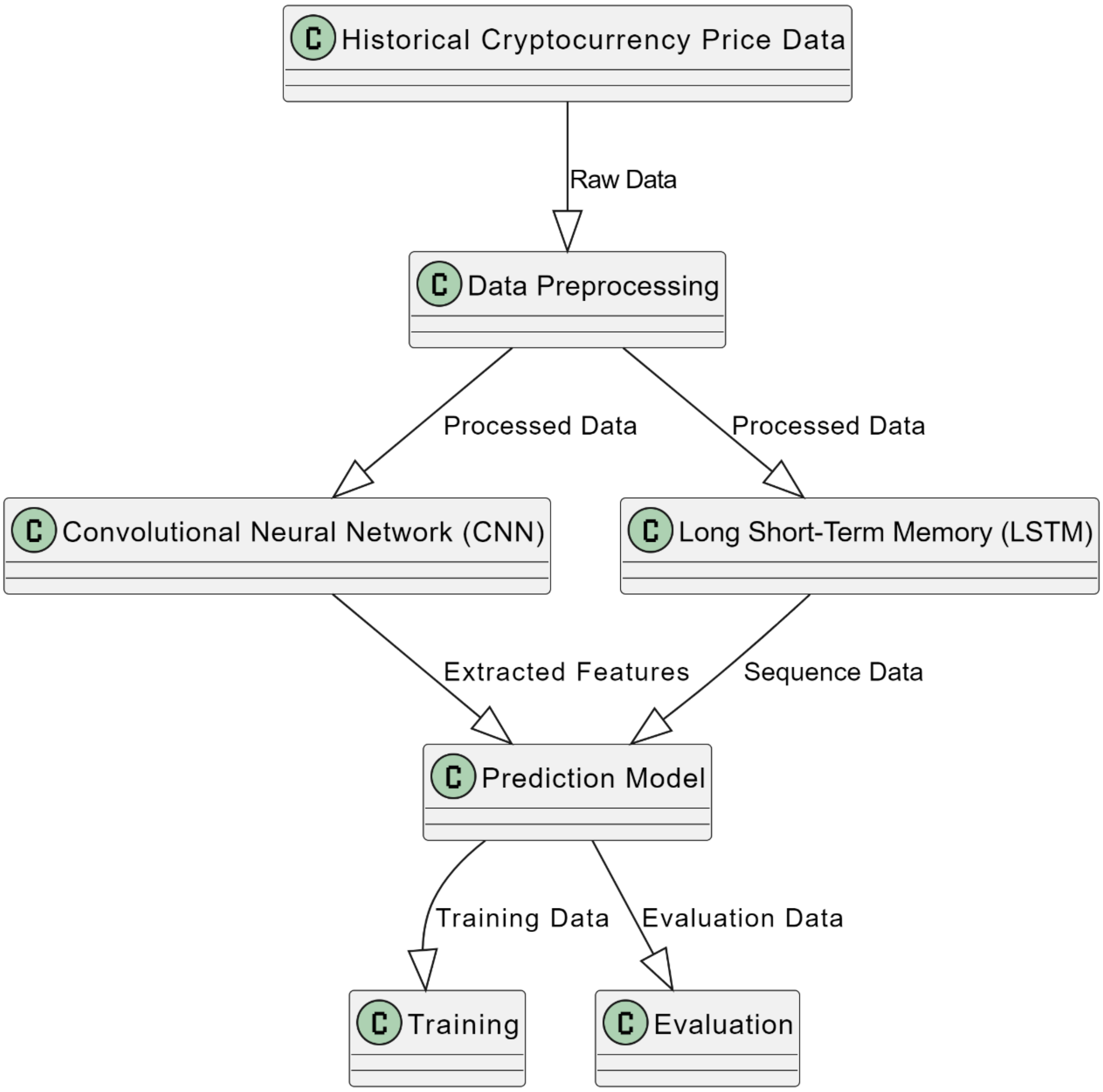
Future Directions and Conclusion

The final section of our literature survey outlines potential directions for future research in cryptocurrency price prediction. These include the integration of macroeconomic indicators, exploration of cross-currency relationships, and addressing real-time prediction challenges in the rapidly changing cryptocurrency market. In conclusion, we underscore the significance of ongoing research in this field, not only for investors seeking to navigate the volatile cryptocurrency market but also for the broader financial industry, which continues to grapple with the unique challenges posed by this emerging digital asset class. Ultimately, our comprehensive survey contributes to a deeper understanding of the past, present, and future of cryptocurrency price prediction.

In the segment on future research directions, provide examples of emerging trends, such as the use of blockchain data for prediction, decentralized finance (DeFi) applications, and the potential influence of central bank digital currencies (CBDCs) on cryptocurrency markets. Discuss how cross-currency relationships and the integration of data from traditional financial markets can enhance predictive models. For the conclusion, reiterate the importance of continued research in this field, emphasizing that cryptocurrency price prediction remains a dynamic area of study, as it is continually influenced by technological, regulatory, and economic developments.

4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE



In this diagram:

- "Historical Cryptocurrency Price Data" represents the data source.
- "Data Preprocessing" indicates the step where data is cleaned, normalized, and prepared for modeling.
- "Convolutional Neural Network (CNN)" and "Long Short-Term Memory (LSTM)" represent the two neural network components in your architecture.
- "Prediction Model" combines the output from CNN and LSTM.
- "Training" and "Evaluation" are the stages where the model is trained and tested.

4.2 UML DIAGRAMS

The Unified Modelling Language (UML) It is a way to visually represent the architecture, design, and implementation of complex software system they are again divided into subcategories:

1. Behavioral Diagrams: Behavioral Diagrams basically capture the dynamic aspect of a system. Dynamic aspects can be further described as the changing/moving parts of a system.
2. Structural Diagrams: The Structural Diagrams represent the static aspect of the system. These static parts are represented by classes, interface, object, components, and nodes.

- The underlying premise of UML is that no one diagram can capture the different elements of a System in its entirety. Hence, UML is made up of nine diagrams that can be used to model a System at different points of time in the software life cycle of a system.
- A software system can be said to have two distinct characteristics: a structural, "static" part and abehavioral, "dynamic" part. In addition to these two characteristics, an additional characteristic that a software system possesses is related to implementation.

4.2.1 Use Case Diagram

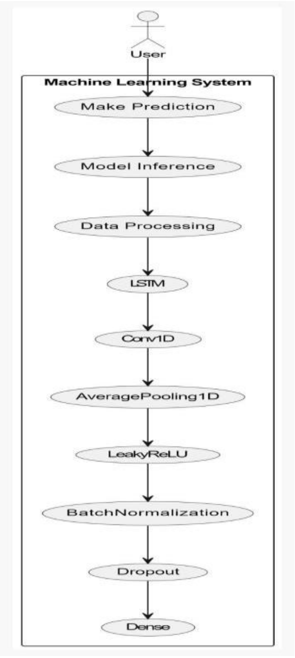


Figure 4.2.1 Use Case Diagram

A use case diagram represents the interactions between external actors and the system, outlining the various use cases or functionalities that the system provides. This system provides four main functionalities: training the prediction model with historical data, validating the model's performance, making real-time price predictions, and displaying these predictions to the user. Users can leverage this system to gain insights into cryptocurrency market trends and make informed investment decisions based on the model's predictions. With its user-friendly interface, the system empowers users to harness the power of deep learning to navigate the volatile cryptocurrency market.

4.2.2 Class Diagram

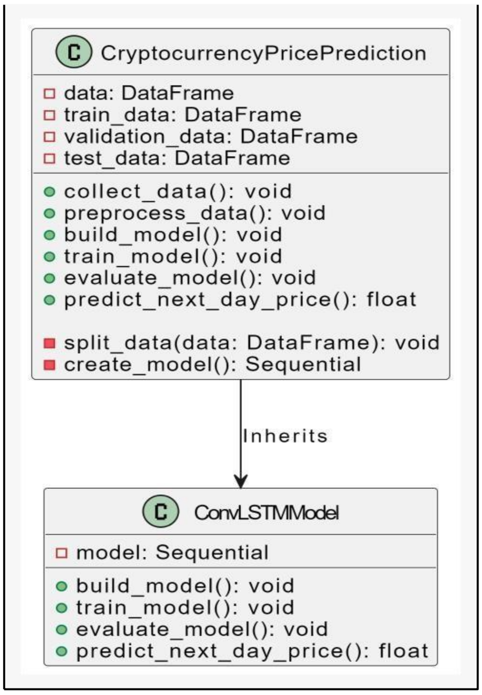


Figure 4.2.2. class diagram

A class diagram for cryptocurrency price prediction illustrates the essential classes and their relationships within the system. In this context, key classes might include "Cryptocurrency," "Exchange," "PredictionModel," and "User." These classes encapsulate the relevant attributes and behaviors, such as historical data storage, model training, and user interaction. The relationships between these classes depict how data flows and interactions occur within the system. Overall, the class diagram serves as a blueprint for implementing the cryptocurrency price prediction system, emphasizing the organization of data and functionality crucial for accurate and insightful predictions in the dynamic world of cryptocurrencies.

4.2.3 Sequence Diagram

A sequence diagram for cryptocurrency price prediction illustrates the dynamic interactions and order of events within the system. It shows how different components, such as data preprocessing, model training, and prediction, collaborate to provide users with price forecasts. Typically, the diagram depicts a sequential flow of activities, beginning with data ingestion and preprocessing, followed by the training of the prediction model. Afterward, it shows how users can request predictions, and the system responds with the forecasted cryptocurrency prices. In a cryptocurrency price prediction sequence diagram, the step-by-step depiction of these processes allows for a clear understanding of how the system operates, providing insights into the prediction pipeline and user interactions.

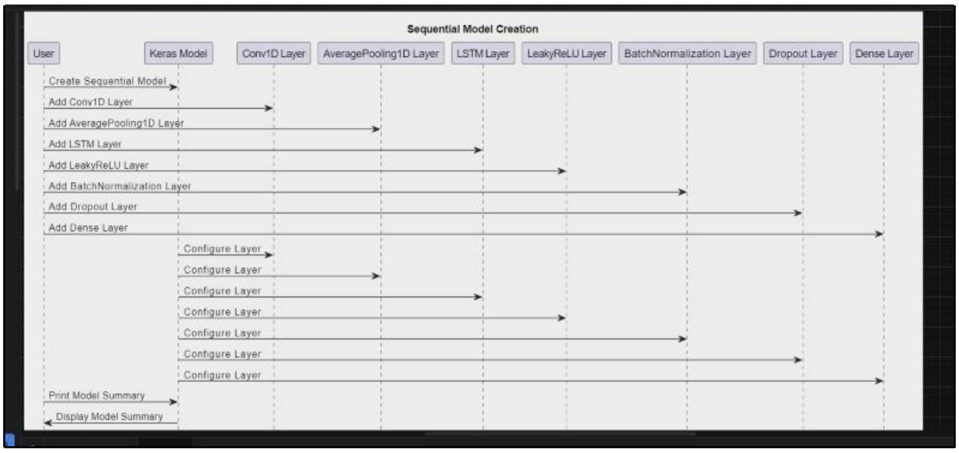


Figure 4.2.3. Sequence Diagram

4.2.4 Activity Diagram

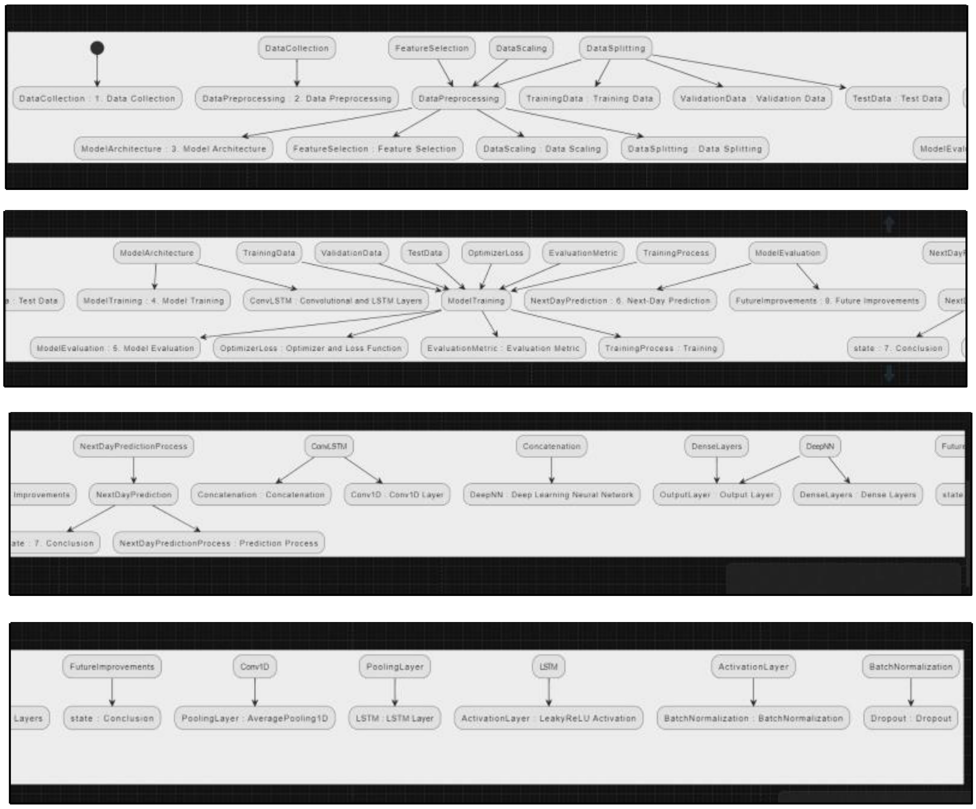

DataCollection
FeatureSelection
DataScaling
DataSplitting
DataCollection : 1. Data Collection
DataPreprocessing : 2. Data Preprocessing
DataPreprocessing
TrainingData : Training Data
ValidationData : Validation Data
TestData : Test Data
ModelArchitecture : 3. Model Architecture
FeatureSelection : Feature Selection
DataScaling : Data Scaling
DataSplitting : Data Splitting
ModelEval
ModelArchitecture
TrainingData
ValidationData
TestData
OptimizerLoss
EvaluationMetric
TrainingProcess
ModelEvaluation
NextDayP
: Test Data
ModelTraining : 4. Model Training
ConvLSTM : Convolutional and LSTM Layers
ModelTraining
NextDayPrediction : 6. Next-Day Prediction
FutureImprovements : 8. Future Improvements
Next
ModelEvaluation : 5. Model Evaluation
OptimizerLoss : Optimizer and Loss Function
EvaluationMetric : Evaluation Metric
TrainingProcess : Training
state : 7. Conclusion
NextDayPredictionProcess
ConvLSTM
Concatenation
DenseLayers
DeepNN
Futur
Improvements
NextDayPrediction
Concatenation : Concatenation
Conv1D : Conv1D Layer
DeepNN : Deep Learning Neural Network
OutputLayer : Output Layer
DenseLayers : Dense Layers
state
ate : 7. Conclusion
NextDayPredictionProcess : Prediction Process
FutureImprovements
Conv1D
PoolingLayer
LSTM
ActivationLayer
BatchNormalization
Layers
state : Conclusion
PoolingLayer : AveragePooling1D
LSTM : LSTM Layer
ActivationLayer : LeakyReLU Activation
BatchNormalization : BatchNormalization
Dropout : Dropout

Figure 4.2.4 Activity Diagram

4.3 SYSTEM DESIGN

4.3.1Module Description

Data Collection and Preprocessing Module:

This module focuses on the collection and preprocessing of historical cryptocurrency price data from various exchanges. It includes processes for data retrieval, cleansing, and transformation. The data is then stored in a structured SQL database for use in subsequent modules.

Model Development Module:

This module is responsible for creating and training the deep learning models used for cryptocurrency price prediction. It involves the development of Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and 1D Convolutional Neural Network (CNN) models.

Ensemble Modeling Module:

This module focuses on combining the outputs of the individual LSTM, GRU, and CNN models using ensemble techniques. The goal is to improve prediction accuracy and robustness by aggregating multiple forecasts.

Feature Engineering Module:

This module is responsible for engineering technical indicators from the cryptocurrency price data. These indicators serve as additional features for the models, enriching their understanding of market dynamics.

Trading Strategy Formulation Module:

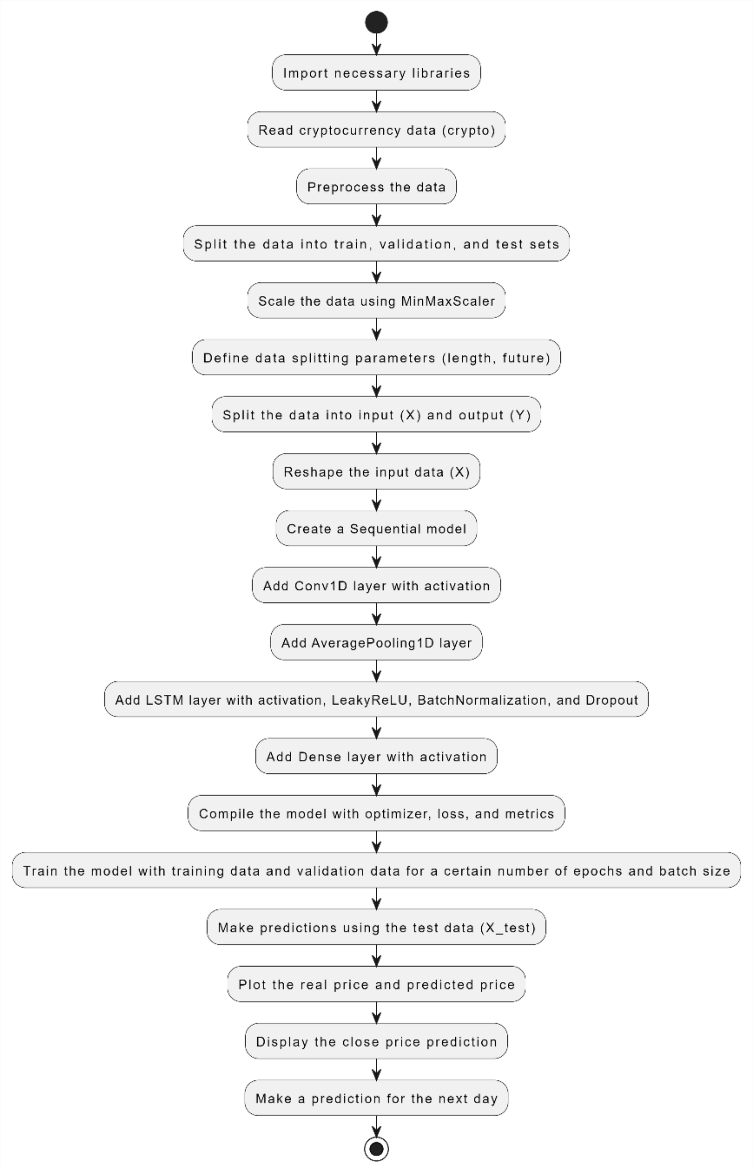
In this module, trading strategies are developed based on the price forecasts and enriched dataset with technical indicators. The strategies aim to provide actionable insights to cryptocurrency traders and investors.

Containerization and Deployment Module:

This module is responsible for packaging the entire system into containers for deployment on cloud platforms. It ensures scalability and accessibility while also incorporating model monitoring and periodic retraining pipelines.

Accuracy and Robustness Monitoring Module:

This module continuously monitors the accuracy and robustness of the cryptocurrency price prediction system. It assesses the system's performance under different market conditions and addresses potential issues in real-time.


Import necessary libraries
Read cryptocurrency data (crypto)
Preprocess the data
Split the data into train, validation, and test sets
Scale the data using MinMaxScaler
Define data splitting parameters (length, future)
Split the data into input (X) and output (Y)
Reshape the input data (X)
Create a Sequential model
Add Conv1D layer with activation
Add AveragePooling1D layer
Add LSTM layer with activation, LeakyReLU, BatchNormalization, and Dropout
Add Dense layer with activation
Compile the model with optimizer, loss, and metrics
Train the model with training data and validation data for a certain number of epochs and batch size
Make predictions using the test data (X_test)
Plot the real price and predicted price
Display the close price prediction
Make a prediction for the next day

4.3.1 Flow Chart

4.3.2 Database Design

Designing a database schema for a cryptocurrency dataset involves defining the tables, their attributes, relationships, and data types. Below is a basic example of how you might design a database schema for cryptocurrency data. Keep in mind that the actual schema can be more complex depending on the specific requirements of your application.

The database consists of three tables: cryptocurrencies, cryptocurrency_prices, and cryptocurrency_news. The cryptocurrencies table stores the basic information about each cryptocurrency, such as its name, symbol, and market cap. The cryptocurrency_prices table stores the historical prices of each cryptocurrency. The cryptocurrency_news table stores news articles about cryptocurrencies.

Tables:

cryptocurrencies: This table stores the basic information about each cryptocurrency, such as its name, symbol, and market cap.
cryptocurrency_prices: This table stores the historical prices of each cryptocurrency.
cryptocurrency_news: This table stores news articles about cryptocurrencies.

Columns:

cryptocurrencies

- id (primary key)
- name
- symbol
- market_cap

cryptocurrency_prices

- id (primary key)
- cryptocurrency_id (foreign key to the cryptocurrencies table)
- timestamp
- open
- high
- low
- close
- volume

cryptocurrency_news

- id (primary key)
- cryptocurrency_id (foreign key to the cryptocurrencies table)
- headline
- summary
- url

Relationships:

- The cryptocurrency_prices table has a one-to-many relationship with the cryptocurrencies table. This means that each cryptocurrency can have many price entries, but each price entry can only belong to one cryptocurrency.
- The cryptocurrency_news table has a one-to-many relationship with the cryptocurrencies table. This means that each cryptocurrency can have many news articles, but each news article can only belong to one cryptocurrency.

Data Types:

- id: integer
- name: string
- symbol: string
- market_cap: decimal
- timestamp: datetime
- open: decimal
- high: decimal
- low: decimal
- close: decimal
- volume: decimal
- headline: string
- summary: text
- url: string

This schema is a basic starting point. Depending on your specific use case, you may need to expand and customize the schema to include additional information or tables. For example, you might want to add tables for order history, wallet addresses, or additional details about each trade. The choice of database system (e.g., MySQL, PostgreSQL, MongoDB) and its specific features should also influence your schema design.

5. IMPLEMENTATION

5.1. IMPLEMENTATION

The implementation of cryptocurrency price prediction using a CNN-LSTM model is a complex yet powerful process that leverages deep learning to make forecasts in the dynamic world of cryptocurrencies. The journey begins with data collection and preprocessing, where historical cryptocurrency price and market data are gathered and transformed into a suitable format for analysis. Next, a hybrid CNN-LSTM model is constructed, combining convolutional neural networks (CNNs) for feature extraction and long short-term memory (LSTM) networks for capturing sequential dependencies. The model is trained using historical data, adjusting its parameters to learn patterns, trends, and anomalies in the cryptocurrency market. Validation is crucial to ensure the model's accuracy and generalization to unseen data. Finally, users can interact with the system to obtain real-time price predictions. This implementation empowers users with the capability to navigate the highly volatile cryptocurrency market with data-driven insights and informed decision-making, opening doors to new opportunities in the world of digital currencies.

Requirements:

- Data Collection: Gather historical cryptocurrency price and market data from reliable sources. This data should cover a variety of cryptocurrencies, including their price, trading volume, and other relevant metrics.

- Data Preprocessing: Clean and preprocess the collected data, handling missing values, outliers, and ensuring it's in a format suitable for analysis.

- Model Selection: Choose an appropriate deep learning model for cryptocurrency price prediction. In this case, you've mentioned using a CNN-LSTM hybrid model, so the requirement is to implement and configure this model.

- Training and Validation: Split the data into training, validation, and testing datasets. Train the model on historical data, validate its performance, and fine-tune its hyperparameters.

- User Interface: Develop a user-friendly interface for users to interact with the system. This interface should allow users to input data, request predictions, and visualize results.

- Data Security: Implement security measures to protect the cryptocurrency data and user information from unauthorized access or breaches.

- Scalability: Design the system to handle a growing volume of data and users over time.

Goals:

- Accurate Predictions: The primary goal is to build a predictive model that accurately forecasts cryptocurrency prices, helping users make informed investment decisions.

- User-Friendly Interface: Create an intuitive and accessible user interface that allows users, even those with limited technical knowledge, to interact with the system effortlessly.

- Real-Time Updates: Provide timely predictions and real-time updates to keep users informed about changing market conditions.

- Scalability: Ensure that the system can accommodate an increasing number of users and data without compromising performance.

- Data Integrity: Maintain data integrity by implementing robust data preprocessing and security measures to protect sensitive information.

5.2. SOURCE CODE

```
import numpy as npimport
pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
df=crypto[["open","high","low","close","Volume BTC","Volume USD"]]df.head()
times = sorted(df.index.values)  #  get  the  times last_10 =
sorted(df.index.values)[-int(0.1*len(times))]last_20 =
sorted(df.index.values)[-int(0.2*len(times))]test_df =
df[(df.index >= last_10)]
validation_df = df[(df.index >= last_20) & (df.index < last_10)]train_df =
df[(df.index < last_20)]

train = train_df.values
valid = validation_df.valuestest=
test_df.values

print("train shape {0}".format(train.shape)) print("valid  shape
{0}".format(valid.shape)


print("test shape {0}".format(test.shape))



from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(train)



train= scaler.transform(train)
```

```
valid= scaler.transform(valid) test= scaler.transform(test)

#splitting the data length=100 future=5
def split_data(data):
X=[]
Y=[]
for i in range(length,len(data)-future+1): X.append(data[i-
length:i]) Y.append(data[i+(future-1),3])
return np.array(X), np.array(Y)


X_train, y_train = split_data(train) X_test, y_test =
split_data(test) X_valid, y_valid = split_data(valid)
X_train = np.reshape(X_train, (X_train.shape[0],
X_train.shape[1], 6)) X_valid = np.reshape(X_valid,
(X_valid.shape[0], X_valid.shape[1], 6)) X_test =
np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 6))

import keras
import tensorflow as tf
from tensorflow.keras import layers from keras.models import
Sequential
from tensorflow.keras.optimizers import Adam


from keras.layers import Dense, LSTM, LeakyReLU,
Dropout, BatchNormalization


'''# Add a CNN layer
model.add(Conv1D(filters=16,  kernel_size=3,
        activation='relu', input_shape=(X_train.shape[1],1)))

# Add an LSTM layer model.add(LSTM(units=32,
activation='relu'))

# Add a dropout layer model.add(Dropout(0.2))
# Add a dense layer
model.add(Dense(1, activation='sigmoid'))



# Compile the model
model.compile(loss='mse', optimizer='adam',
metrics=['accuracy'])''' model = Sequential()
model.add(tf.keras.layers.Conv1D(120, 3,
activation="relu",input_shape=(100, 6)))
model.add(tf.keras.layers.AveragePooling1D(4))
model.add(LSTM(units=120, input_shape=(100, 6)))
model.add(LeakyReLU(alpha=0.5))
model.add(BatchNormalization()) model.add(Dropout(0.5))
#model_LSTM2.add(Dense(64, activation='relu'))
#model_LSTM2.add(BatchNormalization())
#model_LSTM2.add(Dropout(0.1))
model.add(Dense(1,activation='linear')) model.summary()
model.compile(optimizer='adam', loss='mean_squared_error',
metrics = ('MAPE')) model.summary()
history          =          model.fit(X_train,
        y_train,validation_data=(X_valid, y_valid),
        epochs=10, batch_size=128)
#model evaluation

y_pred = model.predict(X_test) y_pred

plt.plot(y_test, color = 'black', label = 'Real Price')
plt.plot(y_pred, color = 'purple', label = 'Predicted Price')
plt.title('Close Price Prediction', fontsize=30)
#plt.xticks(range(0,df.shape[0],50),df['Date'].loc[::50],rotation
=45) plt.xlabel('DateTime')
plt.ylabel('Close Price') plt.legend(fontsize=18) plt.show()

# Make a prediction for the next day next_day_pred =
model.predict(X_test[-1:])

print('Predicted price for the next day:', next_day_pred[0][0]
```

6. SYSTEM TESTING

6.1 TYPES OF TESTCASES

- Functional Testing: It validates that the core features, including user registration, travel plan creation, guide matching, budget management, and community collaboration, work as intended.

- Usability Testing: Assess the overall user experience, including the intuitiveness of the user interface, ease of n avigation, and user guidance.

- Performance Testing: Evaluate the system's response times, scalability, and resource usage to ensure it can handle concurrent users and maintain responsiveness.

- Security Testing: Verify that user data is securely stored, transmitted, and protected against potential security threats and vulnerabilities.

- Compatibility Testing: Test the app or website on different devices, browsers, and operating systems to ensure compatibility and responsiveness across platforms.

- Load Testing: Assess how the system performs under heavy loads, simulating a large number of users to ensure it remains stable and responsive.

- Data Integrity Testing: Verify that user data remains accurate and consistent when interacting with various system components.

- Regression Testing: Ensure that new updates or features do not introduce new issues or break existing functionalities.

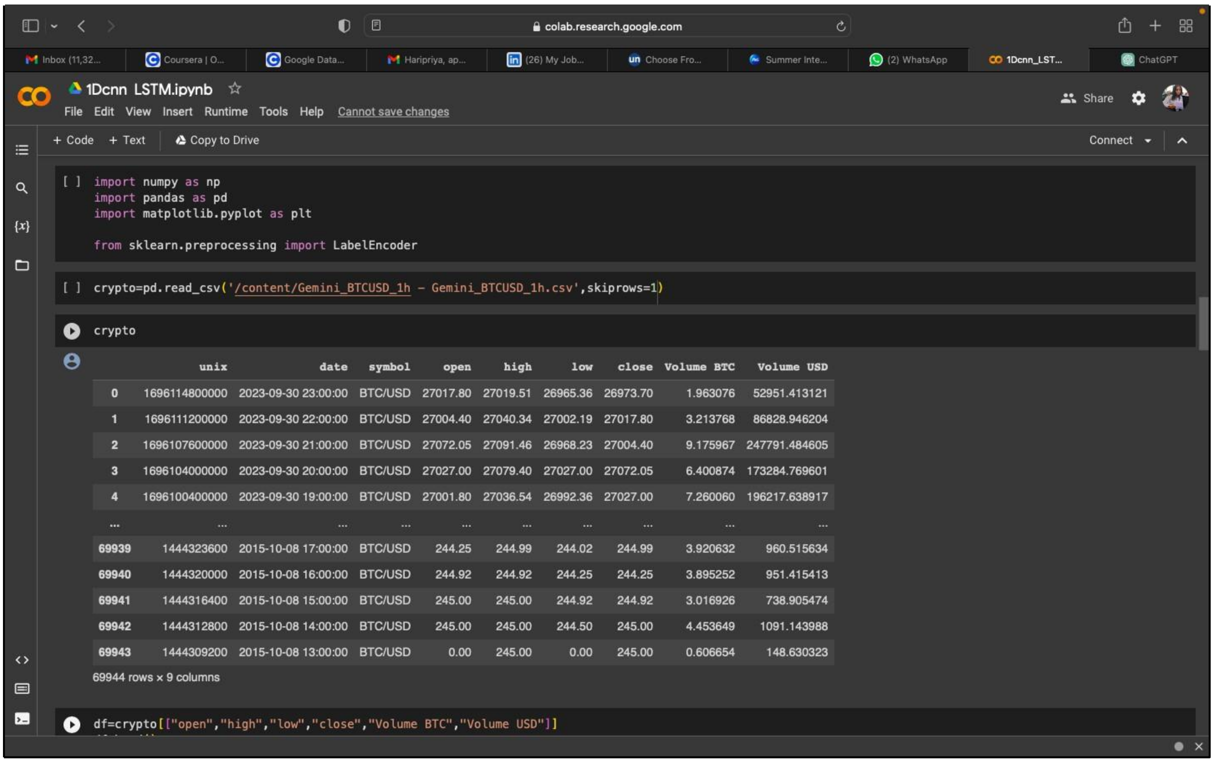
- Cross-Browser and Cross-Device Testing: Test the project on different web browsers and various devices to ensure a consistent experience.

- Exploratory Testing: Conduct exploratory testing to discover unexpected issues, defects, or usability problems.
- Documentation Testing: Review and validate the accuracy and completeness of user documentation and help resources

6.2 TEST CASES

S.NO	**Testcase notation**	Testcase notation	Intput	Requirement	Testcase status
1	Valid Input	T1	Historical price data	Test the model with valid input to ensure it can make accurate predictions.	Pass
2	Invalid Input	T2	Incomplete historical price data for the cryptocurrency.	Test the model with incomplete or missing data to ensure it handles errors gracefully.	Pass
3	Out-of-sample Testing	T3	Historical price data for a period not included in the training data	Test the model with data the falls outside the training dataset to evaluate its generalization capability.	Pass
4	Performance Testing	T4	Historical cryptocurrency prices.	Assess the model's performance in terms of prediction speed and resource.	Pass
5	Sensitivity Analysis	T5	Number of hidden layers in nn.	Analyze the model's sensitivity to parameter changes.	Pass

7. OUTPUT SCREENS



```
[ ] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt

    from sklearn.preprocessing import LabelEncoder
```

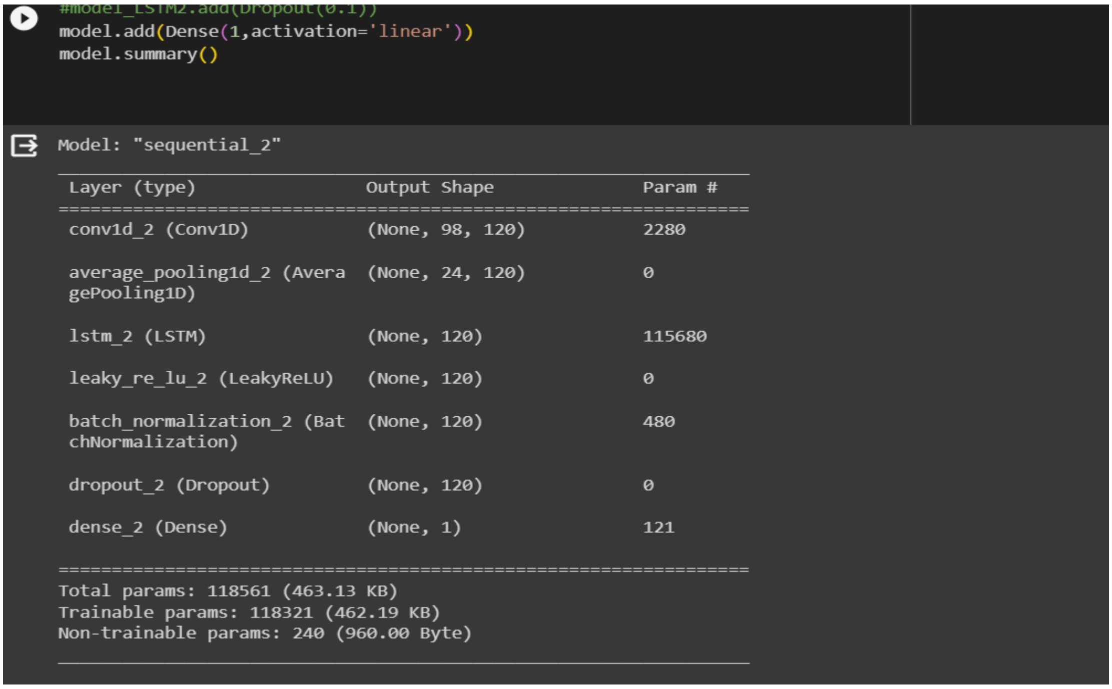
```
[ ] crypto=pd.read_csv('/content/Gemini_BTCUSD_1h - Gemini_BTCUSD_1h.csv',skiprows=1)
```

```
crypto
```

```
                unix                 date   symbol      open      high       low     close  Volume BTC     Volume USD
    0   1696114800000  2023-09-30 23:00:00  BTC/USD  27017.80  27019.51  26965.36  26973.70    1.963076   52951.413121
    1   1696111200000  2023-09-30 22:00:00  BTC/USD  27004.40  27040.34  27002.19  27017.80    3.213768   86828.946204
    2   1696107600000  2023-09-30 21:00:00  BTC/USD  27072.05  27091.46  26968.23  27004.40    9.175967  247791.484605
    3   1696104000000  2023-09-30 20:00:00  BTC/USD  27027.00  27079.40  27027.00  27072.05    6.400874  173284.769601
    4   1696100400000  2023-09-30 19:00:00  BTC/USD  27001.80  27036.54  26992.36  27027.00    7.260060  196217.638917
  ...             ...                  ...      ...       ...       ...       ...       ...         ...            ...
69939      1444323600  2015-10-08 17:00:00  BTC/USD    244.25    244.99    244.02    244.99    3.920632     960.515634
69940      1444320000  2015-10-08 16:00:00  BTC/USD    244.92    244.92    244.25    244.25    3.895252     951.415413
69941      1444316400  2015-10-08 15:00:00  BTC/USD    245.00    245.00    244.92    244.92    3.016926     738.905474
69942      1444312800  2015-10-08 14:00:00  BTC/USD    245.00    245.00    244.50    245.00    4.453649    1091.143988
69943      1444309200  2015-10-08 13:00:00  BTC/USD      0.00    245.00      0.00    245.00    0.606654     148.630323
69944 rows × 9 columns
```

```
df=crypto[["open","high","low","close","Volume BTC","Volume USD"]]
```

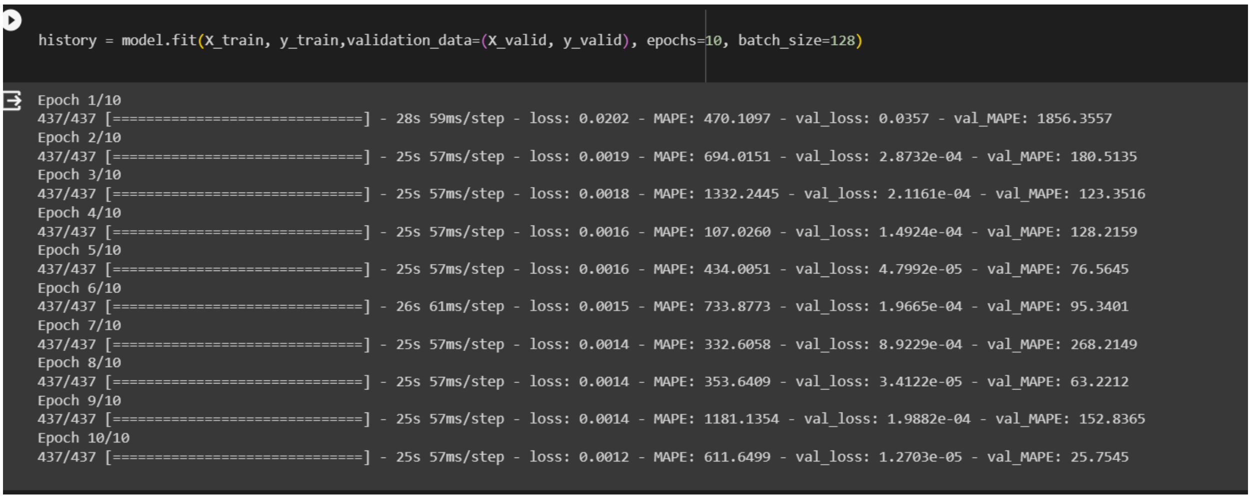
Fig.7.1 Dataset



```
#model_LSTM2.add(Dropout(0.1))
model.add(Dense(1,activation='linear'))
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d_2 (Conv1D)           (None, 98, 120)           2280

 average_pooling1d_2 (Avera  (None, 24, 120)           0
 gePooling1D)

 lstm_2 (LSTM)               (None, 120)               115680

 leaky_re_lu_2 (LeakyReLU)   (None, 120)               0

 batch_normalization_2 (Bat  (None, 120)               480
 chNormalization)

 dropout_2 (Dropout)         (None, 120)               0

 dense_2 (Dense)             (None, 1)                 121

=================================================================
Total params: 118561 (463.13 KB)
Trainable params: 118321 (462.19 KB)
Non-trainable params: 240 (960.00 Byte)
_________________________________________________________________
```

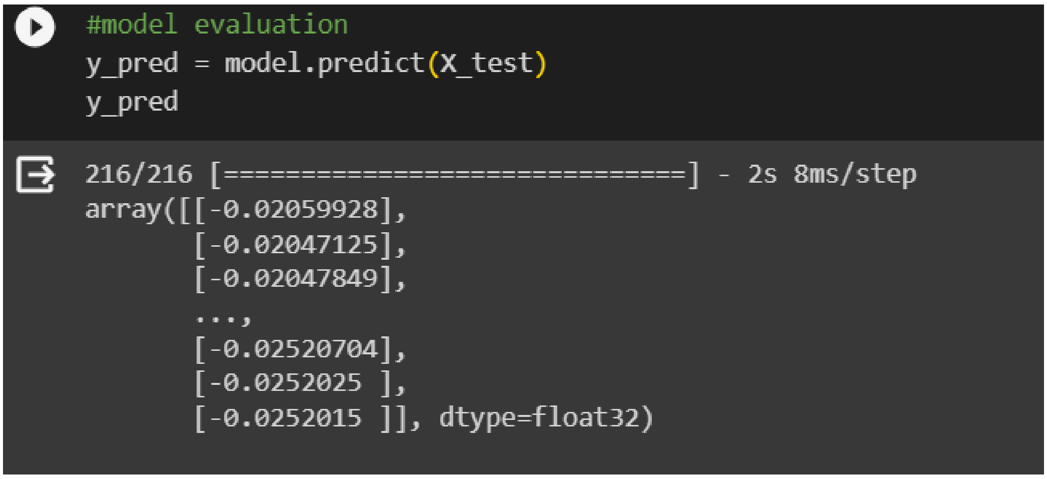
Fig.7.2 .Model summary.



```
history = model.fit(X_train, y_train,validation_data=(X_valid, y_valid), epochs=10, batch_size=128)
```

```
Epoch 1/10
437/437 [==============================] - 28s 59ms/step - loss: 0.0202 - MAPE: 470.1097 - val_loss: 0.0357 - val_MAPE: 1856.3557
Epoch 2/10
437/437 [==============================] - 25s 57ms/step - loss: 0.0019 - MAPE: 694.0151 - val_loss: 2.8732e-04 - val_MAPE: 180.5135
Epoch 3/10
437/437 [==============================] - 25s 57ms/step - loss: 0.0018 - MAPE: 1332.2445 - val_loss: 2.1161e-04 - val_MAPE: 123.3516
Epoch 4/10
437/437 [==============================] - 25s 57ms/step - loss: 0.0016 - MAPE: 107.0260 - val_loss: 1.4924e-04 - val_MAPE: 128.2159
Epoch 5/10
437/437 [==============================] - 25s 57ms/step - loss: 0.0016 - MAPE: 434.0051 - val_loss: 4.7992e-05 - val_MAPE: 76.5645
Epoch 6/10
437/437 [==============================] - 26s 61ms/step - loss: 0.0015 - MAPE: 733.8773 - val_loss: 1.9665e-04 - val_MAPE: 95.3401
Epoch 7/10
437/437 [==============================] - 25s 57ms/step - loss: 0.0014 - MAPE: 332.6058 - val_loss: 8.9229e-04 - val_MAPE: 268.2149
Epoch 8/10
437/437 [==============================] - 25s 57ms/step - loss: 0.0014 - MAPE: 353.6409 - val_loss: 3.4122e-05 - val_MAPE: 63.2212
Epoch 9/10
437/437 [==============================] - 25s 57ms/step - loss: 0.0014 - MAPE: 1181.1354 - val_loss: 1.9882e-04 - val_MAPE: 152.8365
Epoch 10/10
437/437 [==============================] - 25s 57ms/step - loss: 0.0012 - MAPE: 611.6499 - val_loss: 1.2703e-05 - val_MAPE: 25.7545
```

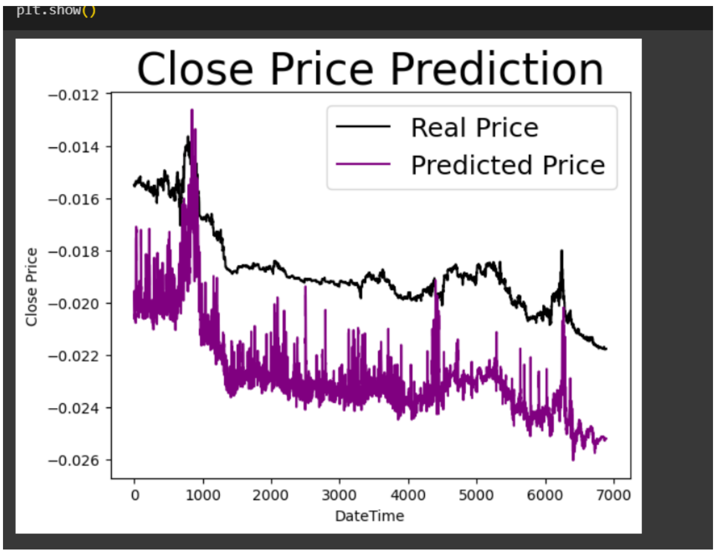
Fig.7.3. Model fitting.



```
#model evaluation
y_pred = model.predict(X_test)
y_pred
```

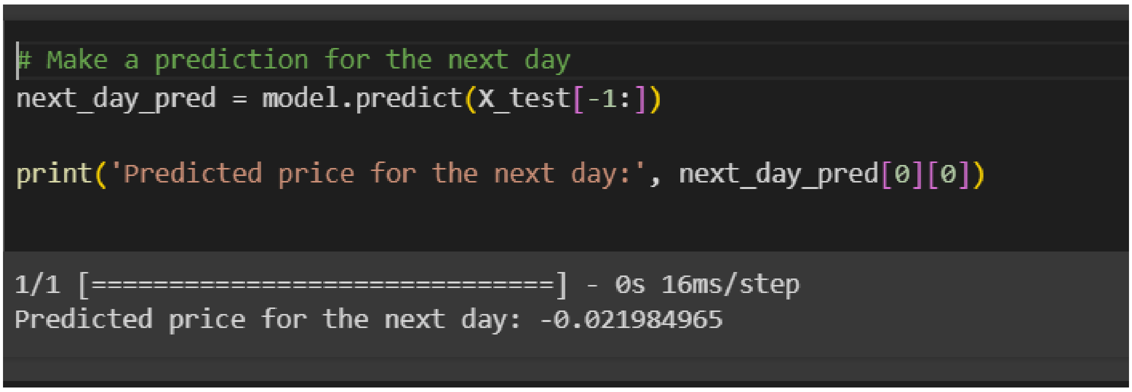
```
216/216 [==============================] - 2s 8ms/step
array([[-0.02059928],
       [-0.02047125],
       [-0.02047849],
       ...,
       [-0.02520704],
       [-0.0252025 ],
       [-0.0252015 ]], dtype=float32)
```

Fig.7.4. Predicted values.



```
plt.show()
```

Fig.7.5. Price prediction.



```python
# Make a prediction for the next day
next_day_pred = model.predict(X_test[-1:])

print('Predicted price for the next day:', next_day_pred[0][0])
```

```
1/1 [==============================] - 0s 16ms/step
Predicted price for the next day: -0.021984965
```

Fig.7.6. Prediction for next day.

8. CONCLUSION

8.1. CONCLUSION

In conclusion, a CRYPTOCURRENCY PRICE PREDICTION project using machine learningis a critical endeavor in the age of digital information. By harnessing advanced algorithms anddata analysis techniques, it holds the potential to mitigate the harmful spread of misinformation. However, its success hinges on continuous improvement, robust evaluation, and ethical considerations to ensure accurate detection, fairness, and privacy protection. As the landscapeof fake news evolves, this project remains an essential tool in upholding the integrity of information dissemination in the digital era.

8.2.FURTHER ENCHANCEMENTS

To further enhance a CRYPTOCURRENCY PRICE PREDICTION project using machine learning, considerations include the integration of advanced techniques like deep learning andreinforcement learning for improved accuracy. Emphasizing explain ability and interpretabilityin models can enhance user trust and understanding. Collaborative filtering methods can help identify coordinated efforts to spread misinformation across social networks. Real-time monitoring and alerting systems can provide timely responses to emerging fake news. Overall,a holistic approach combining technology, user empowerment, and global cooperation is essential for the ongoing evolution of CRYPTOCURRENCY PRICE PREDICTION projects.

9. BIBLOGRAPY

1. https://www.sciencedirect.com/science/article/pii/S2405918821000027
2. https://www.mdpi.com/1911-8074/16/1/51#:~:text=In%20addition%20to%20predicting%20the,with%20an%20accuracy%20of%2084.2%25
3. https://iopscience.iop.org/article/10.1088/1757-899X/928/3/032007/pdf
4. https://www.mdpi.com/2306-5729/7/11/149
5. https://news.mit.edu/2014/mit-computer-scientists-can-predict-price-bitcoin

10. APPENDICES

Project Overview:

Title: cryptocurrency price prediction using CNN-LSTM

Description:

The "Cryptocurrency Price Prediction Using CNN-LSTM" project is an innovative and data-driven initiative that leverages deep learning techniques to forecast the prices of cryptocurrencies in the highly volatile digital currency market. The project's primary goal is to empower users with accurate and real-time insights, enabling them to make informed investment decisions in this dynamic financial landscape.

Features:

1. Hybrid Model: The core of the project involves the implementation of a hybrid model that combines Convolutional Neural Networks (CNNs) for feature extraction and Long Short-Term Memory (LSTM) networks for capturing sequential dependencies. This model is designed to learn complex patterns and trends in cryptocurrency prices.
2. User Interaction: The project incorporates a user-friendly interface that allows users to input data, request real-time predictions, and visualize the results. This interface is designed to be accessible to users with varying levels of technical expertise.
3. Predictive Analytics: The project provides predictions for cryptocurrency prices, enabling users to stay updated on market trends and make well-informed decisions. These predictions are generated based on the model's analysis of historical and current data.
4. Security and Scalability: Data security measures are implemented to safeguard sensitive information. The system is also designed to be scalable, accommodating an increasing number of users and data without compromising performance.

Benefits:

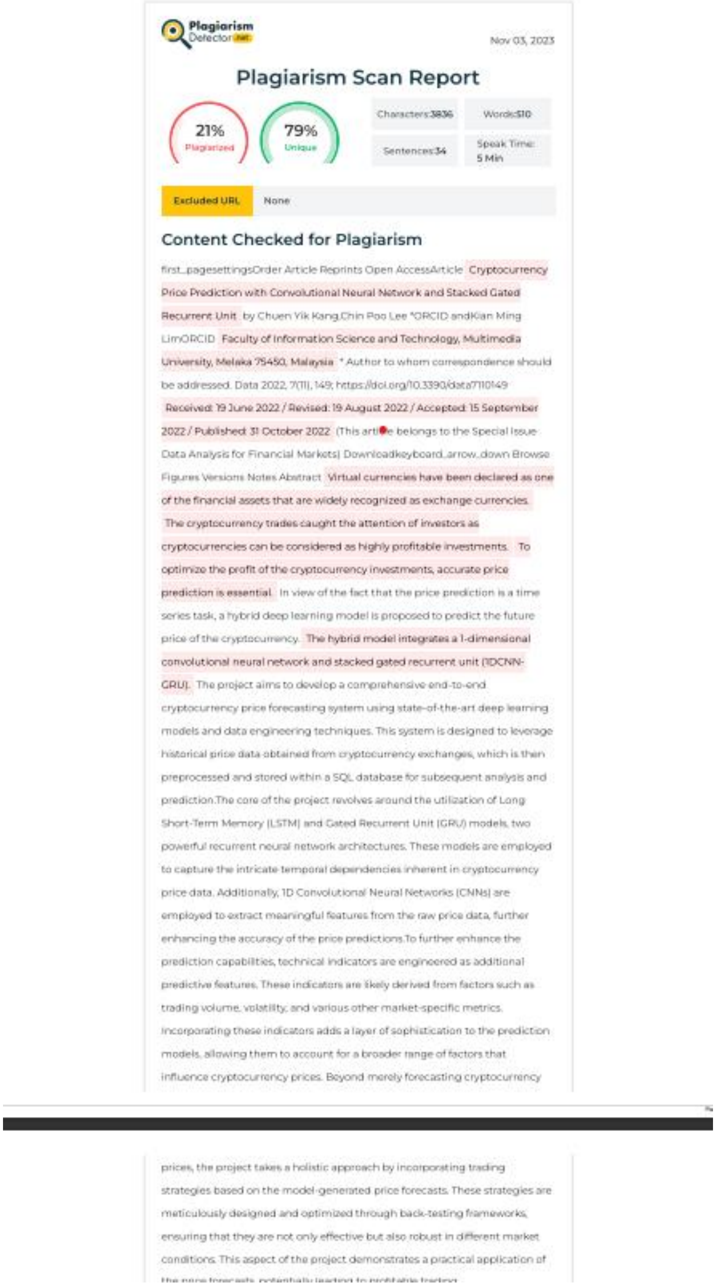
Data-Driven Insights: Users gain access to data-driven insights into cryptocurrency markets, helping them understand price trends, patterns, and potential future movements.

Informed Decision-Making: With real-time predictions at their fingertips, users can make informed decisions regarding cryptocurrency investments, trading strategies, and risk management.

User Accessibility: The user interface ensures accessibility, making the project suitable for a wide range of users, from crypto enthusiasts to investors and traders.

The "Cryptocurrency Price Prediction Using CNN-LSTM" project brings together data science, deep learning, and user-friendly interfaces to bridge the gap between cryptocurrency market dynamics and user decision-making. It offers a valuable tool for users seeking to navigate the complexities of the cryptocurrency world and make data-driven choices for their financial ventures.

11. PLAGIARISM REPORT



Nov 03, 2023

Plagiarism Scan Report

21% Plagiarized

79% Unique

Characters:3836

Words:510

Sentences:34

Speak Time: 5 Min

Excluded URL None

Content Checked for Plagiarism

first_pagesettingsOrder Article Reprints Open AccessArticle Cryptocurrency Price Prediction with Convolutional Neural Network and Stacked Gated Recurrent Unit by Chuen Yik Kang,Chin Poo Lee *ORCID andKian Ming LimORCID Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia * Author to whom correspondence should be addressed. Data 2022, 7(11), 149; https://doi.org/10.3390/data7110149 Received: 19 June 2022 / Revised: 19 August 2022 / Accepted: 15 September 2022 / Published: 31 October 2022 (This article belongs to the Special Issue Data Analysis for Financial Markets) Downloadkeyboard_arrow_down Browse Figures Versions Notes Abstract Virtual currencies have been declared as one of the financial assets that are widely recognized as exchange currencies. The cryptocurrency trades caught the attention of investors as cryptocurrencies can be considered as highly profitable investments. To optimize the profit of the cryptocurrency investments, accurate price prediction is essential. In view of the fact that the price prediction is a time series task, a hybrid deep learning model is proposed to predict the future price of the cryptocurrency. The hybrid model integrates a 1-dimensional convolutional neural network and stacked gated recurrent unit (1DCNN-GRU). The project aims to develop a comprehensive end-to-end cryptocurrency price forecasting system using state-of-the-art deep learning models and data engineering techniques. This system is designed to leverage historical price data obtained from cryptocurrency exchanges, which is then preprocessed and stored within a SQL database for subsequent analysis and prediction.The core of the project revolves around the utilization of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models, two powerful recurrent neural network architectures. These models are employed to capture the intricate temporal dependencies inherent in cryptocurrency price data. Additionally, 1D Convolutional Neural Networks (CNNs) are employed to extract meaningful features from the raw price data, further enhancing the accuracy of the price predictions.To further enhance the prediction capabilities, technical indicators are engineered as additional predictive features. These indicators are likely derived from factors such as trading volume, volatility, and various other market-specific metrics. Incorporating these indicators adds a layer of sophistication to the prediction models, allowing them to account for a broader range of factors that influence cryptocurrency prices. Beyond merely forecasting cryptocurrency prices, the project takes a holistic approach by incorporating trading strategies based on the model-generated price forecasts. These strategies are meticulously designed and optimized through back-testing frameworks, ensuring that they are not only effective but also robust in different market conditions. This aspect of the project demonstrates a practical application of the price forecasts, potentially leading to profitable trading

Figure 11