Unified Model Development for Predicting the Close-Price of Various Cryptocurrencies

Yuan Chen

A Thesis Submitted in Partial Fulfillment of Requirements for
degree of Doctor of Philosophy in Computer Science
March 2025

Unified Model Development for Predicting the Close-Price of Various Cryptocurrencies

วิทยานิพนธ์

ของ

Yuan Chen

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
มีนาคม 2568

Unified Model Development for Predicting the Close-Price of Various Cryptocurrencies

Yuan Chen

A Thesis Submitted in Partial Fulfillment of Requirements
for Doctor of Philosophy (Computer Science)
March 2025

The examining committee has unanimously approved this Thesis, submitted by Ms. Yuan Chen , as a partial fulfillment of the requirements for the Doctor of Philosophy Computer Science at Mahasarakham University

Examining Committee

....Chairman
(Assoc. Prof. Suphakant Phimoltares , Ph.D.)

....Advisor
(Assoc. Prof. Jantima Polpinij , Ph.D.)

....Co-advisor
(Assoc. Prof. Gamgarn Sompasertsri , Ph.D.)

....Committee
(Assoc. Prof. Panida Songram , Ph.D.)

....External Committee
(Asst. Prof. Manasawee Kaenampornpan , Ph.D.)

Mahasarakham University has granted approval to accept this Thesis as a partial fulfillment of the requirements for the Doctor of Philosophy Computer Science

....	....
(Assoc. Prof. Jantima Polpinij , Ph.D.)	(Prof. Anongrit Kangrang , Ph.D.)
Dean of The Faculty of Informatics	Acting Dean of Graduate School

TITLE Unified Model Development for Predicting the Close-Price of Various Cryptocurrencies
AUTHOR Yuan Chen
ADVISORS Associate Professor Jantima Polpinij , Ph.D.
Associate Professor Gamgarn Sompasertsri , Ph.D.
DEGREE Doctor of Philosophy **MAJOR** Computer Science
UNIVERSITY Mahasarakham University **YEAR** 2025

ABSTRACT

This study introduces a unified prediction model based on data fusion techniques for forecasting the prices of multiple cryptocurrencies. The cryptocurrency market is highly volatile and complex, characterized by dynamic interactions among various digital assets. Traditional models that focus on a single cryptocurrency often overlook these interconnections, limiting their predictive accuracy and practical applicability. To overcome this limitation, the proposed model integrates features from three major cryptocurrencies—Bitcoin (BTC), Ethereum (ETH), and Ripple (XRP)—and applies three data fusion strategies: Concatenation, Averaging, and a hybrid method that combines both Concatenation and Averaging. These feature fusion strategies are combined with four prediction algorithms: Support Vector Regression (SVR), Random Forest (RF), Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM). To evaluate the model's performance and generalization capability, extensive prediction experiments were conducted on five cryptocurrencies: BTC, ETH, XRP, Monero (XMR), and Litecoin (LTC). The forecasting tasks were performed over four different time horizons: 5-day, 7-day, 24-day, and 30-day periods. The results demonstrate that for short-term forecasting (5-day and 7-day), the unified model using LSTM with average-based data fusion yields the highest prediction accuracy. For long-term forecasting (24-day and 30-day), the model employing GRU with average-based data fusion performs more effectively. In comparison with traditional single-cryptocurrency prediction models, the proposed unified model achieves notable performance improvements, with average reductions of 2.39% in Mean Absolute Error (MAE), 2.12% in Root Mean Square Error (RMSE), and 2.13% in Mean Absolute Percentage Error (MAPE). These improvements indicate that incorporating data fusion from multiple cryptocurrencies significantly enhances predictive performance. Overall, the findings validate the effectiveness and robustness of the unified data fusion approach in cryptocurrency price prediction. By capturing cross-asset relationships and leveraging advanced machine learning techniques, this study provides a more comprehensive and accurate prediction framework. This not only benefits investors seeking more reliable tools for decision-making but also contributes to the broader research on multi-source time series forecasting in highly volatile markets.

Keyword : Cryptocurrency, closing price prediction, unified predictive model, data fusion, machine learning, deep learning

ACKNOWLEDGEMENTS

The successful completion of this dissertation would not have been possible without the support, guidance, and encouragement of many individuals and institutions, to whom I extend my deepest gratitude.

First and foremost, I sincerely thank my supervisors, Associate Professor Dr. Jantima Polpinij and Associate Professor Dr. Gamgarn Somprasertsri. Your expertise, unwavering support, and insightful guidance have been invaluable throughout my academic journey. Your dedication to research, meticulous approach, and passion for knowledge have profoundly shaped my growth as a researcher. I am also immensely grateful to Dr. Lien Thai Son for his invaluable advice and guidance, which have significantly influenced my research perspective and academic development.

Throughout my Ph.D. studies, I have been fortunate to receive the support of colleagues, mentors, and friends both within and beyond the Faculty of Informatics. Their encouragement and shared experiences have enriched my journey, fostering both academic and personal growth. I am deeply appreciative of the instructors who have imparted knowledge and introduced me to innovative learning techniques.

My deepest gratitude goes to my family, whose unwavering love and belief in me have provided the foundation for my academic pursuits. I would also like to express my sincere appreciation to the Faculty of Informatics at Mahasarakham University for providing essential resources and facilities that have been instrumental in conducting this research.

Finally, I am profoundly grateful to everyone I have encountered on this journey. I sincerely appreciate each and every one of you.

Yuan Chen

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1 INTRODUCTION

1.1 Background

Cryptocurrency is a type of digital or virtual currency that employs cryptography to secure and verify transactions as well as to control the creation of new units [1, 2]. Cryptocurrencies are decentralized, meaning they operate without the intervention of a central authority such as a bank or government [1]. Bitcoin, the first and most well-known cryptocurrency, was introduced in 2009 [3]. Ethereum, Litecoin, and Ripple are some other popular cryptocurrencies. Each cryptocurrency has its own set of features and applications, but they all share the same underlying blockchain technology and use cryptography to secure transactions and control the creation of new units [4]. Cryptocurrencies can be used to purchase goods and services, transfer money, and invest. They have some advantages over traditional currencies, including faster and less expensive transactions, global accessibility, and increased privacy [5].

Today, a cryptocurrency price prediction method or system is required because it can provide valuable information to cryptocurrency investors, traders, businesses, and other stakeholders [6, 7]. It is possible to make accurate predictions using advanced analytics and machine learning algorithms, which can aid in making informed decisions and optimizing strategies [8]. Furthermore, predicting the price of cryptocurrency can provide valuable insights into market trends and dynamics, such as the demand and supply factors that influence prices. These observations can be applied to market analysis and forecasting.

At present, different cryptocurrency types cannot be predicted using a unified (or single) cryptocurrency price predictor model. This is because predicting the prices of different cryptocurrencies using a single model can be difficult due to several reasons, i.e. market dynamics, unique characteristics, data variability, and market interdependencies [9].

Firstly, because of market dynamics, cryptocurrency markets are highly volatile and subject to rapid fluctuations [10]. Market sentiment, investor behavior, regulatory changes, technological advancements, and macroeconomic events all have an impact on prices. These dynamics can differ significantly between cryptocurrencies, making it difficult to capture all of the nuances in a single model.

Secondly, each cryptocurrency has its own unique characteristics, such as underlying technology, use case, market capitalization, and community support [11]. These factors can influence price behavior and must be taken into account when developing prediction models. Attempting to create a single model that encompasses the diverse characteristics of various cryptocurrencies can be overly complex and may not result in accurate forecasts.

Thirdly, different cryptocurrency types may have data variability [12]. This is because cryptocurrencies exhibit varying levels of historical price data, liquidity, and trading volumes [13]. Some cryptocurrencies may have insufficient available historical data, making it difficult to train accurate prediction models. In addition, the data quality and dependability can vary across cryptocurrencies, which further complicates the process of prediction.

Lastly, cryptocurrency markets are interdependent, and the prices of different cryptocurrencies frequently affect one another [14]. This phenomenon is known as market interdependence. Factors such as BTC's dominance, market trends, or investor behavior can have ripple effects on other cryptocurrencies [15, 16]. It can be difficult to capture these interdependencies accurately in a unified model.

As previously mentioned, the diverse characteristics and behaviors of various cryptocurrencies make it challenging to predict their prices using a single, unified model. Technical analysis tools commonly rely on historical price data, with particular attention to the closing price. The closing price is crucial as it represents the consensus reached by market participants at the end of a trading day, capturing not only the overall performance of the market on that day but also providing a key reference for accurate technical analysis. This study focuses on predicting the closing price because it plays a pivotal role in reflecting market sentiment and guiding investment decisions. However, the challenge remains in developing a unified model that can accurately predict the closing prices of different cryptocurrencies, given their market volatility and distinct characteristics.

1.2 Research Objective

This research aims to present a unified method for predicting the closing price of various cryptocurrencies.

1.3 Significance of Research

A unified model for predicting the closing prices of various cryptocurrencies may provide convenience, but it is essential to recognize the difficulties and limitations implicit in predicting the prices of various cryptocurrencies using a single model. There are possible benefits to consider:

A unified model would provide users with a single instrument or portal for predicting the prices of multiple cryptocurrencies. This can facilitate the user experience, as users will no longer have to navigate multiple models or platforms for various cryptocurrencies.

A unified model can facilitate the comparative analysis of various cryptocurrencies. Users can simply assess the predicted price movements of numerous cryptocurrencies and make informed decisions based on the expected relative performance.

Developing and maintaining a single model may be more cost-effective and resource-efficient than developing and maintaining distinct models for each cryptocurrency. It reduces the need for redundant resources and simplifies the process of development and maintenance.

A unified model may be able to capture potential cross-correlations or interdependencies between various cryptocurrencies. By analyzing multiple cryptocurrencies concurrently, the model could detect patterns or tendencies that may not be evident when analyzing each cryptocurrency separately.

A unified model can provide users with portfolios containing multiple cryptocurrencies with a holistic view of their portfolio's performance. Users are able to evaluate the overall risk and prospective returns of their portfolios by consolidating predictions for various cryptocurrencies.

A unified model can be created to accommodate the emergence of new cryptocurrencies on the market. This scalability facilitates the incorporation of new cryptocurrencies into the prediction framework without requiring the development of distinct models for each new addition.

However, it is important to note that accurately predicting cryptocurrency prices is a complex task due to the market’s volatility and the unique characteristics of each cryptocurrency. Careful consideration of the limitations and challenges is necessary when developing and utilizing such a model. Therefore, there are challenges associated with developing a unified model for predicting the closing prices of various cryptocurrencies.

1.4 Research Scope

The scope of this study is outlined as follows:

1. The objective of this research paper is to examine and develop a unified model for predicting the closing price of different types of cryptocurrencies.

2. This study employed five datasets pertaining to cryptocurrencies, specifically Bitcoin (BTC), Ethereum (ETH), Ripple (XRP), Monero (XMR), and Litecoin (LTC). The datasets may be accessed and downloaded from the following URL: https://finance.yahoo.

3. The integration of BTC, ETH and XRP into a single dataset is accomplished through a data fusion technique. In this study, three specific methods are employed: concatenation (Concat), averaging (Ave), and a combination of both concatenation and averaging (Concat&Ave). The resulting fused dataset from these methods is then utilized as the training set

4. Min-max is applied as data normalization in the stage of data pre-processing.

5. The methods based on deep learning (i.e. Long Short-Term Memory: LSTM and Gate Recurrent Unit: GRU) and machine learning (i.e. Support Vector Regression: SVR and Random Forest: RF) are utilized to develop a prediction model for predicting the closing price of various cryptocurrencies. In order to determine the best effective prediction model, the performance of the algorithms are compared.

6. The efficiency of the predictive models in predicting cryptocurrency closing prices were evaluated using Mean Absolute Error (MAE), Root mean squared error (RMSE), and Mean Absolute Percentage Error (MAPE).

1.5 Contributions

The integration of multiple data sources is achieved through a feature-level data fusion technique, resulting in a unified dataset employed as the training set. After pre-processing, this training set is utilized with machine learning or deep learning algorithms to build predictive models. The contributions of this research are as follows:

- Application of three feature fusion techniques: By employing methods such as Concatenation, Average and Concatenation&Average, features from BTC, ETH and XRP, are merged into a single, unified data source. This serves as the input for developing unified predictive models.

- Implementation of the unified models: The proposed models are designed to predict the closing prices of various cryptocurrencies, including dominant currencies such as BTC, ETH, and XRP, derivative currencies like LTC, and independent currencies like XMR.

- Comprehensive model performance evaluation: The proposed unified models are evaluated against traditional baseline models, using metrics such as MAE, RMSE, and MAPE. In addition, the results are contrasted with those reported in similar studies in the literature.

1.6 Terminologies

- A unified prediction model refers to the development of a single model that can reliably predict the closing price for various cryptocurrencies.

- Data fusion is the process of integrating information from multiple data sources into a single source so that the resulting information is more consistent, accurate, and beneficial than that which could be obtained from any one data source alone. This method aids in reducing data redundancy and achieving a balance between variance and bias.

- Feature-level data fusion (also known as feature fusion) involves merging many features obtained from multiple sources of data. This can minimize data dimensionality while also improving predictive model performance.

1.7 Abbreviations

The abbreviations used in this study are described in Table 1.1.

Table 1.1 Terminology

Abbreviation	Term
ML	Machine leaning
DL	Deep leaning
ARIMA	Auto Regressive Integrated Moving Average
LR	Linear Regression
SVR	Support Vector Regression
RF	Random Forest
CNN	Convolutional Neural Networks
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
BTC	Bitcoin
ETH	Ethereum
XRP	Ripple
XMR	Monero
LTC	Litecoin
O	Open price
H	High price
L	Low price
V	Volume
C	Close price
Concat	Data fusion technique of concatenation
Ave	Data fusion technique of averaging
Concat&Ave	Data fusion technique of concatenation and average

CHAPTER 2 LITERATURE REVIEW AND RELATED THOREMS

2.1 Cryptocurrency

2.1.1 What is Cryptocurrency?

Cryptocurrency is a type of digital or virtual currency that employs cryptography to ensure the security of financial transactions, control the issuance of new units, and authenticate the transfer of assets. Unlike fiat currencies, which are issued by governments and central institutions, cryptocurrency operate on decentralized networks based on blockchain technology. Key characteristics of virtual currencies include:

Decentralization [17]: Cryptocurrencies are commonly characterized by their lack of centralized control, wherein they operate independently from governmental or financial institutions. In contrast, these systems function through decentralized networks of computer nodes, which undertake the tasks of validating and documenting transactions.

Blockchain Technology [18]: The majority of cryptocurrencies employ blockchain technology, a decentralized and transparent database that securely and immutably records all transactions. The maintenance of this ledger is carried out by the participants of the network, so assuring both transparency and accountability.

Cryptography [18]: Cryptocurrencies employ cryptographic methodologies to safeguard transactions, regulate the generation of fresh units, and uphold the integrity of the network. Public and private keys are employed to support secure transactions and verification of ownership.

Limited Supply [19]: Several cryptocurrencies possess a predetermined issuance schedule or a set supply, so serving as a mechanism to mitigate inflationary pressures and uphold the principle of scarcity.

Peer-to-Peer Transactions [19]: Cryptocurrency transactions are conducted in a peer-to-peer manner, eliminating the necessity for intermediaries such as financial institutions. This facilitates expeditious and comparatively inexpensive international transactions.

Anonymity and Transparency [20]: Although transactions are documented on a publicly accessible blockchain, the individuals' identities participating in these transactions can be pseudonymous. The utilization of this technology affords a certain degree of confidentiality, nevertheless, the inherent openness of the blockchain guarantees the ability to track and trace transactions.

Global Accessibility [20]: Cryptocurrencies possess the capability to be accessed and utilized by anybody with an internet connection, hence facilitating financial inclusion for those who may lack access to conventional banking services.

Bitcoin (BTC) emerged as the inaugural cryptocurrency in 2009, attributed to an individual or collective entity operating under the pseudonym Satoshi Nakamoto **错误!未找到引用源。**. Bitcoin has played a pivotal role in the emergence and subsequent proliferation of several alternative cryptocurrencies, commonly known as "altcoins." Prominent cryptocurrencies encompass Ethereum (ETH), Ripple (XRP), Litecoin (LTC), and Cardano (ADA), among other others.

2.1.2 Well-known Cryptocurrencies

There exists a multitude of cryptocurrencies, numbering in the hundreds, however, only a select few have garnered substantial levels of popularity, recognition, and market capitalization [21-24]. The following list comprises many widely recognized cryptocurrencies:

Bitcoin (BTC): Bitcoin is widely recognized as the first and preeminent kind of digital currency. As the first cryptocurrency [23], Bitcoin has set the standard for subsequent cryptocurrencies. Many emerging digital currencies refine key attributes such as anonymity and transaction speed, building on the foundational BTC framework [23].

Ethereum (ETH): ETH is the second-largest cryptocurrency by market capitalization, accounting for nearly 19.02% of the market as of July 2023. It has built a vast Ethereum ecosystem. The introduction of the programmable blockchain concept enabled developers to construct a diverse array of apps on the blockchain platform.

Ripple (XRP): Stands as the third-largest independent cryptocurrency globally. Despite fewer forks compared to BTC and ETH, its market value and extensive application in cross-border payments have established a significant reliance on it within the cryptocurrency market.

Binance Coin (BNB): The Binance Coin serves as the indigenous digital currency of the Binance exchange, which is widely recognized as one of the most prominent cryptocurrency exchanges globally. BNB serves the purpose of facilitating trading fee reductions on the Binance platform and enables participation in token sales conducted on Binance Launchpad.

Cardano (ADA): Cardano is a blockchain platform that endeavors to furnish a heightened level of security and scalability in order to facilitate the creation and implementation of smart contracts and decentralized applications (DApps). The approach prioritizes the utilization of research-based methodologies and protocols that have undergone rigorous evaluation by experts in the field [22].

Solana (SOL): Solana is a blockchain platform that has gained recognition for its exceptional transaction speeds and cost-effectiveness [22]. The primary objective is to provide extensive support for decentralized applications and cryptocurrencies.

Polkadot (DOT): Polkadot is a blockchain platform that facilitates interoperability and information sharing among many chains. The primary objective of this initiative is to establish a web infrastructure that is both decentralized and interoperable [22].

Litecoin (LTC): Litecoin is frequently denoted as the "silver counterpart to Bitcoin's gold." In numerous aspects, it has resemblance to Bitcoin, although distinguishes itself through expedited transaction confirmation durations and utilization of an alternative hashing technique [24].

Chainlink (LINK): Chainlink is a decentralized oracle network that endeavors to furnish precise and dependable data inputs to smart contracts across diverse blockchain platforms [24].

Dogecoin (DOGE): Dogecoin began as a meme cryptocurrency but has now grown in popularity and community. It is frequently used for tipping and charitable contributions [24].

Tether (USDT): Tether is a type of stablecoin that is designed to maintain a fixed value relative to a specific fiat currency, such as the United States Dollar. Frequently employed for commercial transactions and serving as a reliable medium for preserving wealth within the cryptocurrency ecosystem [24].

These aforementioned instances represent a limited selection of widely recognized cryptocurrencies, with BTC, ETH, and XRP being particularly noteworthy as crucial players in the market, holding significant positions, and contributing unique value through their distinct technologies. However, it should be noted that the cryptocurrency domain is subject to continuous transformation, marked by the frequent introduction of new initiatives and tokens. Engaging in comprehensive research and exercising caution is imperative when investing in or utilizing cryptocurrencies, given the highly speculative and volatile nature of the market.

Furthermore, there is interdependence in the cryptocurrency market, meaning the prices of different cryptocurrencies often mutually influence each other, known as market interdependency [9]:such as the dependence of derivative coins on parent cions [10] and the dependence of cryptocurrency market on dominant coins [11]. It is well-known that BTC, ETH, and XRP are the three dominant cryptocurrencies [12]. Therefore, when modeling, it is crucial to consider that the features of these dominant currencies may have a ripple effect on other cryptocurrencies, further enhancing the robustness of the model.

2.1.3 Challenges for Cryptocurrency's Price Prediction

The challenge of predicting the price of cryptocurrencies has distinct difficulties owing to their fundamental attributes and the intricate dynamics of financial markets. The following are key issues that are commonly connected with prediction of cryptocurrency prices [25-28]:

1. Volatility: Cryptocurrencies are notorious for their extremely volatile prices. Prices can fluctuate significantly and rapidly over brief time periods because of influenced by various factors such as market sentiment, technical elements, supply and demand dynamics, and community activities and so on, making accurate forecasting difficult [25].

2. Lack of Regulation: The regulatory framework surrounding cryptocurrencies is now undergoing continuous development in numerous jurisdictions. The influence of regulatory pronouncements or modifications on pricing can be significant, and accurately forecasting regulatory advancements poses a considerable challenge [26].

3. Market Sentiment: The prices of cryptocurrencies are significantly impacted by various factors, including market mood, media exposure, debates on social media platforms, and endorsements from prominent individuals. The analysis of sentiment is a multifaceted task that is susceptible to abrupt fluctuations [27].

4. Limited Historical Data: Numerous cryptocurrencies have comparatively brief histories compared to traditional assets, making it difficult to construct robust long-term predictive models [27].

5. Data Noise: The data sources for cryptocurrencies can be chaotic and unreliable, with discrepancies between exchanges and sources. Providing clear and accurate data for modeling is difficult [28].

6. Model Complexity: Numerous factors can influence cryptocurrency markets, and building accurate prediction models requires a comprehensive comprehension of both the cryptocurrency market and financial market dynamics in general [28].

7. Technical Analysis Limitations: The application of technical analysis for price predition is widespread; nevertheless, its efficacy may be constrained inside markets characterized by high volatility and speculation, such as cryptocurrency [28].

2.2 Generic Price Prediction Method for a Single Cryptocurrency

Here are some of the general steps involved in predicting the price of a single cryptocurrency [29-31]:

Data collection: The first step is to collect the historical price and volume data for the cryptocurrency. This data can be obtained from cryptocurrency exchanges, market data providers, or other sources.

2. Data preprocessing: The data needs to be cleaned and preprocessed to remove any outliers, missing values, or other anomalies [29]. This step also involves selecting the appropriate time interval and data resolution for the analysis.

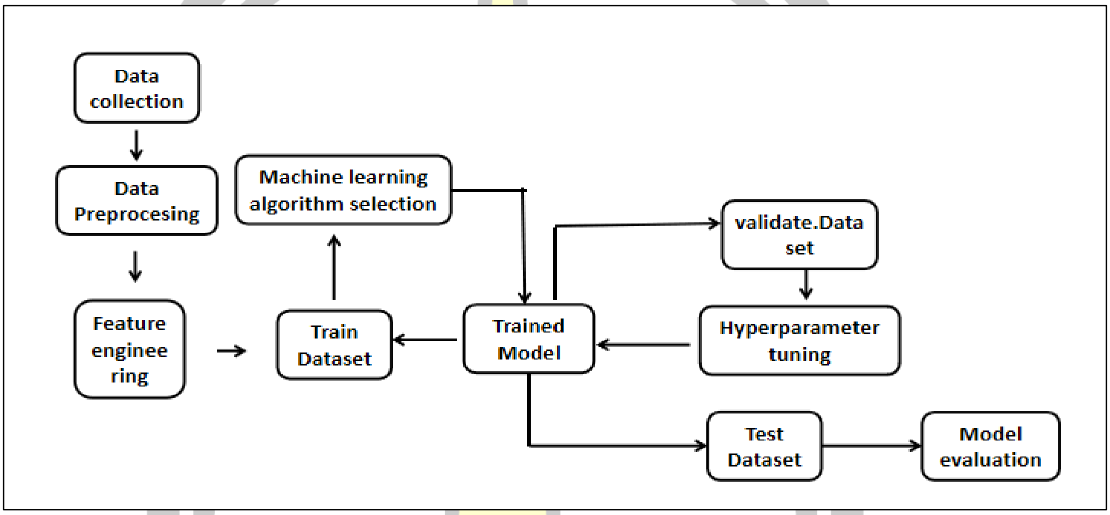


Figure 2.1 General steps in predicting the price

3. Feature engineering: The next step is to extract relevant features from the data, such as the daily closing price, trading volume, volatility, and other technical indicators [29]. Additional features, such as news sentiment, social media activity, and market trends, can also be included.

4. Machine learning algorithm selection: There are various machine learning algorithms that can be used for cryptocurrency price prediction, such as linear regression, time series forecasting, and neural networks. The choice of model depends on the characteristics of the data and the accuracy requirements of the prediction.

5. Model training: The model needs to be trained using the historical data, which involves splitting the data into training and testing sets, often in an 8:2 ratio. The model is trained on the training data.

6. Hyperparameter tuning: The performance of the model can be further improved by tuning the hyperparameters, such as the learning rate, regularization parameter, and model architecture. This step involves iterating through different combinations of hyperparameters and evaluating the performance on the validation set.

7. Model evaluation: The final step is to evaluate the performance of the model on the testing data. This involves calculating various performance metrics, such

as mean squared error, mean absolute error, and R-squared, to measure the accuracy of the prediction [31].

The above steps provide a general framework for predicting the price of a single cryptocurrency using machine learning. However, the specific methods and algorithms used may vary depending on the data and application. It is also important to use appropriate feature selection techniques and to validate the prediction results using real-world data.

2.3 Data Fusion

2.3.1 Definition

Data fusion is the act of amalgamating and merging data from several origins in order to provide information that is more precise, inclusive, and dependable [32-34]. The objective of data fusion is to enhance the data's quality by capitalizing on the unique strengths of each data source while simultaneously addressing their respective limitations. An overview of data fusion can be presented as Figure 2.2.

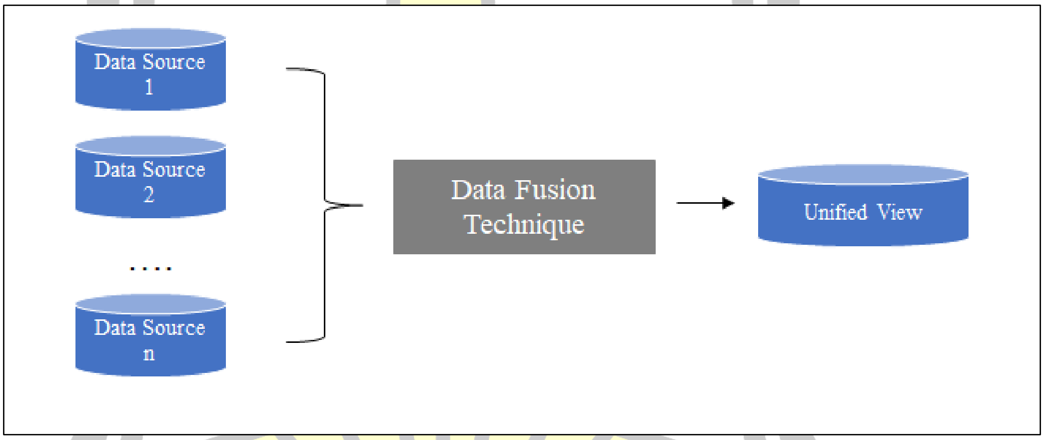


Figure 2.2 Overview of data fusion [32]

1. Input: Various data sources are depicted by icons or symbols, like a satellite to represent remote sensing data, a camera to symbolize surveillance data, and a medical cross to signify healthcare data, among others.

2. Process: The data fusion process involves the establishment of connections between several data sources, facilitating the merging and integration of data.

3. Output: The presentation should aim to provide a cohesive depiction of the amalgamated data, symbolized by an icon such as a globe or an all-encompassing dashboard.

4. Analysis: The diagram depicts a directional arrow originating from the unified representation and extending towards symbols that represent various

analytical techniques, including machine learning models, statistical methods, and others.

2.3.2 Data Fusion approaches

Data fusion approaches can be categorized into multiple different levels, including:

1. Sensor-level fusion refers to the process of amalgamating unprocessed data obtained from many sensors. This task involves the direct combination of raw data from various sensors without undergoing substantial processing. The main objective is to enhance the volume of available data. In the context of a surveillance system, sensor-level fusion refers to the process of combining video feeds obtained from many cameras. The processes are show as Figure 2.3.

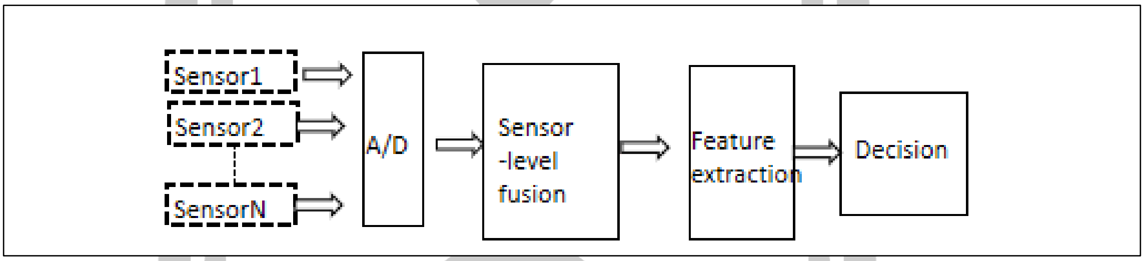


Figure 2.3 The fusion process of sensor-level fusion [30]

2. Feature-level fusion involves the extraction of pertinent features from individual data sources, followed by their amalgamation. The objective of this operation is to extract appropriate features or attributes from each data source and subsequently merge them. The extracted features are frequently employed to display the data in a condensed and significant manner. The processes are show as Figure 2.4.

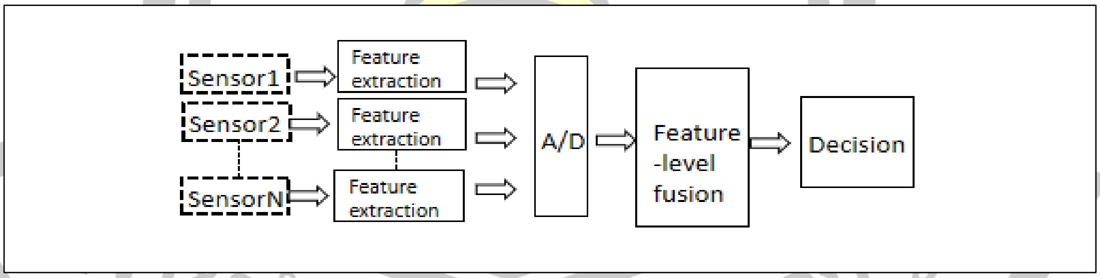


Figure 2.4 The fusion process of feature-level fusion [30]

3. Decision-level fusion refers to the process of amalgamating decisions or conclusions derived from multiple sources [34]. This level integrates decisions or conclusions derived from many sources in order to arrive at a final decision. Each individual source may offer its own assessment, and these assessments are aggregated through diverse methodologies such as voting, averaging, or weighted averaging. In the context of a multi-sensor tracking system, the fusion of outputs from individual

sensors is employed to ascertain the optimal and precise position of a mobile entity. The processes are show as Figure 2.5.

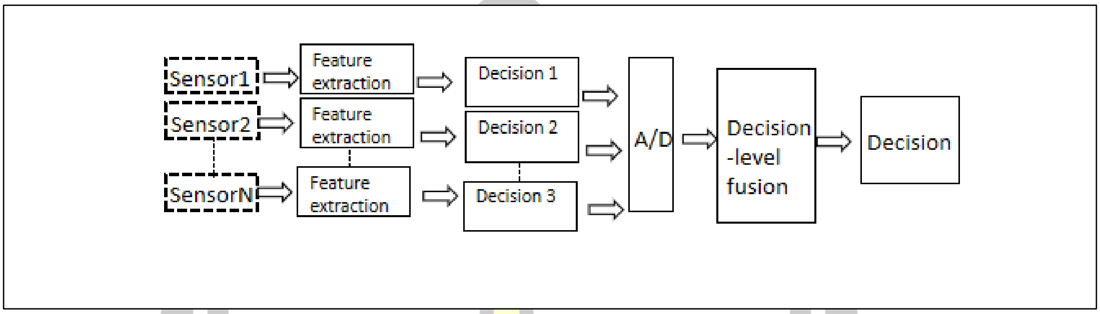


Figure 2.5 The fusion process of decision-level fusion [30]

4. Knowledge-level fusion involves the integration of information at a more advanced semantic level, which may include the utilization of expert knowledge [35]. The processes are show as Figure 2.6.

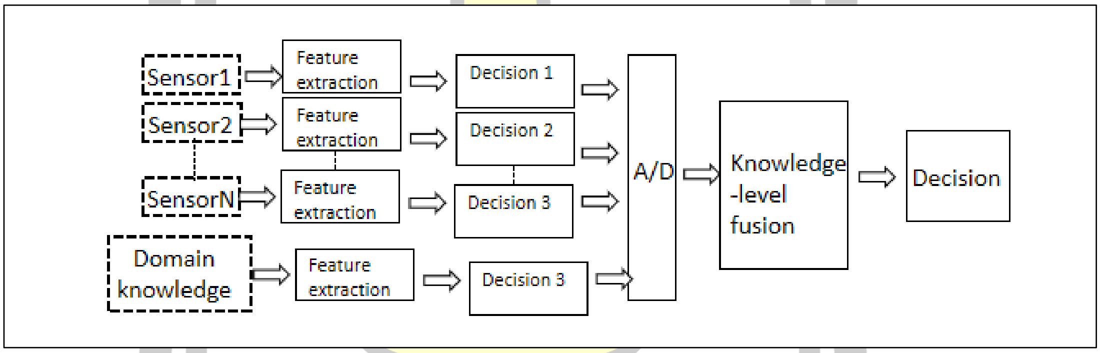


Figure 2.6 The fusion process of knowledge-level fusion [30]

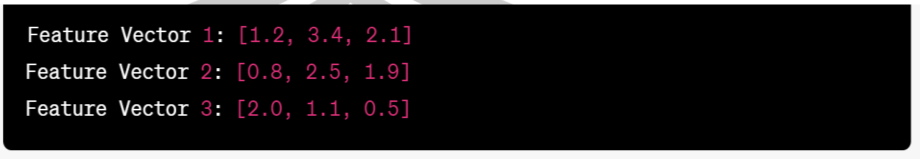
2.3.3 Feature-level fusion for many structured datasets

Feature-level fusion in the context of structured data entails the amalgamation of pertinent features extracted from several structured datasets, resulting in a cohesive representation that encompasses crucial information from all the datasets [36] The objective is to integrate the respective merits of various datasets to achieve a more comprehensive and informative perspective on the data. The following are few prevalent methodologies employed for feature-level fusion of several structured datasets:

1. Concatenation: One of the simplest approaches entails concatenating the features of each dataset along a newly created axis [37, 38]. This methodology proves to be effective when the datasets display similar structures and have common features. For example, if an organization possesses multiple datasets, one of which contains customer data, the features can be consolidated using horizontal

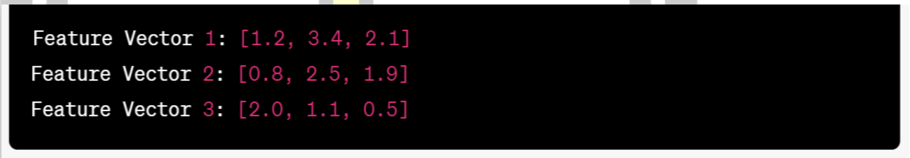
concatenation. Consider following two sets of feature vectors extracted from distinct sources or modalities:

Dataset 1:



```
Feature Vector 1: [1.2, 3.4, 2.1]
Feature Vector 2: [0.8, 2.5, 1.9]
Feature Vector 3: [2.0, 1.1, 0.5]
```

Dataset 2:



```
Feature Vector 1: [1.2, 3.4, 2.1]
Feature Vector 2: [0.8, 2.5, 1.9]
Feature Vector 3: [2.0, 1.1, 0.5]
```

The process involves merging various feature vectors into a single feature vector by concatenating them. The final fused feature vector will incorporate all the features from both data sources. Therefore, concatenated feature vector can be represented as Figure 2.7.

```
[1.2,3.4,2.1,0.8,2.5,1.9,2.0,1.1,0.5,1.2,3.4,2.1,0.8,2.5,1.9,2.0,1.1,0.5]
```

Figure 2.7 A concatenated feature vector

Ultimately, the newly generated feature vector might be employed as input (or training set) for future processing or machine learning processes.

However, it is crucial to note that although concatenation is a simple procedure, it may not always be the most efficient approach. This is especially accurate when the attributes being combined have different sizes or when there are duplications within the collections. Advanced fusion techniques, such as weighted averaging, principal component analysis (PCA), or attention mechanisms, can be utilized in many situations to improve the overall quality of the fused information. The selection of the fusion technique depends on the specific scenario and the fundamental features of the data.

2. Averaging or Aggregation [39]: When datasets contain numerical data, it is feasible to use averaging or aggregation processes to combine the values of common properties. This approach is commonly used to integrate time-series or sensor data from many sources.

Instead of combining the feature vectors, this approach computes the average or another summary statistic of the features from several sources. This approach is particularly advantageous when the features are numerical and indicate a score or measurement. Consider the same sets of feature vectors as in the previous illustration.

For feature-level fusion, averaging is done by calculating the mean value for every element of the feature vectors in both datasets. An example can be illustrated as follows:

[(1.2+0.8+2.0+1.2+0.8+2.0)/6, (3.4+2.5+1.1+3.4+2.5+1.1)/6, (2.1+1.9+0.5+2.1+1.9+0.5)/6]

Finally, a single feature vector generated by taking the average of many feature vectors may be represented as follows.

[1.033, 1.76666666667, 1.5]

Now, a single feature vector which represents the mean features from dataset-1 and dataset-2 is acquired. The average feature vector can be used as input for further processing or machine learning operations.

The use of averaging as a fusion technique is beneficial when features have similar scales and provide different perspectives or measurements of a common underlying concept. Implementing this strategy has the capacity to reduce noise and improve the overall accuracy of the combined data. However, while implementing any fusion technique, it is crucial to thoroughly assess the features of the data and the specific circumstances to determine the most appropriate fusion process.

3. Weighted Fusion [40]: By applying different weights to datasets based on their relative importance or dependability, it is possible that situations could be improved in which some datasets offer more valuable information or are more trustworthy than others. Weighted fusion facilitates the enhancement of the importance attributed to particular datasets throughout the amalgamation procedure. The following serves to illustrate an example. Let's consider two students, Student-A and Student-B in Table 2.1 Weighted fusion of datasets

Table 2.1 Weighted fusion of datasets

Subject	Student-A Score	Student-B Score	Weight
Math	90	85	0.6
Science	80	90	0.4

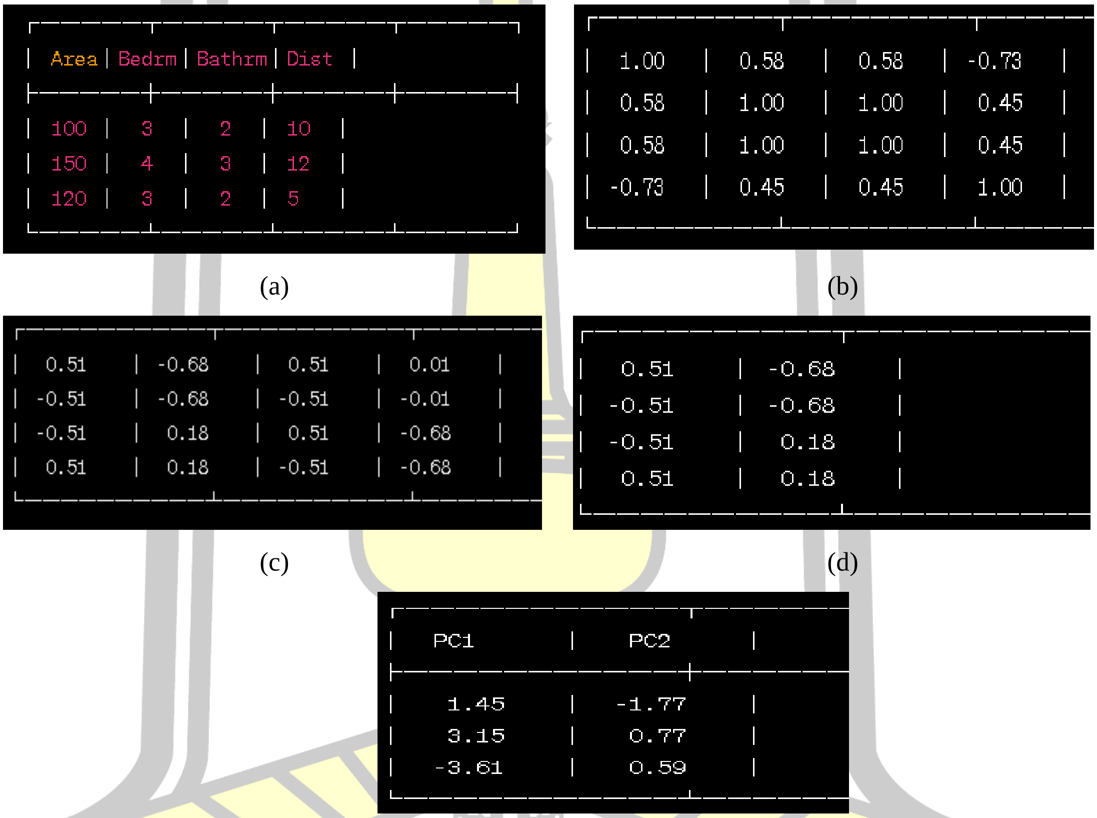
The weighted composite scores for Students-A and B can be determined by multiplying the scores by their corresponding weights and subsequently adding the results.

*Student A Weighted Composite Score = (90 * 0.6) + (80 * 0.4) = 54 + 32 = 86*

*Student B Weighted Composite Score = (85 * 0.6) + (90 * 0.4) = 51 + 36 = 87*

By considering the relative significance of Science and Mathematics, Student-B obtains a greater weighted composite score, which indicates an enhanced level of achievement. Weighted Fusion is a viable approach for incorporating multiple factors with varying degrees of importance; however, it is susceptible to data uncertainty, nonlinear relationships, and instability resulting from subjective weight selection.

4. Feature Selection and Dimensionality Reduction [41]: Before commencing the fusion process, feature selection or dimensionality reduction approaches, e.g. Principal Component Analysis (PCA, may be applied to each dataset in order to determine the most significant and informative features. The identified features have the potential to be incorporated across numerous datasets following the process of selection.



```
┌───────┬───────┬───────┬───────┐
| Area | Bedrm | Bathrm | Dist  |
├───────┼───────┼───────┼───────┤
| 100 |  3  |  2  | 10  |
| 150 |  4  |  3  | 12  |
| 120 |  3  |  2  | 5   |
└───────┴───────┴───────┴───────┘
```

(a)

```
┌──────────┬──────────┬──────────
| 1.00  | 0.58  | 0.58  | -0.73 |
| 0.58  | 1.00  | 1.00  | 0.45  |
| 0.58  | 1.00  | 1.00  | 0.45  |
| -0.73 | 0.45  | 0.45  | 1.00  |
└──────────┴──────────┴──────────
```

(b)

```
┌──────────┬──────────┬──────────
|  0.51  | -0.68  |  0.51  |  0.01  |
| -0.51  | -0.68  | -0.51  | -0.01  |
| -0.51  |  0.18  |  0.51  | -0.68  |
|  0.51  |  0.18  | -0.51  | -0.68  |
└──────────┴──────────┴──────────
```

(c)

```
┌──────────┬──────────
|  0.51  | -0.68  |
| -0.51  | -0.68  |
| -0.51  |  0.18  |
|  0.51  |  0.18  |
└──────────┴──────────
```

(d)

```
┌──────────┬──────────
| PC1     | PC2    |
├──────────┼──────────
|  1.45   | -1.77  |
|  3.15   |  0.77  |
| -3.61   |  0.59  |
└──────────┴──────────
```

(e)

Figure 2.8 The processing step of PCA

As illustrated in Figure 2.8 (a), the subsequent illustration will analyze a dataset consisting of a variety of features associated with multiple houses, including house area, distance to the city center, and number of bedrooms and bathrooms. To reduce the density of the dataset by minimizing the number of dimensions, Principal Component Analysis (PCA) may be applied. The procedures of PCA can be described as follows.

Step 1: Data Standardization (Z-score normalization) – PCA normally starts with the standardization of the data. This process entails adjusting the magnitude of each feature to a mean of zero and a standard deviation of one. PCA requires standardization because it ensures that all variables are treated identically. When variables are assessed using distinct scales, the first principal components would be dominated by those variables that have greater variances and ranges. In general, the Z-score normalization procedure is an essential step in standardizing all features to a common scale, a prerequisite for the PCA algorithm.

Step 2: Computing the Covariance Matrix: Subsequently, it is necessary to compute the covariance matrix of the given data. The covariance matrix quantifies the degree of connection between the various variables in the dataset. For example, the covariance of two variables indicates whether there is an opportunity for one variable to increase in conjunction with the other (positive covariance), or to decrease as the other increases (negative covariance). The covariance matrix is visually represented in Figure 2.8 (b).

Step 3: Calculating Eigenvalues and Eigenvectors: The eigenvalues and eigenvectors are then calculated based on the covariance matrix. The eigenvectors of the covariance matrix correspond to the primary components or directions of the data, while the eigenvalues indicate the magnitude of these directions. To simplify, eigenvectors indicate the direction of highest variability, whereas eigenvalues determine the magnitude of such directions. The corresponding eigenvalues are [3.03, 0.00, 0.06, 0.91], and Figure 2.8 (c) illustrates the eigenvectors.

Step 4: Forming Principal Components (or Feature Transformation): The eigenvectors are arranged in decreasing order based on their associated eigenvalues throughout this stage. The order of the eigenvectors (principal components) defines their level of importance. The first principal component is the primary direction along which the data exhibits the most variability. The second principal component represents the direction with the subsequent highest variability, and so on. The selection of the number of primary components is often based on the desired amount of overall variation in the data that they should capture (see in Figure 2.8 (d))

Step 5: Projecting the Original Data: After the original data is projected onto the principal components, it results in transforming the original data into a new set of variables, the principal components, which are uncorrelated and which capture the maximum variance in the data.

5. Ensemble Methods [42]: Ensemble methods in data fusion involves the combination of numerous models or algorithms to enhance the overall performance, robustness, and reliability of the prediction or classification task. The concept is that by combining the predictions or results of numerous models, the ensemble may frequently achieve better performance than any individual model in the group, particularly when the different models are varied and their mistakes are uncorrelated or weakly correlated. Types of ensemble methods are Bagging (Bootstrap Aggregating), Boosting, and Stacking (Stacked Generalization).

- Bagging (Bootstrap Aggregating): This approach involves training numerous models, often of the same type, using different subsets of the training data. The development of these subsets involves randomly selecting samples from the original dataset, allowing for the possibility of selecting the same sample more than once (bootstrap samples). The ultimate prediction is often derived by taking the average of the predictions (for regression tasks) or by determining the majority vote (for classification tasks). Random Forest is a widely used technique that combines decision trees through bagging.

- Boosting: These algorithms iteratively train a sequence of models, with each accomplishing model aiming to correct the mistakes caused by the prior models. The models are trained in a sequential manner, and each model puts more weight on the training instances that were previously misclassified or predicted inaccurately. The final prediction is a weighted sum of the predictions made by the individual models. Some examples of boosting algorithms include AdaBoost, Gradient Boosting Machines (GBM), and XGBoost.

- Stacking (Stacked Generalization): Stacking is the process of training a new model, known as a meta-learner or blender, to combine the predictions made by of several base models. During the initial stage, the foundational models undergo training using the whole training set. Subsequently, the meta-learner undergoes training using the outputs of its foundational models as features. This enables the stacked model to acquire the optimal method of amalgamating the predictions generated by the fundamental models. It is noted that if using stacking, the meta-learner can also learn from the feature-level representations of the data, effectively combining feature-level and decision-level fusion.

6. Feature Engineering [43]: Feature engineering is an essential and crucial step in the data fusion process, especially when combining data from multiple sources. The objective is to convert unprocessed data into significant features that offer a more holistic understanding of the fundamental phenomena and enhance the efficacy of machine learning models. Here is a systematic method to feature engineering for data fusion. Firstly, it is to identify the nature of each data source. The objective of this process is to understand the type of data (numerical, categorical, text, images, etc.) and the domain from which each data source originates. This will help in determining the appropriate feature extraction and transformation techniques. Later, it is to assess data quality by checking for issues like missing values, outliers, and inconsistencies within each data source. It should address these issues early on is crucial for ensuring the quality of the fused dataset.

2.4 Data Preprocessing

It is important to ensure that the data from various datasets are standardized in terms of scale and exhibit comparable distributions. Normalized data allows faster convergence [44], the performance is show as Figure 2.9. To achieve this, one can apply normalization and scaling techniques such as Min-Max scaling, Z-score normalization, and Log Transformation.

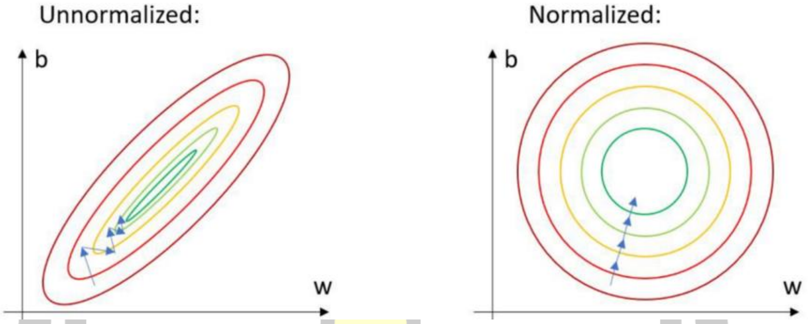


Figure 2.9 Normalized data allows faster convergence

From: https://xydida.com/page/2/

2.4.1 Min-Max Scaling

This method scales the data to a predetermined range, usually between 0 and 1 [44]. The min-max scaling formula is show as follow:

$$x' = \frac{x - \min(x)}{max(x) - \min(x)} \tag{2.1}$$

where x is a feature value and min(x) and max(x) are the approximate lower and upper bounds on the dataset with few or no outliers, respectively.

This normalization method guarantees that the dataset's minimum value is converted to zero, its maximum value is converted to one, and all other values are scaled linearly in between. Due to its reliance on maximum and minimum values, Min-Max Scaling is exceptionally susceptible to outliers.

2.4.2 Z-Score Standardization

The data are transformed by Z-score standardization to have a mean of 0 and a standard deviation of 1 [44]. The formula for standardizing z-scores can be show as:

$$x' = (x - u)/SD \tag{2.2}$$

where x is considered as a feature value, while (SD) is a standard deviation and u is a mean. For the purpose of illustration, suppose we possess the subsequent two datasets:

Height datasets (in centimeters): [*160, 170, 155, 180, 175*]

Weight datasets (in kilograms): [*65, 75, 60, 90, 80*]

We will utilize the Z-score normalization method to standardize these datasets. First, we need to calculate the mean (u) and standard deviation (SD) for each dataset.

For the Height (H) datasets,

$$u_H = \frac{160+170+155+180+175}{5} = 168$$

$$SD_H = \sqrt{\frac{(160-168)^2+(170-168)^2+(155-168)^2+(180-168)^2+(175-168)^2}{5}} \approx 9.27$$

For the Weight (W) datasets,

$$u_w = \frac{65+75+60+90+80}{5} = 74$$

$$SD_w = \sqrt{\frac{(65-74)^2+(75-74)^2+(60-74)^2+(90-74)^2+(80-74)^2}{5}} \approx 10.95$$

The subsequent step is to calculate the standardized values using the Z-score formula; the outcomes are presented in Table 2.2.

Table 2.2 Z-score of height and weight

H(cm)	μ (cm)	SD (cm)	Z-score (H)	W(kg)	μ(kg)	SD (kg)	Z-score (W)
160	168	9.27	−0.86	65	74	10.95	−0.82
170	168	9.27	0.22	75	74	10.95	0.09
155	168	9.27	−1.40	60	74	10.95	−1.28
180	168	9.27	1.3	90	74	10.95	1.47
175	168	9.27	0.76	80	74	10.95	0.55

By applying Z-score standardization, the height and weight data are converted into distributions characterized by a mean of zero and a standard deviation of one. This facilitates a fairer amalgamation and comparison of information derived from diverse datasets. However, the sensitivity of the Z-score to outliers may have an effect on the outcome of standardization.

2.4.3 Log Scaling

This method uses the logarithm (log) of floating-point feature values to convert data range into a limited data range [45] It can employ the following formula.

$$x' = \log(x) \tag{2.3}$$

where x denotes the value of the feature. It is important to utilize caution when implementing log transformation, particularly when dealing with zero or negative values. Such cases may necessitate the addition of a constant or a data shift prior to executing the transformation.

The mentioned normalization methods are applicable to various aspects of cryptocurrency data, such as price, trading volume, market capitalization, and any other relevant attributes. The selection of the normalization method is dependent upon the specific features of the data and the specifications of the employed predictive or analytical model.

2.5 Predictive Algorithms for Cryptocurrency Price

Several machine learning algorithms can be used to model cryptocurrency price predictors. Here are some prevalent algorithms.

2.5.1 Support Vector Regression (SVR)

SVR is a variant of support vector machines (SVM) that is suitable for regression tasks [46, 47]. It seeks to identify a hyperplane that maximizes the profit margin while minimizing the difference between predicted and actual prices. SVR can handle nonlinear relationships and is useful for predicting the price of cryptocurrencies.

The SVR model optimizes a cost function to find the hyperplane that best fits the training data while controlling the trade-off between model complexity and error tolerance when training. Typically, this optimization process entails solving a quadratic programming problem. Therefore, SVR models have a number of hyperparameters that must be tuned for optimal performance. The regularization parameter (C), kernel type (linear, polynomial, radial basis function), kernel-specific parameters (e.g., degree, gamma), and the epsilon parameter, which controls the size of the error tolerance zone are all examples. To find the best combination of hyperparameters that yields the best performance on the validation set, use techniques such as grid search or random search.

However, SVR models, like any other predictive model, have limitations and may not capture the full complexities of cryptocurrency price movements.

Cryptocurrency markets are highly volatile and influenced by a variety of factors, making accurate forecasting difficult. It is recommended that SVR be used in conjunction with other techniques such as ensemble methods or time series analysis, and that the model be continuously monitored and adjusted as market conditions change. Furthermore, use caution when making investment decisions based on predictive models and consider them as part of a broader investment strategy.

Furthermore, SVR models also exhibit sensitivity to hyperparameters and can incur significant computational costs. When confronted with large-scale datasets, SVR models may encounter substantial computational burdens. In high-dimensional scenarios, the interpretability of SVR predictions could prove challenging, particularly for intricate nonlinear relationships.

In SVR, it is trying to find a function f(x) which can be linear or non-linear. The linear function is of the form:

$$f(x) = w * x + b \tag{2.4}$$

where w is the weight vector, x is the feature vector, and b is the bias. The goal of SVR is to make sure that the errors do not exceed the threshold ε, but also attempts to reduce model complexity by minimizing the value of $||w||$.

The SVR employs the ε-insensitive loss function, which means that errors below a certain threshold ε are considered acceptable and not penalized. The loss function is:

$$L_{\varepsilon}(y, f(x)) = max(\, 0, |y - f(x)| - \varepsilon \tag{2.5}$$

The optimization problem for the linear SVR is formulated as:

$$\text{Minimization:} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i{}^*) \tag{2.6}$$

This formula uses to keep the model as flat as possible. For each data point i,

$$\text{Subject to:} \quad \begin{aligned} y_i - (w \cdot x_i + b) &\leq \varepsilon + \xi_i \\ (w \cdot x_i + b) - y_i &\leq \varepsilon + \xi_i^* \end{aligned} \tag{2.7}$$

where $\xi_i$and $\xi_i^*$are slack variables allowing for errors larger than ε.

In order to address this optimization problem, Lagrange multipliers are frequently employed to convert the problem into its dual formulation, thereby facilitating its manipulation and permitting the incorporation of non-linear kernels.

To solve the dual problem, which involves Lagrange multipliers $\alpha_i$and α_i^* for each constraint, the dual objective function must be maximized under specific conditions. The decision function is modified to reflect the optimal weight vector w and bias b, which are obtained from the dual problem solution.

$$f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(x_i, x_j) + b \tag{2.8}$$

where $K(x_i, x)$ is the kernel function, while α_i and α_i^* are the Lagrange multipliers.

In order to extend SVR to non-linear functions, the kernel technique is utilized to map the original features to a higher-dimensional space. By performing the inner product of two coordinates in the feature space, a kernel function $K(x_i, x_j)$ enables the algorithm to accommodate the maximum-margin hyperplane in a feature space that has been transformed. Polynomial, radial basis function (RBF), and sigmoid kernels are prevalent.

2.5.2 Random Forest (RF)

Random Forest is an ensemble learning algorithm that combines multiple decision trees to make predictions. Each tree in the forest is trained on a random subset of the data [48], and the final prediction is obtained by averaging the predictions of each tree. Random Forest can handle complex relationships, feature interactions, and noisy data, which makes it a popular choice for predicting the price of cryptocurrencies.

In order to train the RF Model, individual decision trees are trained separately on different subsets of the data. The forecasts of these trees are then integrated using majority voting (for classification) or averaging (for regression) to provide the final prediction [49]. Furthermore, RF models require the specification of certain hyperparameters, like the number of trees, the maximum depth of the trees, and the amount of features examined for each split, to control the learning process. By employing methods like grid search or random search, adjust hyperparameters to discover the most favorable combination that produces the highest performance on the validation set.

It is crucial to acknowledge that RF models possess inherent limitations. RF, in comparison to SVM, often exhibit more regular and uncomplicated decision limits, which might potentially restrict their effectiveness in dealing with intricate and nonlinear patterns. For enhanced prediction accuracy, it is advisable to augment Random Forest models using other approaches such as fundamental analysis, sentiment analysis, or time series modeling. Moreover, it is crucial to consistently observe and adjust the model when market circumstances fluctuate, and show prudence when making investment choices relying on predictive models.

There are several steps for developing random forest. Each step can be described as follows.

1. Bootstrap Sampling - Initially, it generates several bootstrap samples from the original dataset of size N. Every bootstrap sample is also of size N, generated by randomly selecting instances with replacement from the original dataset. Consequently, certain instances have the potential to be duplicated in every bootstrap sample.

2. Building Decision Trees - Each bootstrap sample is used to construct a decision tree. At each node:

(1) The algorithm randomly chooses a subset of features (variables) without replacement.

(2) Subsequently, it selects the optimal division from the collection of characteristics at the node.

(3) Finally, the tree is expanded to its maximum size without any pruning. This implies that little effort is exerted to prevent a tree from excessively fitting the training data, a phenomenon known as overfitting.

3. Prediction - In regression tasks, the random forest's prediction is determined by calculating the average of the predictions made by each individual tree. If the random forest consists of M trees, and each tree gives a prediction Yi(x) for an input x, the final prediction of the random forest is:

$$Y_{RF}^{reg}(x) = \frac{1}{M}\sum_{i=1}^{M} Y_i(x) \tag{2.9}$$

Important parameters are number of trees (M), number of features at each split, depth of trees, and bootstrap sample size.

- *Number of Trees (M):* Increasing the number of trees can enhance the resilience of a forest, yet the benefits will reach a maximum level at a particular threshold. An excessive number of trees might lead to longer calculation time. In classification task, 100-1000 trees can be a good starting point. However, it might need more trees than in classification, so starting with a higher number and using cross-validation to find the optimal number is recommended.

- *Number of Features at Each Split:* Usually, a smaller collection of features is selected for division at each node. The square root of the total number of features is commonly employed for classification issues. For regression issues, a subset of features comprising one-third of the entire number may be employed. In classification task, a common default option is to select $\sqrt{n}$, where n is the total number of features. Consequently, if you possess a total of 100 features, you would evaluate 10 of them at each split. However, in regression task, a common choice is to use $\frac{n}{3}$ (one-third of the total number of features) for regression tasks. Similar to classification, this is based on empirical results and is a starting point.

- *Depth of Trees:* Although trees are often allowed to grow to their maximum depth, it might be advantageous to regulate tree depth in some situations to avoid overfitting.

- *Bootstrap Sample Size:* The size of the bootstrap sample can impact the bias and variance of the random forest.

In summary, the random forest method is very adaptable, capable of effectively handling regression and classification problems. Additionally, it excels in scenarios with missing values and extensive feature spaces. The use of randomization via bootstrap sampling and feature selection enhances the model's resilience against overfitting in comparison to a solitary decision tree.

2.5.3 Gradient Boosting (GB)

Gradient Boosting algorithms, such as Gradient Boosting Regression (GBR) or XGBoost, iteratively build an ensemble of weak prediction models [50]. Each subsequent model is trained to correct the mistakes of the previous model, leading to a precise and robust prediction model. Gradient Boosting algorithms are widely recognized for their exceptional predictive capability and have been effectively utilized in forecasting the prices of cryptocurrencies.

In order to train the GBR Model, each subsequent tree is trained to forecast the residuals, which are differences between the anticipated and actual values, from the preceding trees. The sequential training method enhances the predicted precision of the model. The learning process in GBR models is governed by many hyperparameters, including the number of trees, learning rate, maximum depth of the trees, and regularization parameters. By employing methods like grid search or random search, adjust hyperparameters to discover the most favorable combination that produces the highest performance on the validation set.

Despite its efficacy and versatility, the Gradient Boosting Regression (GBR) technique is not without its constraints. Gradient Boosting Regression (GBR) is susceptible to overfitting, very responsive to hyperparameters, demands substantial processing resources, is unsuitable for high-dimensional sparse data (e.g., text data), and is particularly sensitive to outliers.

The objective function in XGBoost consists of two parts: the loss function and the regularization term. For a set of n training samples and K trees, the prediction y_i for a sample i is the sum of the predictions from all trees:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \qquad (2.10)$$

where f_k is the prediction from tree k and x_i is the feature vector for the i-th sample.

The objective function L that XGBoost tries to minimize is given by:

$$L(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \qquad (2.11)$$

where $l(y_i, \hat{y}_i)$ is a differentiable convex loss function that measures the difference between the prediction $\hat{y}_i$ and the target y_i. $\Omega(f_k)$ is the regularization term. For a f_k, it can be defined as:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\|\omega\|^2 \qquad (2.12)$$

where T is the number of leaves in the tree, ω are the scores on the leaves, γ is the complexity control on the number of leaves, and λ is the L_2 regularization term on the leaf scores.

For learning process, XGBoost builds the model in an additive manner. For the *t*-th tree, the model is updated as:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \qquad (2.13)$$

XGBoost improves the model by adding a tree f_t that best reduces the objective L.

For tree learning, XGBoost employs a greedy method to include a new tree. The construction of each tree follows a top-down approach. The process begins with a solitary leaf and continues by repeatedly dividing each leaf into two separate leaves. The optimal split is determined by maximizing the gain, which is calculated as follows:

$$Gain = \frac{1}{2}\left[\frac{{G_L}^2}{H_L + \lambda} + \frac{{G_R}^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \gamma \qquad (2.14)$$

where G_L are the sums of gradients for the left and right nodes after the split. H_L and H_R are the sums of the second order gradients for the left and right nodes after the split. λ and $\gamma$are regularization parameters.

Optimizing XGBoost's hyperparameters, such as the number and depth of trees, learning rate, and regularization terms, is essential for maximizing performance.

The resilience, efficiency, and adaptability of XGBoost have contributed to its widespread adoption in many regression and classification problems.

2.5.4 Long Short-Term Memory (LSTM)

The LSTM, or Long Short-Term Memory, is a kind of recurrent neural network (RNN) that is particularly effective at analyzing sequential data. This model has the ability to detect and analyze long-term relationships and trends in the time series data of cryptocurrency prices. The ability of LSTM models to retain and integrate prior knowledge into their predictions enables the modeling of intricate pricing fluctuations [51] LSTM networks excel in capturing long-term relationships and discerning patterns in cryptocurrency price data by efficiently regulating the information flow through memory cells. This allows them to construct forecasts using past data and capture market patterns and intricate connections.

In order to employ LSTM for cryptocurrency price forecasting, the LSTM network is provided with historical price data, as well as other pertinent factors such as trade volume, market indicators, or sentiment analysis [52]. Subsequently, the network undergoes training to discern the fundamental patterns and connections among these characteristics and the goal variable (future price). Once trained, the LSTM model can accurately forecast future price movements based on the patterns it has learnt.

The efficacy of LSTM models for cryptocurrency price prediction is impacted by several aspects, including the quality and amount of historical data, feature selection, model architecture, and hyperparameter tuning [53]. It is crucial to consider these factors. Accurate and reliable predictions need experimentation and improvement.

For the LSTM structure, it consists of four parts to control the flow of information. Given an input sequence x_t, the LSTM unit updates for timestep t are as follows:

1) Forget Gate - This gate determines which information will be discarded from the cell state. The main formula is:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.15}$$

where σ denotes the sigmoid function, W_f is the weight matrix for the forget gate, $b_f$is the bias term, h_{t-1} is the previous hidden state, and x_t is the input at the current timestep.

2) Input Gate and Candidate Cell State – This gate plays critical roles in determining what new information should be stored in the cell state. They work

together to update the cell state in a way that enables the LSTM to capture and retain relevant information over long sequences. The main formula is:

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \end{aligned} \tag{2.16}$$

where W_i and W_C are the weight matrices, and b_i and b_C are the bias terms for the input gate and candidate cell state respectively.

3) Cell State Update - The cell state update mechanism is at the heart of the LSTM's ability to process and learn from sequence data. It allows the network to maintain a stable and controlled memory of past information, which it can use to make accurate predictions or decisions based on the temporal context in the data. The main formula is:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{2.17}$$

where the old cell state C_{t-1} is updated to the new cell state C_t The forget gate decides what to keep from the old state, and the input gate decides what to add from the new candidate values.

4) Output Gate and Hidden State – The output gate and hidden state are crucial for determining what information from the cell state should be outputted at each timestep, and they form the final part of the LSTM's internal process during a timestep. Simply speaking, this gate determines the output based on the cell state. The main formula is:

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * tanh(C_t) \end{aligned} \tag{2.18}$$

where W_o is the weight matrix, and b_o is the bias term for the output gate. The hidden state W_z is based on the cell state, but filtered by the output gate.

2.5.5 Gated Recurrent Unit (GRU)

GRU is a specific architectural design used for recurrent neural networks (RNN). The purpose of GRU is to overcome certain limitations of traditional RNNs, such as the issue of vanishing gradients and the inability to capture long-term relationships [51, 52]. Every individual unit or cell inside a GRU possesses gating mechanisms that regulate the movement of data across the whole network. The gating mechanisms consist of an update gate and a reset gate, which control the movement of data between the previous time step and the current input. The GRUs has the capability to selectively update and discard information, which allows them to effectively record long-term relationships and process sequential data with more efficiency. GRUs have a wide range of applications, including sequence modeling, language translation, speech recognition, and time series analysis. GRUs may be employed to simulate the temporal dynamics of bitcoin price data and identify temporal patterns and relationships when forecasting cryptocurrency values. GRUs have the ability to forecast future price changes by examining previous price sequences and identifying patterns in the training data [51].

Training the GRU Model entails improving the model's parameters by minimizing a loss function, such as mean squared error (MSE), using backpropagation and gradient descent methods. The training method seeks to minimize the disparity between the anticipated and actual prices in the training dataset. Typically, factors such as the number of GRU layers, the number of hidden units in each layer, the learning rate, and regularization can have a substantial impact on the performance of a GRU model [52]. To determine the most effective combination of hyperparameters that produces the highest performance on a validation set, employ methodologies such as grid search or random search.

It should be emphasized that predicting cryptocurrency prices is a challenging endeavor because to the market's substantial volatility and unpredictability. Although GRU models excel at capturing temporal relationships and patterns in data, they may not effectively capture sudden market movements or transient variations. Therefore, it is advisable to utilize GRUs in combination with other methodologies, such as ensemble approaches or fundamental analysis, and to consistently oversee and modify the model as market circumstances evolve. Additionally, exercise prudence and regard prediction models as a component of a broader investing plan when making investment selections.

The structure of GRU consists of four components. They are update gate, reset gate, current memory content, and final memory at current timestamp. Given an input sequence x_i, the GRU unit updates for timestep t are as follows:

1) Update Gate – The Update Gate in a GRU is a critical component that regulates the transmission of information inside the GRU cell. The Update Gate serves the aim of quantifying the amount of past information (from previous states) that should be sent to the future, hence determining the extent to which the prior hidden state should influence the current hidden state.

$$\mathrm{Z_t} = \sigma(\mathrm{W_z} \cdot [h_{t-1}, x_t] + \mathrm{b_z}) \tag{2.19}$$

where W_z is the weight matrix for the update gate, b_z is the bias term, h_{t-1} is the hidden state from the previous time step, and x_t is the input at the current time step. The sigmoid function σ outputs values between 0 and 1.

2) Reset Gate – The Reset Gate in a GRU plays a vital role in defining the extent to which old information should be disregarded or merged with the present input to impact the current state. It collaborates with the Update Gate to efficiently regulate the transmission of data inside the GRU cell.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{2.20}$$

where W_r is the weight matrix for the reset gate, and b_r is the bias term. The reset gate decides how much of the past information to forget.

3) Current Memory Content – The Current Memory Content, often known as the Candidate Hidden State ($\tilde{h}_t$) in the context of GRUs, signifies the suggested updated material for the hidden state at the present timestep. The output is determined by the modulation of the reset gate on a combination of the current input and the previous concealed state. The Current Memory Content is crucial in identifying the appropriate information to be kept in the concealed state.

$$\tilde{h}_t = tanh(W_n \cdot [r * h_{t-1}, x_t] + b_n) \tag{2.21}$$

where W_n is the weight matrix, and b_n is the bias term for the current memory content. The element-wise multiplication of r_t (reset gate output) and h_{t-1} determines how much of the past information will influence the current memory content.

4) Final Memory at Current Time step – The final memory at the current timestep in a GRU is the output of the GRU for that particular timestep, which also serves as the hidden state passed to the next timestep in the sequence.

$$h_t = (1 - z_t) * h_t + z_t * h_{t-1} \tag{2.22}$$

The update gate z_t decides how much of the old hidden state h_{t-1} should be kept and how much of the new candidate hidden state $\tilde{h}_t$ should be used to create the final hidden state h_t for the current time step.

2.5.6 Gaussian Processes (GPs)

Gaussian Processes (GPs) are probabilistic models that may include the uncertainty associated with predictions [47]. They possess the ability to adjust and handle non-linear connections without making assumptions about certain functional patterns. GPs are particularly useful when dealing with limited data or when uncertainty plays an important role in predicting the price of a cryptocurrency.

It is important to acknowledge that although GPs offer a versatile and probabilistic framework for modeling time series data, effectively projecting bitcoin values is challenging given to the market’s extreme volatility and non-linear characteristics. Therefore, it is advisable to utilize GP models in combination with other methodologies and indicators to enhance the precision of predictions and the management of risks. Additionally, it is important to note that previous success does not serve as an indicator of future outcomes, and engaging in cryptocurrency investments has inherent risks.

The formulae related to GPs involve the mean function and the covariance function (or kernel function).

1) Prior: A GP prior can be defined as:

$$f(x) \sim gp(m(x), k(x, x')) \tag{2.23}$$

where $f(x)$ is the function we want to learn, $m(x)$ is the mean function (often assumed to be zero). $k(x, x')$ is the covariance (or kernel) function which defines the covariance between any two-function values $f(x)$ and $f(x^{'})$.

2) Covariance (Kernel) Function - The selection of the kernel function is crucial in Gaussian Process modeling. The encoding incorporates assumptions on the function, such as its smoothness and periodicity. One often used option is the Squared Exponential (SE) kernel:

$$k_{SE}(x, x') = {\sigma_f}^2\, exp(-\frac{1}{2l^2} \parallel x - x' \parallel^2) \tag{2.24}$$

where $\|x - x'\|^2$ is the squared Euclidean distance between the input points, l is the length scale, and σ_f^2 is the signal variance.

3) Predictions - Given a set of training points X with corresponding target values$y$ and a set of test-points X_*, the joint distribution of the observed target values and the function values at the test points under the GP prior is:

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim N(0, \begin{bmatrix} K(X,X) + {\sigma_n}^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}) \tag{2.25}$$

where $K(X, X')$ is the covariance matrix computed by applying the kernel function to the sets of input points, σ_f^2 is the noise variance in the observations, and I is the identity matrix.

GPs offer a practical approach to modeling functions with flexibility, and the selection of the kernel enables practitioners to include their assumptions into the underlying function. One of the main benefits of GP is the incorporation of an uncertainty measure in predictions.

2.5.7 Convolutional Neural Networks (CNN)

CNNs, or Convolutional Neural Networks, are deep learning models frequently utilized for image analysis. However, they may also be applied to predict the value of cryptocurrency [53]. CNNs have the ability to acquire knowledge about patterns and features that are associated with forthcoming changes in prices by seeing price charts as visual representations. CNNs may be highly efficient at identifying predicting visual patterns in price charts.

In order to train the CNN Model, it is necessary to optimize its parameters using an appropriate loss function, such as mean squared error, and a training procedure, such as stochastic gradient descent. The training procedure is designed to minimize the disparity between the projected and actual prices in the training dataset. CNN models are influenced by several hyperparameters, such as the quantity and dimensions of convolutional filters, the pooling approach, learning rate, and regularization parameters. To determine the optimal combination of hyperparameters that produces the highest performance on a validation set, employ methodologies such as grid search or random search.

It is important to note that using CNNs for predicting the price of cryptocurrencies is an unconventional approach whose efficacy depends on the dataset and market dynamics. Cryptocurrency markets are highly complex, influenced by a variety of variables, and susceptible to sudden shifts. To improve prediction accuracy, it is recommended to combine CNNs with other techniques, such as recurrent neural networks (RNNs), ensemble methods, or fundamental analysis. In addition, exercise caution when making investment decisions based on predictive models and incorporate them into a comprehensive investment strategy.

A CNN is composed of many layers that sequentially process volumes of activations using differentiable functions to turn them into new volumes.

1) Input Layer: The input layer will receive the 2D structured data. The configuration of the input layer is contingent upon the dimensions of your structured data.

2) Convolutional Layer: The core building block of a CNN that does most of the computational heavy lifting. The primary function of this layer is to execute a convolution operation, where filters (also known as kernels) are applied to the input in order to generate feature maps. The equation for a discrete convolution process in a two-dimensional space is as follows:

$$(f * g)(i,j) = \sum_{m} \sum_{n} f(m,n) \cdot g(i-m, j-n) \tag{2.26}$$

The input data is represented by the variable f, the kernel is represented by the variable g, and the coordinates in the output feature map are represented by (i, j).

3) Activation Function: Following the convolution process, an activation function is utilized to incorporate non-linear characteristics into the model. The Rectified Linear Unit (ReLU) is the most often used activation function.

$$Re\,L\,U(x) = max(0, x) \tag{2.27}$$

4) Pooling Layer (Pool Layer): This layer minimizes the spatial dimensions (width and height) of the input volume in preparation for the subsequent convolutional layer. By minimizing the number of parameters and processing in the network, it effectively mitigates the problem of overfitting. Max Pooling and Average Pooling are two common types.

- Max Pooling extracts the highest value inside the region of the picture that is encompassed by the kernel.
- Average Pooling computes the mean of all values within the region of the picture that is encompassed by the kernel.

5) Fully Connected Layer (FC Layer): The neurons in a fully connected layer establish connections with all the activations in the previous layer. The output of this layer represents the high-level reasoning in the neural network. After doing matrix multiplication, it is customary to apply a bias offset and an activation function. The operation for a fully connected layer is:

$$y = W_x + b \tag{2.28}$$

where x is the input, W is the weight matrix, $b$is the bias, and y is the output.

6) Softmax or Sigmoid Function (in classification tasks): The final layer often consists of a Softmax function (for multi-class classification) or a Sigmoid function (for binary classification), which yields probabilities indicating the likelihood of the input belonging to a certain class.

$$\begin{gathered} Soft\,max(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \\ Sigmoid(x) = \frac{1}{1 + e^x} \end{gathered} \tag{2.29}$$

7) Output Layer: The output layer for a regression problem (like price prediction) typically has a single neuron. For a classification (like predicting the price direction), it may have one neuron (binary classification) or multiple (multiclass classification).

The parameters of each layer, such as the filters in the convolutional layers and the weights in the fully connected layers, are acquired during training in order to reduce the discrepancy between the anticipated output and the desired target values.

For training predictive model, Mean Squared Error (MSE) is a common loss function. For classification, Cross-Entropy is commonly used. Meanwhile, an optimizer (e.g. Adam, SGD, etc) is used to minimize the loss function. This will involve backpropagation to update the model weights.

2.5.8 Kernel Regression (KR)

Kernel Regression (KR), sometimes referred to as Nadaraya-Watson kernel regression, is a non-parametric method utilized for estimating the conditional expectation of a random variable [54, 55]. It is especially advantageous when the correlation between the variables is intricate and unsuitable for parametric models. Within the realm of cryptocurrency price prediction, KR may be employed to depict the correlation between diverse predictors (such as past prices, trading volume, market mood, etc.) and forthcoming prices.

The formula for KR is relatively straightforward. Given a set of data points (x_i, y_i) where x_i represents the predictors (e.g., features of the cryptocurrency market at time i) and y_i represents the response variable (e.g., the price of the cryptocurrency at time i), the KR estimator $\hat{f}(x)$ at a point x is given by:

$$\hat{f}(x) = \frac{\sum_{i=1}^{n} K_h(x - x_i) y_i}{\sum_{i=1}^{n} K_h(x - x_i)} \tag{2.30}$$

K_h refers to the kernel function, which is a non-negative function that integrates to one and represents the weight. Popular options for the kernel function include of Gaussian, Epanechnikov, and rectangular kernels. The bandwidth parameter, denoted as h, determines the extent or breadth of the kernel. The parameter is critical as it dictates the degree of smoothing: a higher value of h leads to increased smoothing (bias), whereas a smaller h follows the data more closely (variance).

In the context of cryptocurrency price prediction, there are many processing steps for modeling of cryptocurrency price predictor.

1) Feature Selection: It is to choose relevant features x_i (e.g. past prices, volume, technical indicators, market sentiment indicators, etc.) that are predictive variables of predicting of future cryptocurrency prices.

2) Kernel Selection: The purpose is to select a suitable kernel function K. The Gaussian kernel is frequently selected because of its smoothness and localization characteristics.

3) Bandwidth Selection: The selection of bandwidth h is crucial. The selection may be made via cross-validation, aiming to optimize the trade-off between bias and variance.

4) Modeling: In this step, the KR formula is utilized to predict the future price of the cryptocurrency based on the chosen features.

5) Evaluation: Once the cryptocurrency prediction model has been acquired, it is necessary to evaluate its performance using suitable measures. Common metrics used for regression tasks include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

When the data do not satisfy the assumptions of parametric models and the relationship between the variables is complex, kernel regression is especially advantageous. It is essential to note, however, that similar to any other model, this one has its limitations and should be applied with caution, particularly in volatile and unpredictable markets such as cryptocurrency. The selection of parameters (x_i), kernel(K), and bandwidth (h) is critical and can have a substantial influence on the efficacy of the model. Moreover, a multitude of recognized and unidentified variables exert an impact on financial markets; therefore, the utilization of any model should be supplemented with comprehensive market analysis and judgment from experts.

2.6 Evaluation Matrices

Several measurement metrics can be utilized to evaluate the accuracy and effectiveness of a model for predicting the price of a cryptocurrency when evaluating its performance [56, 57]. Here are some common evaluation metrics for predicting the price of cryptocurrencies.

2.6.1 Mean Squared Error (MSE)

MSE measures the average squared difference between the predicted prices and the actual prices [58]. It measures the overall error in prediction, with larger errors contributing more to the final score. Given a set of n datapoint $(y_1, y_2, \ldots, y_n)$ representing the true values and corresponding prediction values, denoted as $(\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)$, the MSE formula can be defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.31}$$

where y_i represents the true value for the i-th data point, $\hat{y}_i$ represents the predicted value for the i-th data point, and n is the total number of data points.

The MSE is a single number that represents the quality of a prediction. A lower MSE suggests a better match with the data. The MSE has the desirable virtue of being expressed in the same units as the function's output. However, one disadvantage is that squaring the mistakes lends more weight to larger errors (outliers), which may not always be desirable depending on the situation.

2.6.2 Mean Absolute Error (MAE)

MAE computes the average absolute difference between predicted and actual prices [59]. It measures the typical magnitude of prediction errors. Similar to MSE, a lower MAE value indicates a more accurate prediction. Given a set of n datapoint $(y_1, y_2, \dots, y_n)$ representing the true values and corresponding prediction values, denoted as $(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n)$, the MAE formula can be defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{2.32}$$

where y_i represents the true value for the i-th data point, $\hat{y}_i$ represents the predicted value for the i-th data point, and n is the total number of data points.

The MAE provides a concise understanding of the average error size. It is robust to outliers since it does not square mistakes. However, depending on the application or environment, it may fail to punish significant mistakes appropriately.

2.6.3 Root Mean Squared Error (RMSE)

The RMSE is a typical method for measuring a model's mistake in predicting quantitative data [59]. It is especially beneficial when huge inaccuracies are highly undesirable. The RMSE is calculated by taking the square root of the average of the squared discrepancies between anticipated and actual values. Large mistakes are given a relatively high weight since they are squared before being averaged. Given a set of n data point $(y_1, y_2, \dots, y_n)$ representing the true values and corresponding prediction values, denoted as $(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n)$, the RMSE formula can be defined as:

1) Compute the Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{2.33}$$

2) Take the square root of the MSE to obtain the RMSE:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.34}$$

where y_i represents the true value for the i-th data point, $\hat{y}_i$represents the predicted value for the i-th data point, and$n$ is the total number of data points.

The RMSE is always non-negative, and a value of 0 represents a perfect fit to the data. In general, a lower RMSE is preferable to a greater one; however, whether a specific RMSE is good or bad depends on the context of the problem and the variability of the target variable. The RMSE uses the same units as the target variable, making it more interpretable than the MSE.

2.6.4 Mean Absolute Percentage Error (MAPE)

The MAPE is a metric for determining the accuracy of a prediction model [59]. It presents accuracy as a percentage, making it simple to understand and especially valuable for comparing the performance of several forecasting models on different sizes. Given a set of n data point $(y_1, y_2, \dots, y_n)$ representing the true values and corresponding prediction values, denoted as $(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n)$, the MAPE formula can be defined as:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{y_i - \hat{y}_i}{y_i} \times 100\% \tag{2.35}$$

where y_i represents the true value for the i-th data point, $\hat{y}_i$ represents the predicted value for the i-th data point, and n is the total number of data points.

MAPE computes the absolute error as a percentage of the actual value, averages these percentages over the dataset to obtain the mean, then multiplies by 100 to describe the result as a percentage. This measure is extensively used since it is straightforward to perceive and comprehend.

However, MAPE has a few limitations. It can be skewed by data points where the actual value y_i is very small, as the error percentage can become very large. It is undefined for data points where $y_i = 0$. Similar to other absolute error metrics, it fails to differentiate between under-prediction and over-prediction.

2.6.5 Directional Accuracy (DA)

DA is a statistic used to evaluate the effectiveness of a predictive model by calculating the proportion of times the model correctly predicts a variable's direction of change. It is especially useful in financial modeling and economics, where anticipating the proper direction of a market or price movement (up or down) is more significant than predicting the precise value. The DA formula can be defined as:

$$DA = \frac{N_{correct}}{N} \times 100\% \tag{2.36}$$

where $N_{correct}$ is the number of times the model correctly predicts the direction of change, and N is the total number of predictions made.

In the context of time series or sequential data (such as stock prices, cryptocurrency prices, etc.), the direction of change is usually regarded proper if:

- The actual value at time $t+1$ is higher than at time t, and the model predicted an increase.
- The actual value at time $t+1$ is lower than at time t, and the model predicted a decrease.

To compute $N_{correct}$, it would typically:

- It calculates the difference between consecutive actual values.
- It calculates the difference between consecutive predicted values.
- It compares the signs (positive or negative) of these differences. If the signs match, the prediction for that step is considered correct.

DA is expressed as a percentage and provides a concise understanding of the model's ability to accurately anticipate the trend or direction. However, it is important to note that, while DA is beneficial for assessing model performance in terms of trend prediction, it does not give information on the magnitude of errors or the model's performance in terms of anticipated values.

2.6.6 Hit Ratio (HR)

The HR is a statistic that is widely used in recommendation systems and finance to assess the accuracy of model predictions [58]. It represents the proportion of times the model correctly predicts an event of interest (a “hit”) out of all forecasts or chances. In finance, for example, it might reflect the proportion of profitable deals to the total number of trades executed. In recommendation systems, it might indicate how many times a suggested item was liked or interacted with by the user. The HR formula can be defined as:

$$HR = \frac{\text{Number of Hits}}{\text{Total Number of predictions or Opportunities}} \times 100\% \tag{2.37}$$

where ‘Number of Hits’ is the number of times the model successfully predicts or recommends the correct outcome, and ‘Total Number of Predictions or Opportunities’ is the total number of predictions made or the total number of opportunities where a prediction could have been made.

The HR is denoted in percentage form. A greater hit ratio signifies superior performance, where 100% denotes absolute accuracy. For example, in a trading algorithm, if the algorithm makes 10 trades, and 6 of those trades are profitable, then

The HR score = (6/10) * 100% = 60%

Although the HR offers a straightforward indication of a model’s precision, it is critical to acknowledge that it fails to consider the significance of successes or the financial impact of failures. It measures only the correctness of the outcomes and not their degree of accuracy.

Nevertheless, while evaluating a model for predicting the price of a cryptocurrency, it is crucial to take into account many assessment criteria in order to have a comprehensive comprehension of its performance. Different metrics capture various facets of the model's accuracy, and it is advisable to choose metrics that align with your specific objectives and needs. Additionally, it is essential to validate the model's performance on out-of-sample data or conduct cross-validation to ensure its generalizability and prevent overfitting to the training data.

2.7 Related Works

Many researchers have attempted to predict time series at different periods using various algorithms. It started with Statistical Methods and moved on to using Machine Learning in Single Models (ML-single models). This progression continued with the adoption of Hybrid Models that combine different Machine Learning techniques (ML-hybrid models). With the high market volatility and instability, data correlations were considered to further enhance the robustness of models, improving the predictive accuracy of the model.

2.7.1 Statistical-Method

During the initial rise of the cryptocurrency market, statistical methods were predominantly applied for predicting cryptocurrency prices. The vector error correction model (VECM), vector auto regression (VAR) model [59, 60], autoregressive integrated moving average (ARIMA) model [61], and neural network autoregressive (NNAR) [62] are basic models for cryptocurrency price time-series. The ARIMA is one of the most representative statistical methods for time series data analysis [62]. However, the above four models are only suitable for predicting stationary time-series data under certain conditions, and their variances are supposed to be constant. Statistical -methods mainly use the linear regression to predict cryptocurrency trends based on stationary linear historical data. However, cryptocurrency prices are non-stationary and non-linear, making it challenging for statistical-methods to capture these complex market dynamic relationships.

2.7.2 Machine Learning -Single Model

Machine learning algorithms with excellent non-linear regression performance are becoming popular methods for cryptocurrency prices prediction. Derbentsev et al. [63] found that random forest (RF), logistic regression (LR), and linear discriminant analysis (LDA) outperformed ARIMA in predicting BTC prices. McNally [64] constructed a BTC price prediction model using recurrent neural networks (RNN) and long short-term memory (LSTM), concluding that nonlinear deep learning methods were superior to ARIMA. Chen et al. [65] employed decision trees (DT) and support vector machines (SVM) to forecast BTC prices, demonstrating that ML methods were generally more suitable for BTC price prediction compared to traditional statistical methods. Mallqui [66] achieved promising results in BTC price prediction using ensemble machine learning algorithms like RNN and tree classifiers. Poongodi [67] compared linear regression (LR) and support vector machine (SVM) for Bitcoin price prediction, with SVM exhibiting higher accuracy (96.06%) than LR (85.46%). Sun et al. [68] showed that gradient boosting outperformed gradient boosting decision tree (GBDT) in predictive performance. Zoumpekas [69] utilized deep learning algorithms to forecast Ethereum cryptocurrency closing prices, with convolutional neural networks (CNN), LSTM, stacked LSTM (SLSTM), Bidirectional LSTM (Bi LSTM), and gated recurrent unit (GRU) showing promising accuracy and experimental profitability in real-time Ethereum closing price prediction. Mudassir [70] employed four machine learning models – artificial neural network (ANN), seasonal artificial neural network (SANN), SVM, and LSTM – for short-term (7 days), medium-term (30 and 90 days) Bitcoin price predictions, with LSTM outperforming others. Several articles have shown that LSTM outperforms traditional machine learning algorithms in cryptocurrency price predictions. LSTM uses a set of memory cells with gate structure to replace hidden neurons of RNN. As such, through the gate structure feature, the information is retained and persistently updated in the following training iterations. So, it has the advantage of solving the gradient explosion and gradient vanishing problems in neural network algorithms. Table 2.3 shows the review performances of ML-single models.

Table 2.3 Review of Machine Learning -Single Model

Ref	Author	Description	Features	Technique	cryptocurrency	Expected Result	Performance
[63]	Derbentsev	Given live streaming BTC activity, aim to forecast BTC.	C	RF, LR, LDA	BTC	Found that RF, LR, and LDA outperformed ARIMA	ML>statistical methods
[64]	McNally	Construct a BTC price prediction model using RNN and LSTM.	OHLC	RNN, LSTM	BTC	Concluding that nonlinear DL were superior to ARIMA	
[65]	Chen Z	Employed DT and SVM to forecast BTC.	Property, trading and market, attention and gold spot price.	DT, SVM	BTC	ML were more suitable	
[66]	Mallqui	The ANN and SVM were employed for regression of the maximum, minimum and closing prices of the BTC	Direction, HL	ANN, SVM	BTC	Showed that SVM achieved an improvement of more than 10% in accuracy	SVM>ANN

Table 2.3 (Cont’)

Ref	Author	Description	Features	Technique	cryptocurrency	Expected Result	Performance
[67]	Poongod	Compared LR and SVM for BTC price prediction	OHLV, weighted, average	LR, SVM	BTC	SVM exhibiting higher accuracy (96.06%) than LR (85.46%)	SVM>LR
[68]	Sun et al.	A novel GBDT, Light GBM, to forecast the price trend of cryptocurrency market.	V, market ranking, circulation	GBDT, Light GBM and RF	cryptocurrency price trend	GBDT, Li GBM outperformed RF in predictive performance	GBDT, Li GBM>RF
[69]	Zoumpekas	Utilized DL algorithms to forecast ETH cryptocurrency closing prices	C	CNN, LSTM, SLSTM, Bi LSTM, GRU	ETH	Showed that LSTM, Bi LSTM, GRU have high accuracy	LSTM, Bi LSTM, GRU>SLSTM, CNN
[70]	Mudassir	Employed four ML models for short-term (7 days), medium-term (30 and 90 days) BTC price predictions	C, direction	ANN, SANN, SVM, LSTM	BTC	LSTM outperforming others	LSTM>ANN, SANN, SVM

2.7.3 Machine Learning -Hybrid Model

Although machine learning has the advantage of non-linear prediction performance in cryptocurrency price prediction, most of the studies only utilize a single prediction model to realize cryptocurrency price prediction, which has limitations in capturing and understanding intricate data representations. To overcome these limitations, researchers turned to hybrid models that combine various machine learning techniques for a more comprehensive analysis of cryptocurrency price prediction. Wu et al. [71] developed a BTC prediction framework using LSTM, presenting two LSTM (traditional and AR (2)-enhanced LSTM) where the AR (2)-enhanced LSTM outperformed the traditional LSTM. Marne et al. [72] used SVM, Multilayer Perceptron, and RNN, highlighting the superior accuracy and efficiency of RNNs with LSTM models for cryptocurrency price forecasting. Kristjanpoller et al. [73] proposed a hybrid volatility prediction model for BTC, incorporating Artificial Neural Networks (ANN), Generalized Auto Regressive Conditional Heteroskedasticity (GARCH), and Principal Component Analysis. Li et al. [74] combined LSTM and CNN structures in a novel hybrid neural network model that utilized Bitcoin transaction data and external factors like macroeconomic variables and investor sentiment, demonstrating improved accuracy over single-structure neural networks. Guo et al. [75] concluded that the MRC-LSTM hybrid model outperforms CNN, LSTM, and CNN-LSTM in predicting BTC, highlighting its superior predictive capability. Patel et al. Hashish [76] proposed a novel hybrid model using Hidden Markov Models to describe historical cryptocurrency movements and LSTM for future trend prediction, outperforming traditional time series models. Patel [77] introduced a hybrid cryptocurrency prediction approach based on LSTM and Gated Recurrent Units (GRU), showing accurate price prediction with high precision, implying potential applicability in predicting prices of different cryptocurrencies. Seabe [78] Bi-LSTM outperforms LSTM and GRU in predicting major cryptocurrency prices with higher accuracy. Hansun [79] investigates the use of LSTM, Bi-LSTM, and GRU for predicting cryptocurrency prices, revealing that while Bi-LSTM perform similarly in accuracy. Birim [80] demonstrates that Bi-LSTM outperforms LSTM, GRU, and Bi-GRU in predicting cryptocurrency prices, highlighting the advantage of bi-directional models in time series forecasting. Table 2.4 shows the review performances of ML -hybrid models.

Table 2.4 Review of Machine Learning -Hybrid Model

Ref	Author	Description	Features	Technique used	Crypto	Expected Results	Performance
[71]	Wu et al.	Developed a BTC price prediction framework using LSTM, AR-LSTM	C and V	LSTM, AR- LSTM	BTC	AR-LSTM outperformed the traditional LSTM	AR-LSTM>LSTM
[72]	Marne et al.	Used SVM, RNN, RNNs-LSTM for cryptocurrency price forecasting.	OHLV, weighted price	SVM, RNN, RNNs - LSTM	BTC	highlighting the superior and efficiency of RNNs-LSTM models	RNNs-LSTM>SVM, RNN
[73]	Kristjanpoller et al.	Propose ANN-GARCH model with preprocessing to forecast the price volatility	Closing price return and volatility	ANN-GARCH-PCA	BTC	The incorporation PCA preprocessing increases the accuracy of the hybrid model.	ANN-GARCH-PCA>ANN-GARCH
[74]	Li Y	Combined LSTM and CNN structures in a novel hybrid neural network model for BTC price prediction	Transaction amount，OHLV	LSTM-CNN	BTC	Demonstrating improved accuracy over single-structure neural networks	LSTM-CNN>CNN
[75]	Guo et al.	Proposed MRC-LSTM hybrid model	OHLC	CNN, LSTM, CNN-LSTM, MRC-LSTM	BTC	BTC (MAE:166.52 RMSE:261.44 MAPE:1.56 LTC (0.14, 0.26, 3.17) ETH (0.70, 1.13， 5.43)	MRC-LSTM>CNN, LSTM, CNN-LSTM

Table 2.4 (Cont')

Ref	Author	Description	Features	Technique used	Crypto	Expected Results	Performance
[76]	Hashish	Proposed a novel hybrid model using HMM to describe historical cryptocurrency movements and LSTM for future trend prediction	OHL	HMM-LSTM	BTC	HMM-LSTM models outperforming traditional time series models	HMM - LSTM>traditional time series models
[77]	Patel et al.	Introduced a hybrid cryptocurrency prediction approach based on LSTM and GRU	Average price, OHL	LSTM-GRU	LTC	LSTM-GRU model better than the LSTM	LSTM-GRU>LSTM
[78]	Seabe	Bi-LSTM outperforms LSTM	OHLV	Bi-LSTM	BTC ETH XRP	Bi-LSTM better than the LSTM	Bi-LSTM >LSTM
[79]	Hansun	Investigates the use of LSTM, Bi-LSTM, and GRU for predicting cryptocurrency prices	OHLV	Bi-LSTM	BTC ETH	Bi-LSTM better than the LSTM, GRU	Bi-LSTM >LSTM, GRU
[80]	Birim	Demonstrates that Bi-LSTM outperforms LSTM, GRU, Bi-GRU	OHLV	Bi-LSTM	BTC ETH XRP XMR	Bi-LSTM better than the LSTM, GRU, Bi-GRU	Bi-LSTM >LSTM, GRU, Bi-GRU

2.7.4 Machine learning based on data fusion

Data fusion is an advanced technology that amalgamates diverse data from multiple sensors monitoring the same entity while filtering out redundant or noisy information from shared data. Given its simplicity and efficiency, financial research often relies on feature-level fusion to develop a preferred model. While the articles under review predominantly center on fusion techniques, a few of them may not have explicitly labeled their methodology as a fusion approach.

Guo et. al. [81] proposed a model of ICA-SVR and CCA-SVR, and utilizes feature fusion tool to extract the union feature. SVR is used to predict the future closing price of stock. All the measure indicators of the ICA-CCA-SVR model are significantly improved after feature fusion. Kim et al. [82] proposed model is composed of LSTM and CNN, which are utilized for extracting temporal features and image features. Our feature fusion LSTM-CNN model outperforms the single models in predicting stock prices. Tanwar et al. [83] introduced a hybrid model comprising GRU and LSTM to estimate LTC and Z-cash close prices, achieving accurate predictions by fusion the directional values of the parent cryptocurrency. Patel [84], taking correlations among cryptocurrencies into account, forecasting Dash prices by fusing the daily average prices of BTC, Dash, and LTC as input features in an LSTM-GRU hybrid model. The results indicate that the proposed model yielded low error and high precision. Maleki et al. [85] introduced an innovative method for predicting BTC prices by fusion features from cryptocurrencies with the highest cointegration. The results show that this model demonstrates high accuracy. However, these models make predictions for one type or a specific category financial product. These models cannot be referred to as unified models since they cannot predict various cryptocurrencies. Table 2.5 shows the review performances of ML based on data fusion.

Table 2.5 Review of Machine Learning based on data fusion

Ref	Author	Description	Features	Technique used	Financial product	Result
[83]	Tanwar et al.	Introduced a hybrid model comprising GRU and LSTM to estimate LTC and Zcash close prices	OHLV, direction of the parent coin	GRU-LSTM, Feature-level fusion	LTC，Z-cash	Achieving accurate predictions by fusion the directional values of the parent cryptocurrency.
[84]	Patel	To forecast Dash prices by fusing the daily average prices of BTC, Dash, and LTC as input features in an LSTM-GRU hybrid model.	OHL, average price of parent coin	LSTM-GRU, Feature-level fusion	Dash	The results indicate that the proposed model yielded low error and high precision
[85]	Maleki et al.	Introduced an innovative method for predicting BTC prices by fusion feature with the highest cointegration.	Price of Zcash	ADF, ARIMA, Feature-level fusion, LR, GBR, SVR, RFR	BTC	Zcash had the most trend's similarity to Bitcoin.

In summary, most of current cryptocurrency price prediction models are limited to forecasting individual cryptocurrencies, relying heavily on the unique features of each cryptocurrency and lacking the ability to predict different types of cryptocurrencies in a unified manner. This is due to challenges faced in using a single model to predict prices of various cryptocurrencies, including market dynamics, distinct cryptocurrency attributes, data variability, and interdependencies between markets. Thus, proposing a unified prediction model for diverse cryptocurrency closing prices remains a challenge.

In this study, we aim to explore and develop a unified model for predicting cryptocurrency closing prices by combining feature fusion techniques (feature-level fusion) with machine learning and deep learning models such as SVR, RF, LSTM, and GRU. This model focuses on various cryptocurrencies, including dominant ones like BTC, ETH, and XRP, as well as derivatives like LTC and independent coins like XMR. We will use evaluation metrics such as MAE, RMSE, and MAPE to compare the performance of the unified model against baseline models. Additionally, this study will conduct an in-depth comparison of the performance of deep learning methods versus machine learning methods. To validate the model's accuracy and stability across different window sizes, we have established both short-term and long-term prediction windows. The ultimate goal of this research is to develop a unified model that integrates the features of dominant cryptocurrencies, enabling accurate predictions of closing prices for various types of cryptocurrencies, thereby providing a comprehensive and reliable price prediction solution for the cryptocurrency market, offering valuable insights for investors and decision-makers.

CHAPTER 3 RESEARCH METHODOLOGY

3.1 Datasets

3.1.1 Data Source

From the aforementioned discussion, we observe the interdependencies among various cryptocurrencies and the influence of three dominant currencies on the market. The primary focus of this study is to utilize the features of these three cryptocurrencies—BTC, ETH, and XRP—to predict the closing prices of a variety of cryptocurrencies. Additionally, this analysis includes derivative currency LTC and independent currency XMR, expanding the scope of the study. To facilitate this analysis, five datasets for cryptocurrencies are employed in this study, which can be downloaded from the website https://finance.yahoo.com/cryptocurrencies/.

The datasets for this study were retrieved from the internet on July 15th, 2023. The cryptocurrency datasets used were downloaded from 9-11-2017 to 11-07-2023. Those datasets are represented in the csv. format. Each cryptocurrency dataset used in this study consists of 2,071 continuous price data points, encompassing eight common features. Each is a complete record without any missing values, ensuring the continuity and reliability of the data, which contributes to enhancing the performance and accuracy of the predictive model. For detailed descriptions of the variables in the dataset, please refer to Table 3.1.

Table 3.1 Dataset specifications

Parameter	Description
Open	Daily opening price of the selected cryptocurrency
High	Daily high price of the selected cryptocurrency
Low	Daily low price of the selected cryptocurrency
Close	Daily close price of the selected cryptocurrency
Volume	Daily Volume of the selected cryptocurrency

These datasets consist of eight common features as cryptocurrency name, time stamp, highest price during the day (high), lowest price during the day (low), adj close price during the day (adj close), the number of assets traded the day (volume), open price (open), and close price (close). However, high, low, volume, and open are used as the predictive variables to learn a mapping from input variables to a target variable (close). Examples of three dominant cryptocurrency datasets are depicted in Figure 3.1, Figure 3.2, and Figure 3.3.

	Date	Open	High	Low	Close	Adj Close	Volume
0	2017/11/9	7446.830078	7446.830078	7101.520020	7143.580078	7143.580078	3.226250e+09
1	2017/11/10	7173.729980	7312.000000	6436.870117	6618.140137	6618.140137	5.208250e+09
2	2017/11/11	6618.609863	6873.149902	6204.220215	6357.600098	6357.600098	4.908680e+09
3	2017/11/12	6295.450195	6625.049805	5519.009766	5950.069824	5950.069824	8.957350e+09
4	2017/11/13	5938.250000	6811.189941	5844.290039	6559.490234	6559.490234	6.263250e+09
...	...	...	...	...	...	...	...
2066	2023/7/7	29907.998050	30434.644530	29777.285160	30342.265630	30342.265630	1.338477e+10
2067	2023/7/8	30346.921880	30374.437500	30080.160160	30292.541020	30292.541020	7.509379e+09
2068	2023/7/9	30291.611330	30427.589840	30085.591800	30171.234380	30171.234380	7.903328e+09
2069	2023/7/10	30172.423830	31026.083980	29985.394530	30414.470700	30414.470700	1.482821e+10
2070	2023/7/11	30423.626950	30643.537110	30375.580080	30493.019530	30493.019530	1.500564e+10

Figure 3.1 Examples of BTC

	Date	Open	High	Low	Close	Adj Close	Volume
0	2017/11/9	308.644989	329.451996	307.056000	320.884003	320.884003	893249984
1	2017/11/10	320.670990	324.717987	294.541992	299.252991	299.252991	885985984
2	2017/11/11	298.585999	319.453003	298.191986	314.681000	314.681000	842300992
3	2017/11/12	314.690002	319.153015	298.513000	307.907990	307.907990	1613479936
4	2017/11/13	307.024994	328.415009	307.024994	316.716003	316.716003	1041889984
...	...	...	...	...	...	...	...
2066	2023/7/7	1847.512573	1876.963257	1832.025391	1870.602539	1870.602539	6468885150
2067	2023/7/8	1871.002075	1872.501587	1844.641724	1865.539551	1865.539551	4299007854
2068	2023/7/9	1865.594971	1878.668945	1857.748291	1863.009766	1863.009766	4392863807
2069	2023/7/10	1863.240234	1905.460815	1848.777222	1880.556396	1880.556396	6336468234
2070	2023/7/11	1879.693970	1887.249512	1876.514893	1877.229370	1877.229370	6060186112

Figure 3.2 Examples of ETH

	Date	Open	High	Low	Close	Adj Close	Volume
0	2017/11/9	0.217911	0.221791	0.214866	0.217488	0.217488	147916992
1	2017/11/10	0.218256	0.219068	0.205260	0.206483	0.206483	141032992
2	2017/11/11	0.205948	0.214456	0.205459	0.210430	0.210430	134503008
3	2017/11/12	0.210214	0.210214	0.195389	0.197339	0.197339	251175008
4	2017/11/13	0.197472	0.204081	0.197456	0.203442	0.203442	132567000
...	...	...	...	...	...	...	...
2066	2023/7/7	0.463715	0.470485	0.460630	0.468765	0.468765	682747263
2067	2023/7/8	0.468757	0.473962	0.466045	0.470644	0.470644	409435365
2068	2023/7/9	0.470644	0.471789	0.468117	0.468550	0.468550	324938269
2069	2023/7/10	0.468554	0.479986	0.464606	0.477664	0.477664	885726455
2070	2023/7/11	0.477685	0.478288	0.471808	0.475719	0.475719	645625637

Figure 3.3 Examples of XRP

3.1.2 Trend of Previous Cryptocurrencies' Analysis

This section provides an analysis of the individual price trends of the three dominant cryptocurrencies: Bitcoin (BTC), Ethereum (ETH), and Ripple (XRP). The analysis focuses on how these currencies influence each other and how they collectively affect the broader cryptocurrency market. The analysis encompasses individual price trends, distribution and outlier analysis.

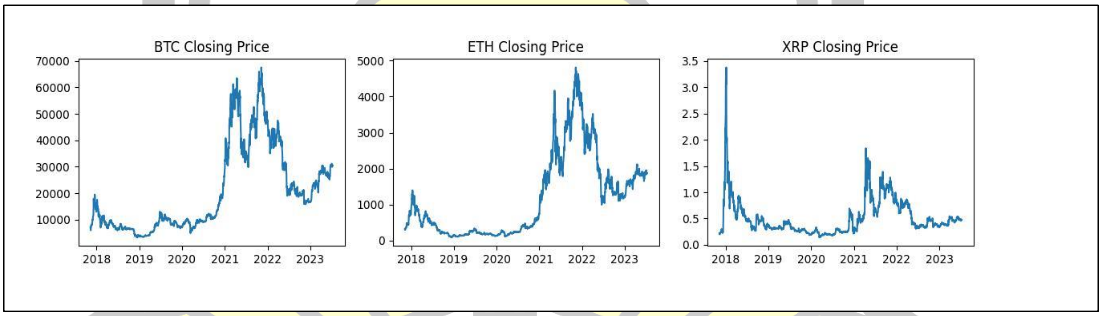


Figure 3.4 Data analysis: individual price trends of BTC, ETH, and XRP

Figure 3.4 shows the price trends of BTC, ETH and XRP from 2017 to 2023. BTC's steep rise to $67,566 in 2021 set the market's tone, followed by ETH's peak at $4,800. XRP, while showing some correlation, experienced higher volatility, with a notable peak in 2018 at $3.30. An analysis of individual price trends indicates that BTC plays a leading role in influencing the market, while ETH serves as a secondary driver, exhibiting some correlation with BTC but maintaining its unique characteristics. XRP, on the other hand, demonstrates distinctive market behavior with heightened volatility. Despite the interconnected nature of these cryptocurrencies, the analysis reveals that BTC, ETH, and XRP retain relative independence within the market.

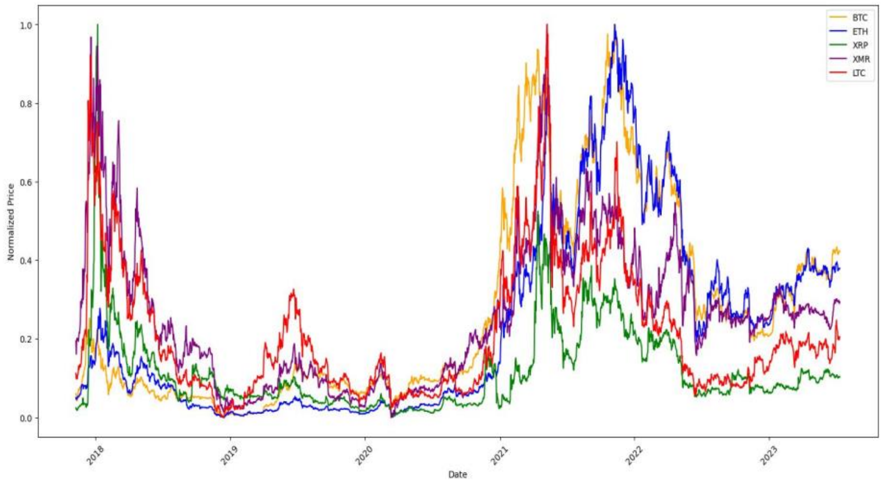


Figure 3.5 Comparative analysis of merged price trends (min-max scaler)

Figure 3.5 demonstrate how BTC and ETH move in close alignment, with ETH following BTC's lead. XRP shows greater volatility and some divergence, especially during its 2018 spike. XMR and LTC are more stable, generally following the overall market trend set by BTC and ETH, but with smaller price fluctuations. The trend analysis of the five normalized cryptocurrencies provides a clearer perspective on the relative price movements of BTC, ETH, and XRP and their influence on the broader cryptocurrency market.

Figure 3.6 Distribution and outlier analysis

Figure 3.6 reveals the distribution and outliers in the price data. BTC and ETH display significant price spikes, especially in 2021, impacting the entire market. XRP shows more dispersed distribution and frequent outliers, particularly in 2018. The distribution analysis confirms that BTC and ETH's significant price movements create market-wide effects, while XRP's frequent outliers reflect its higher volatility.

From above comprehensive analysis, it is clear that BTC's price fluctuations play a critical leading role in the entire cryptocurrency market. ETH and XRP act as key followers and volatility amplifiers, further driving overall market changes. When constructing high-dimensional models, it is essential to thoroughly consider the dominant roles of BTC and ETH in the market, while also addressing XRP's high volatility and frequent outliers, which exert a dual influence on the broader market. Therefore, during the data processing phase, it is necessary to integrate the distinct characteristics of the three cryptocurrencies while carefully mitigating the potential impact of outliers on model stability, in order to better capture the relationships between features and enhance the model's robustness.

3.2 Framework Overview

This section provides a detailed explanation of the procedures involved in constructing unified models for forecasting the closing price of cryptocurrency. Figure 3.7 depicts an overview of the framework.

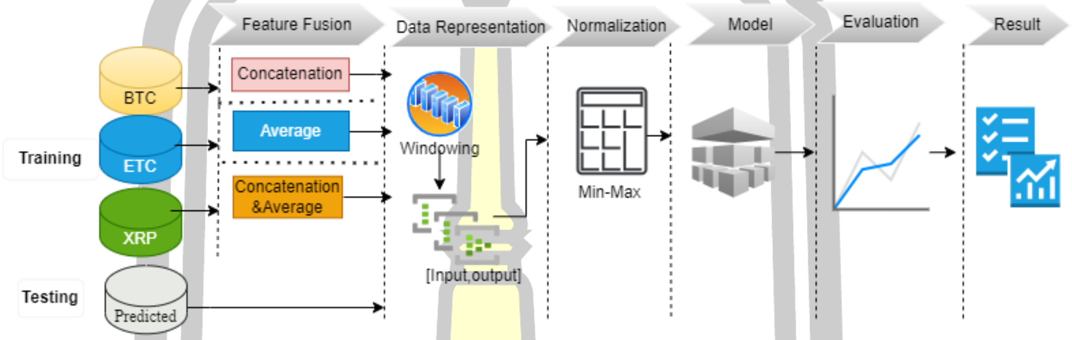


Figure 3.7 The framework overview of unified model

As depicted in the framework diagram, the training dataset comprises only the three domain cryptocurrencies: BTC, ETH, and XRP, while the testing dataset consists of the target cryptocurrency to be predicted. To maintain a balanced distribution for both training and evaluation processes, each dataset is typically split in an 80:20 ratio, with 80% allocated to the training set and the remaining 20% to the testing set. Generally, the dataset is randomly shuffled during splitting; however, for tasks with strong temporal dependencies, such as cryptocurrency price prediction, the data is not shuffled. To ensure the reproducibility of the results, a random seed is also set.

First, feature fusion techniques are utilized to integrate the attributes of BTC, ETH, and XRP, resulting in a unified and comprehensive training dataset. Three fusion strategies are implemented for comparison: Concatenation, Average, and Concat&Ave. Next, the training dataset is segmented into windows in vectorized form, with 5-day and 7-day windows capturing short-term temporal dependencies, and 24-day and 30-day windows representing long-term temporal dependencies, as required by the research objectives. The data is further organized into input-output pairs, with the past days' price within each window serving as inputs and the present-day price acting as the output. To ensure consistency across data scales, Min-Max normalization is applied within each window. The training data is then fed into predictive algorithms to construct and optimize the unified model.

During the testing phase, five cryptocurrencies are selected to comprehensively evaluate the performance of the unified model. These include the three domain currencies (BTC, ETH and XRP), as well as LTC, a derivative currency, and XMR, an independent cryptocurrency. The details of each processing step are outlined as follows.

3.3 Data Preprocessing

The preprocessing steps are designed based on continuous time-series data and include, feature fusion, data representation and normalization.

3.3.1 Feature-based Data Fusion

Data fusion is a methodology employed to integrate information from diverse data sources by emphasizing the characteristics or properties of the data. This methodology enables the extraction of pertinent characteristics from each data source, which can then be amalgamated to provide a dataset that is both more useful and complete. The prediction model should consider the comprehensiveness of the metrics, particularly the necessity for metric fusion. This study is characterized by the use of quantitative data, and the five cryptocurrencies have the same structure and units for their properties. Thus, the three methods suggested for feature fusion are as follows.

1) Concatenation (Concat)

Concatenation is a method of merging data where features or attributes from different data sources are combined into a single feature vector by simply stacking them together. This approach is straightforward and commonly employed in many applications, especially when the characteristics from several sources have the same or compatible data format. The process of concatenation is illustrated in Figure 3.8, taking a one-day window period as an example.

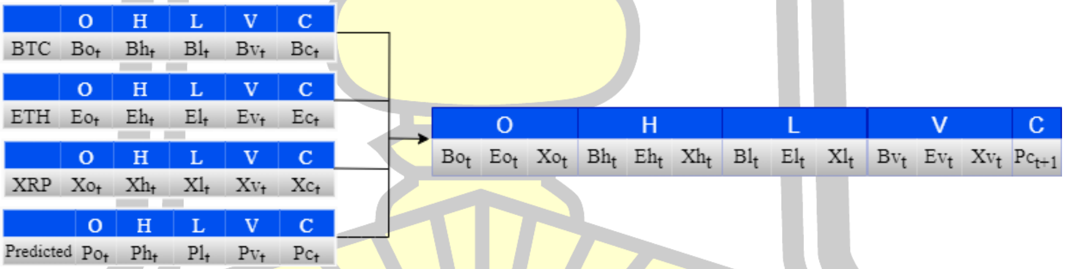


Figure 3.8 Concatenation of BTC, ETH, XRP datasets

We own comprehensive historical datasets encompassing all cryptocurrencies. This data can be utilized to predict the closing price of the following day. Let B represent Bitcoin (BTC), E represent Ethereum (ETH), and X represent Ripple (XRP). t represents the current time step, while t+1 represents the next time step. The current opening price of BTC is denoted as Bo_t, the current high price of BTC is denoted as Bh_t, the current low price of BTC is denoted as Bl_t, the current closing price of BTC is denoted as Bc_t,and the current volume of BTC is denoted as Bv_t. Similarly, using the identical notation to symbolize both ETH and XRP.

$$B_t = [Bo_t, Bh_t, Bl_t, Bv_t]$$

$$E_t = [Eo_t, Eh_t, El_t, Ev_t]$$

$$X_t = [Xo_t, Xh_t, Xl_t, Xv_t]$$

The features of training dataset are represented as x_train, the label of the training dataset is represented as y_train. Pc_{t+1} denotes the next time step closing price of the coin that is being forecasted. Similarly, the features and labels of the test dataset are denoted as x_test and y_test, respectively. The output is indicated as O. $\tilde{P}c_{t+1}$ denotes the forecasted closing price of the coin to be predicted. Mathematically x_train,y_train, x_test, y_test and O may be expressed as:

$$x\_train = [Bo_t, Bh_t, Bl_t, Bv_t, Eo_t, Eh_t, El_t, Ev_t, Xo_t, Xh_t, Xl_t, Xv_t]$$

$$y\_train = [Pc_{t+1}]$$

$$x\_test = [Po_t, Ph_t, Pl_t, Pv_t, Po_t, Ph_t, Pl_t, Pv_t, Po_t, Ph_t, Pl_t, Pv_t,]$$

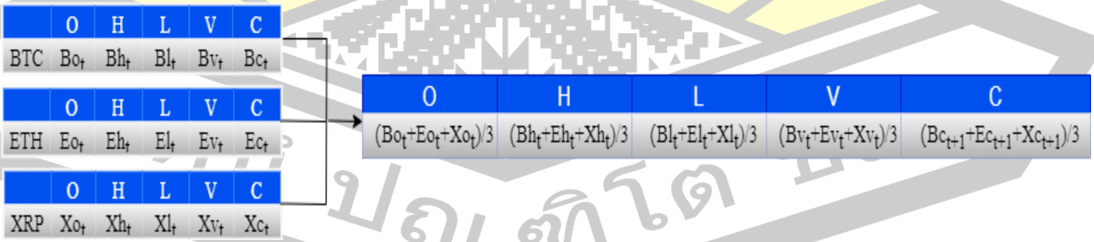
$$y\_test = [Pc_{t+1}]$$

$$O = [\tilde{P}c_{t+1}]$$

The objective is to forecast the output by utilizing the unified feature (x, y), which encompasses the characteristics of the previous values of BTC, ETH, and XRP. By combining the OHLV of the three cryptocurrencies, the total number of input features climbs to twelve. This can enhance the diversity of the feature space, enabling the model to acquire a more comprehensive understanding of market dynamics. Since the predicted cryptocurrency includes only four features (open, low, high, and volume), we expand these to twelve to match the feature set used during model training. Each test set is adjusted to include the same twelve variables, ensuring consistency with the training data.

2) Average (Ave)

Feature-level averaging, also known as feature-level mean fusion, is a data fusion approach that calculates the mean or average value of associated characteristics from several data sources. This approach creates a merged feature representation by gathering the core tendency of each feature and combining information from several sources. Feature-level averaging is straightforward to perform when the characteristics from different sources are numerical and reflect similar elements of the researched phenomena. Taking a one-day window period as an example, Figure 3.9 illustrates the process of feature level averaging.



	O	H	L	V	C
BTC	Bo_t	Bh_t	Bl_t	Bv_t	Bc_t

	O	H	L	V	C
ETH	Eo_t	Eh_t	El_t	Ev_t	Ec_t

	O	H	L	V	C
XRP	Xo_t	Xh_t	Xl_t	Xv_t	Xc_t

O	H	L	V	C
$(Bo_t+Eo_t+Xo_t)/3$	$(Bh_t+Eh_t+Xh_t)/3$	$(Bl_t+El_t+Xl_t)/3$	$(Bv_t+Ev_t+Xv_t)/3$	$(Bc_{t+1}+Ec_{t+1}+Xc_{t+1})/3$

Figure 3.9 Averaging of BTC, ETH, XRP datasets

The average values of OHLVC are computed at specific time intervals for BTC, ETH, and XRP. The average values of the close price are computed at next step time intervals for BTC, ETH, and XRP. These values are then employed as input vector (x, y) in the training stage. In the testing set, predictions are made using the features of the coin to be predicted and its associated label values as the input vector.

And the resulting value is O. x_train, y_train, x_test, y_test and O can be mathematically represented as follows:

$$\begin{aligned} x\_train &= [(Bo_t + Eo_t + Xo_t)/3, (Bh_t + Eh_t + Xh_t)/3, (Bl_t + El_t + Xl_t)/3, \\ &\quad (Bv_t + Ev_t + Xv_t)/3] \end{aligned}$$

$$y\_train = [(Bc_{t+1} + Ec_{t+1} + Xc_{t+1})/3]$$

$$x\_test = [Po_t, Ph_t, Pl_t, Pv_t]$$

$$y\_test = [Pc_{t+1}]$$

$$O = [\tilde{P}c_{t+1}]$$

It is noteworthy to mention that the averaging fusion algorithm requires only four input features to predict the closing price of a cryptocurrency. By employing this averaging fusion, the volatility in cryptocurrency features is effectively mitigated, thus augmenting the stability of subsequent model training. Additionally, this methodology aids in mitigating the influence of data disturbance, thereby facilitating the model's ability to discern latent trends and patterns.

3) Concatenation&Average (Concat&Ave)

Based on the two feature fusion approaches discussed above, it may proceed to a concatenation-average feature fusion method. In this method, the input features are created by concatenating the four features of the three distinct cryptocurrencies, yielding a total of twelve input features. The goal variable is calculated as the average of the closing prices of three dominant cryptocurrencies. Taking a one-day window period as an example, Figure 3.10 depicts the process of concatenation and average.

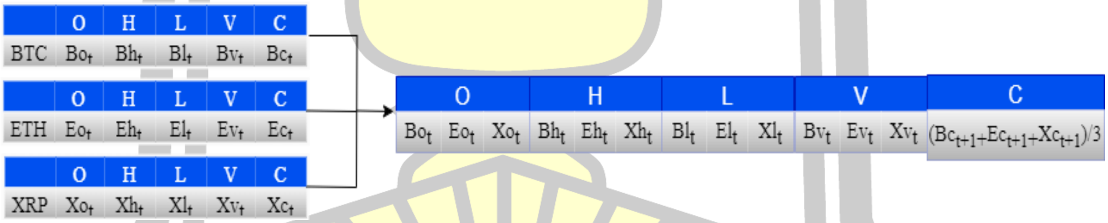


Figure 3.10 Concat&Ave of BTC, ETH, XRP datasets

The predicted cryptocurrency is denoted as P, while the remaining notation remains unchanged from the previous example. x_train, y_train, x_test, y_test can be mathematically represented as follows:

$$x\_train = [Bo_t, Bh_t, Bl_t, Bv_t, Xo_t, Xh_t, Xl_t, Xv_t, Eo_t, Eh_t, El_t, Ev_t]$$

$$y\_train = [(Bc_{t+1} + Ec_{t+1} + Xc_{t+1})/3]$$

$$x\_test = [Po_t, Ph_t, Pl_t, Pv_t, Po_t, Ph_t, Pl_t, Pv_t, Po_t, Ph_t, Pl_t, Pv_t,]$$

$$y\_test = [Pc_{t+1}]$$

We are experimenting with the Concat&Ave feature fusion method to develop a methodology that can successfully merge features from several cryptocurrencies, resulting in a more complete and accurate understanding. Furthermore, this strategy seeks to decrease data volatility by averaging. In principle, this technique has the ability to significantly improve cryptocurrency price prediction. However, it is crucial to remember that in actual applications, data redundancy may cause complications that result in lower model training performance. Therefore, careful management and optimization are essential.

3.3.2 Data Representation

After feature fusion, the data is segmented into vectorized sliding time windows along the time dimension. The data at each time point is represented as a vector, with each element corresponding to a feature.

To fully consider the impact of different time windows on prediction outcomes, this study selects 5-day, 7-day, 24-day, and 30-day intervals as prediction periods. The 5-day and 7-day windows represent short-term prediction periods, as these intervals effectively capture short-term market fluctuations and reflect traders' behavior patterns within a weekly cycle, commonly observed in financial markets. In contrast, the 24-day and 30-day windows are chosen as longer-term prediction periods to capture broader market trends. The 24-day window closely aligns with the number of trading days in a typical month, while the 30-day period extends this to encompass a full monthly cycle, allowing for the identification of more extensive market movements and trends.

The time windows are organized into input-output pairs to facilitate supervised learning. Specifically, the feature data from the previous days within each window is used as the input, while the target variable (e.g., closing price) on the day following the window is treated as the output. For example, in a 5-day window, the feature data from the previous 5 days serves as the input, and the target variable on the 6th day is the output. This structure transforms the sequential time-series data into a format suitable for supervised learning tasks, while preserving temporal dependencies for effective model training. Figure 3.11 provides a visual representation of 5 and 7-day examples of predicting periods on a daily basis.

(a) A period of 5 days for the prediction

(b) A period of 7 days for the prediction

■ Train

□ Predicted

Figure 3.11 An illustration of utilizing 5- and 7-day periods for prediction

3.3.3 Data Normalization

This study employed data normalization techniques to standardize heterogeneous data into a narrower range, typically between 0.0 and 1.0 or -1.0 and

10. Data normalization eliminates dimensional discrepancies between features, enhancing the stability of the model training process and improving the model's generalization capability. Normalization also simplifies data classification and understanding, thereby increasing the accuracy and efficiency of data mining algorithms applied to standardized datasets.

In this study, the Min-max normalization technique was used to process cryptocurrency data. The advantages of Min-max normalization include its simplicity and intuitive nature, as it preserves the relative proportions of features and is particularly suitable for data within bounded ranges. However, it is sensitive to outliers, which can potentially distort the normalization results.

Furthermore, normalization was applied within each time window to ensure that the features within each window are on a consistent scale. This approach offers several advantages: it reduces the impact of time-dependent fluctuations and ensures that each window's data is treated independently, which helps preserve temporal dependencies and improves model performance. For the cryptocurrency price prediction task, this window-level normalization is particularly important.

3.4 Models and Setting Parameters for Predictor

This study compared between machine learning-based and deep learning-based methods. Detailed descriptions of setting algorithms for training predictor models are provided below.

3.4.1 Modeling of Cryptocurrency Closing-price Predictor

When developing a model to predict the closing-price of a cryptocurrency using machine learning methods, two specific algorithms, namely Support Vector Regression (SVR) and Random Forest (RF), are utilized. When developing the cryptocurrency closing-price prediction model utilizing deep learning algorithms, two specific algorithms, namely LSTM and GRU, are utilized. Algorithm pseudocodes can be represented by Table 3.2, Table 3.3, Table 3.4 and Table 3.5.

Table 3.2 The modeling process for the SVR model

Algorithm	SVR
Input:	Data point D = (x_1,y_1) , (x_2,y_2) ,… (x_i,y_i)
Output:	The predicted values f(x)
1:	Read the Data
2:	Preprocess the Data
3:	Split the Data into training and testing sets: [x_train, y_train] [x_test, y_test]
4:	Create a SVR model: from sklearn.svm import SVR Svr = SVR (kernel, c, gamma) Find the best hyperparameters using Grid Search
5:	Fit the model: Svr.fit (x_train, y_train)
6:	Make predictions using the test set: f(x) = Svr.predict(x_test)

Table 3.3 The modeling process for the RF model

Algorithm	RF
Input:	Data point D = (x_1,y_1) , (x_2,y_2) ,… (x_i,y_i)
Output:	The predicted values f(x)
1:	Read the Data
2:	Preprocess the Data
3:	Split the Data into training and testing sets: [x_trian, y_train] [x_test, y_test]
4:	Create a RF model: from sklearn. ensemble import RF rf = RF (n_estimators, random_state)
5:	**Fit the model:** rf.fit (x_train, y_train)
6:	Make predictions using the test set: f (x) =rf. predict (x_test)

Table 3.4 The modeling process for the LSTM model

Algorithm	LSTM
Input:	Data point D = （x_1,y_1）,（x_2,y_2）,...（x_i,y_i）
Output:	The predicted values f(x)
1:	Read the Data
2:	Preprocess the Data
3:	Split the Data into training and testing sets: [x_trian, y_train] [x_test, y_test]
4:	Create a LSTM model: lstm = Sequential () lstm.add (LSTM (neurons, input_shape)) lstm.add (Dropout (dropout=0.2)) lstm.add (Dense (units=1))
5:	Compile the model lstm. compile (loss, optimizer)
6:	Fit the model: lstm.fit (x_train, y_train, epochs, batch_size)
7:	Make predictions using the test set: f (x) =lstm. predict (x_test)

Table 3.5 The modeling process for the GRU model

Algorith:	GRU
Input:	Data point D = （x_1,y_1）,（x_2,y_2）,...（x_i,y_i）
Output:	The predicted values f(x)
1:	Read the Data
2:	Preprocess the Data
3:	Split the Data into training and testing sets: [x_trian, y_train] [x_test, y_test]
4:	Create a GRU model: gru = Sequential () gru.add (LSTM (neurons, input_shape)) gru.add (Dropout (dropout)) gru.add (Dense (units=1))
5:	Compile the model gru. compile (loss, optimizer)
6:	Fit the model: gru.fit (x_train, y_train, epochs, batch_size)
7:	Make predictions using the test set: f (x) =gru. predict (x_test)

3.4.2 Setting Parameters for Predictors

To develop predictive models for cryptocurrency closing prices, this research investigates Support Vector Regression (SVR) and Random Forest (RF) algorithms in the realm of machine learning (ML), as well as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms for deep learning (DL). Parameter tuning was conducted through rigorous experimentation to identify optimal configurations. Detailed modeling and parameter settings are as follows:

SVR is employed with a Radial Basis Function (RBF) kernel, selected for its robust nonlinear mapping capabilities crucial for handling complex time series data. The parameter C serves as a critical regularization parameter, balancing the trade-off between model complexity and tolerance for misclassification. A smaller C increases the tolerance for misclassification, which may result in underfitting, while a larger C enhances model complexity, potentially leading to overfitting. In this configuration, C is set to 1000, a value considered appropriate based on existing literature [45]. A slower learning rate takes much time and has more probability of converging or being stuck in an undesirable local minimum. At the same time, a higher one makes the learning jump over minima. Another significant parameter is $gamma$. An excessively high $gamma$ value can make the model overly complex and prone to overfitting, while a low $gamma$ value may prevent the model from capturing intricate data patterns. In this study, the specific numbers for the SVR hyperparameters of epsilon and gamma were initially set at 0.1 and 0.1, respectively. However, through extensive experimentation and model validation, the most optimal values for these parameters varied across different fusion strategies. For the SVR+Concatenation-based model, the optimal hyperparameters were found to be [C = 1000, epsilon = 0.01, gamma = 0.6], which provided a good balance between fitting the training data and generalizing to unseen data. In the case of the SVR+Average-based model, the best values were [C = 1000, epsilon = 0.8, gamma = 2], which allowed the model to focus on capturing more complex patterns while accommodating larger deviations. Finally, for the SVR+Concatenation&Average-based model, the optimal parameters were [C = 1000, epsilon = 0.1, gamma = 0.01], ensuring a smoother fit that minimized overfitting while maintaining model performance.

RF model is also utilized, with key parameters including $n\_estimators$ and $max\_depthmax$. The $n\_estimators$ parameter controls the number of trees in the forest, where a higher value may lead to overfitting. Initially, the max_depth was set to 3, and n_estimators was started with a small value of 100. Through extensive experimentation, the optimal parameter settings for different fusion strategies were identified. For the RF+Concatenation-based model, the best parameters were [max_depth = 3, n_estimators = 500], which provided a balanced trade-off between model accuracy and computational efficiency. For the RF+Average-based model, the optimal settings were [max_depth = 3, n_estimators = 200], while for the RF+Concatenation&Average-based model, the best parameters were [max_depth = 3, n_estimators = 800]. Other parameters were retained at their default values.

In this research, deep learning models such as LSTM and GRU are employed with dropout layers to prevent overfitting. The models use a four-layer architecture: an input layer, a fully connected layer with 100 neurons, a dropout layer, and an output layer. The Adam optimizer is used for mini-batch gradient descent, with a batch_size range from 32 to balance training speed and generalization. The ReLU activation function is applied to the fully connected layer to capture nonlinear relationships, and a dropout rate of 0.2 is set to reduce overfitting. The models are trained from 5 epochs, ensuring optimal performance while avoiding overfitting. Through experimentation, the best parameters for each model were identified:

- LSTM+Concatenation-based and GRU+Concatenation-based models: [batch_size = 64, epochs = 20]
- LSTM+Average-based and GRU+Average-based models: [batch_size = 32, epochs = 5]
- LSTM+Concatenation&Average-based and GRU+Concatenation&Average-based models: [batch_size = 32, epochs = 20]

The main parameter settings for all the models discussed above are provided in Table 3.6.

Table 3.6 Main Parameters Setting

Model	Parameter	Concat	Ave	Concat&Ave
SVR	*C*	1000	1000	1000
	gamma	0.6	2	0.1
	epsilon	0.01	0.8	0.01
RF	*max_depth*	3	3	3
	n_estimators	500	200	800
LSTM	*batch_size*	64	32	32
	epochs	20	5	20
GRU	*batch_size*	64	32	32
	epochs	20	5	20

3.5 Prediction Model Evaluation

This section provides a concise overview of the assessment measures employed to evaluate the performance of the proposed model. The study utilizes Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). These metrics are well-established and frequently used to assess the accuracy of continuous variables and evaluate prediction models.

Among these, MAPE is particularly useful as it represents errors as percentages, allowing for easier comparison between models. MAE and RMSE express average model prediction errors in units corresponding to the variable under study. These metrics range from 0 to infinity and do not account for the direction of errors. Lower values are preferable, as they indicate better model performance. These metrics can be used together to understand the variability in discrepancies or inaccuracies within a set of predictions.

CHAPTER 4 RESULTS AND DISCUSSION

This chapter presents the empirical result focusing on development of three unified models for predicting the closing price of various cryptocurrencies. All models discussed are built using Python. Employing the Keras deep learning framework with TensorFlow as the backend, these models are executed in Jupyter notebook to generate predictive outcomes. Additionally, a comparative analysis of each model's predictive performance is conducted based on three evaluation metrics.

4.1 Experimental Design

We will train and evaluate the model using four different window lengths. The size of the training and test datasets varies depending on the chosen window length:

5-day window: 1,652 training points, 409 test points.

7-day window: 1,650 training points, 407 test points.

24-day window: 1,633 training points, 390 test points.

30-day window: 1,627 training points, 384 test points.

Testing different window lengths is essential for analyzing the model's ability to capture short-term and long-term trends in cryptocurrency price movements.

- Shorter windows (5-day and 7-day): These windows allow the model to focus on short-term price fluctuations, which are particularly useful for identifying immediate market trends and volatility.

- Longer windows (24-day and 30-day): These windows provide a broader historical context, enabling the model to learn more stable and long-term patterns in price movements, which can enhance predictive robustness for long-term investors.

4.2 Data Normalization Technique: Min-Max

This study used only min-max normalization without comparing it to other normalization techniques is reasonable and well-justified, given the primary focus of this study on developing a predictive model that integrates Data Fusion techniques (i.e., Concatenation, Averaging, and Concatenation-Averaging). The following reasons support the adequacy and appropriateness of this approach:

- Alignment with Research Objectives - The core objective of this study is to enhance predictive performance through Data Fusion techniques rather than conducting a comparative analysis of different data normalization methods. Since normalization is a preprocessing step rather than a key methodological focus, keeping it consistent ensures that the study remains centered on evaluating fusion strategies.

- Effectiveness and Widespread Use of Min-Max Normalization - Min-max normalization is one of the most widely used techniques in financial time series prediction because it scales data within a fixed range (e.g., [0,1] or [-1,1]), preserving the relative relationships between values. Given that cryptocurrency price data often exhibits significant fluctuations, this method helps maintain the interpretability of price variations without distorting the data distribution.

- Consistency in Feature Transformation - This study involves multiple feature fusion strategies. Introducing multiple normalization techniques could introduce inconsistencies in feature representations, potentially affecting the comparability of fusion results. By standardizing all features using a single, well-established normalization technique, you ensure that performance differences across fusion techniques are attributed solely to the fusion methods themselves, rather than variations in data preprocessing.

- Computational Efficiency and Stability - Unlike other normalization techniques (e.g., z-score normalization, robust scaling), min-max normalization is computationally efficient and does not introduce unnecessary complexity. This efficiency is crucial in deep learning models, where excessive computational overhead from alternative normalization methods might not yield substantial improvements in predictive performance.

- Empirical Justification from Prior Studies - Many previous studies on cryptocurrency price prediction and financial time series modeling have successfully utilized min-max normalization as a default standardization approach due to its ability to handle highly dynamic price movements effectively. Since this study emphasizes fusion techniques rather than preprocessing variations, deviating from this well-established practice would add unnecessary complexity without a direct contribution to the research question.

- Maintaining the Integrity of Comparative Analysis - Since this study compares different Data Fusion techniques, it is critical to isolate the effect of fusion strategies rather than introducing additional variables, such as differing normalization methods. A single, uniform preprocessing approach ensures that the observed model performance variations are solely attributed to the differences in Data Fusion techniques, rather than inconsistencies in feature scaling.

4.3 Input Data Test after Performing Data Fusion Techniques

This section presents examples of input data used for model testing after applying each data fusion technique.

4.3.1 Input Test Data after Applying Concatenation-Based Data Fusion Techniques

To test the proposed unified predictive model using concatenation-based data fusion techniques, each test set for the predicted currencies initially contains only four variables: open, low, high, and volume. To ensure consistency with the 12 variables used during training, the test set must be expanded from original 4 to 12 variables. Both training and testing labels are based on the next day's closing price of the predicted currency. The pre-processed data, after applying the concatenation-based data fusion technique, is shown in Table 4.1.

Table 4.1 Pre-processed data after performing Concatenation-based data fusion

Train Data	**Test Data（Original）**	**Test Data（Expanding）**
BTC, ETH, XRP	Predicted	
Variable：[Bo_t, Bh_t, Bl_t, Bv_t, Eo_t, Eh_t, El_t, Ev_t, Xo_t, Xh_t, Xl_t, Xv_t] Lable：[Pc_(t+1)]	Variable：[Po_t, Ph_t, Pl_t, Pv_t] Lable：[Pc_(t+1)]	Variable：[Po_t, Ph_t, Pl_t, Pv_t, Po_t, Ph_t, Pl_t, Pv_t, Po_t, Ph_t, Pl_t, Pv_t] Lable：[Pc_(t+1)]
12-Dimensional Features	4-Dimensional Features	12-Dimensional Features

4.3.2 Input Test Data after Applying Average -Based Data Fusion Techniques

Compared to the concatenation method, the average-based data fusion method generates a 4-dimensional feature vector by averaging the features from multiple cryptocurrencies. In the training data, the relevant features of BTC, ETH, and XRP are averaged, such as (Bo_t + Eo_t + Xo_t)/3, to consolidate the information from multiple cryptocurrencies into a smoother feature vector. The test set directly uses the four features of the predicted currency and the next day's closing price as the label. The pre-processed data, after applying the average-based data fusion technique, is shown in Table 4.2.

Table 4.2 Pre-processed data after performing Average-based data fusion

Train Data	**Test Data**
BTC, ETH, XRP	Predicted
Variable：[(Bo_t+Eo_t+Xo_t)/3, (Bh_t+Eh_t+Xh_t)/3, (Bl_t+El_t+Xl_t)/3, (Bv_t+Ev_t+Xv_t)/3] Lable：[(Bc_(t+1) +Ec_(t+1) +Xc_(t+1))/3]	Variable：[Po_t, Ph_t, Pl_t, Pv_t] Lable：[Pc_(t+1)]
4-Dimensional Features	4-Dimensional Features

4.3.3 Input Test Data after Applying Concatenation&Average-based Data Fusion Techniques

By combining the two methods, the training set for Concatenation & Average-Based Data Fusion Techniques has 12 dimensions for the variables, with the label being the average closing price of the three main cryptocurrencies. During testing, the variables of the predicted currency also need to be expanded to 12 dimensions. The pre-processed data, after applying the concatenation & average-based data fusion technique, is shown in Table 4.3.

Table 4.3 Pre-processed data after performing Concatenation&Average-based data fusion

Train Data	**Test Data（Original）**	**Test Data（Transforming）**
BTC, ETH, XRP	Predicted	
Variable：[Bo_t, Bh_t, Bl_t, Bv_t, Eo_t, Eh_t, El_t, Ev_t, Xo_t, Xh_t, Xl_t, Xv_t] Lable：[(Bc_(t+1) +Ec_(t+1) +Xc_(t+1))/3]	Variable：[Po_t, Ph_t, Pl_t, Pv_t] Lable：[Pc_(t+1)]	Variable：[Po_t, Ph_t, Pl_t, Pv_t, Po_t, Ph_t, Pl_t, Pv_t, Po_t, Ph_t, Pl_t, Pv_t] Lable：[Pc_(t+1)]
12-Dimensional Features	4-Dimensional Features	12-Dimensional Features

4.4 Results of the Proposed Predictive Models

This section presents the results of each model evaluated using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). These metrics provide a comprehensive assessment of model performance by capturing different aspects of prediction accuracy.

MAE measures the average magnitude of errors in absolute terms, making it useful for understanding overall prediction deviation without considering error direction. RMSE penalizes larger errors more heavily due to squaring, making it particularly effective in highlighting significant prediction deviations, which are critical in financial forecasting where extreme fluctuations can impact decision-making. MAPE, expressed as a percentage, normalizes the error relative to actual values, allowing for a scale-independent comparison of model accuracy across different price ranges.

By incorporating these three metrics, the evaluation provides a balanced perspective on model performance, ensuring that both general accuracy and sensitivity to large deviations are considered, while also allowing for practical interpretation of errors in real-world trading and investment scenarios.

The test results for each model are presented as follows.

4.4.1 Results of 5-day Period Prediction

The results of predictive models for 5-day period prediction can be presented as Table 4.4.

Table 4.4 The results for 5-day prediction

Crypto	Algorithms	Data Fusion Technique	MAE	RMSE	MAPE
BTC	SVR	Concatenation	530.6048	766.0708	0.0233
		Average	568.6229	824.8025	0.0249
		Concatenation & Average	682.3836	963.5251	0.0295
	RF	Concatenation	461.1602	698.7625	0.0202
		Average	464.2645	697.6566	0.0204
		Concatenation & Average	699.7733	1037.5772	0.0307
	GRU	Concatenation	485.6288	697.8597	0.0213
		Average	466.8412	696.4325	0.0204
		Concatenation & Average	458.8724	685.9159	0.0201
	LSTM	Concatenation	464.1796	692.0714	0.0209
		Average	**458.6837**	**690.9105**	**0.02**
		Concatenation & Average	499.7478	720.0335	0.0218
ETH	SVR	Concatenation	48.7428	68.7142	0.0323
		Average	50.6212	72.0662	0.0334
		Concatenation & Average	61.6994	85.2395	0.04
	RF	Concatenation	42.5107	61.7401	0.0279
		Average	42.0182	61.0124	0.0279
		Concatenation & Average	60.4627	85.393	0.0404
	GRU	Concatenation	41.8599	59.4176	0.0276
		Average	41.3083	59.3172	0.0272
		Concatenation & Average	42.5236	61.4188	0.0281
	LSTM	Concatenation	42.1937	59.2749	0.02790
		Average	**41.0975**	**59.0418**	**0.0271**
		Concatenation & Average	43.0535	63.1536	0.0285
XRP	SVR	Concatenation	0.0116	0.0173	0.0281
		Average	0.0119	0.0178	0.0291
		Concatenation & Average	0.0175	0.0277	0.0419
	RF	Concatenation	0.0108	0.0157	0.0263
		Average	0.0112	0.0162	0.0271
		Concatenation & Average	0.0135	0.0199	0.033
	GRU	Concatenation	0.0115	0.0169	0.0279
		Average	0.0107	0.0157	0.026
		Concatenation & Average	0.0136	0.0225	0.0323
	LSTM	Concatenation	0.0111	0.0161	0.027
		Average	**0.0105**	**0.0154**	**0.0257**
		Concatenation & Average	0.0133	0.0209	0.0319

Table 4.4 (Cont')

Crypto	Algorithms	Data Fusion Technique	MAE	RMSE	MAPE
XMR	SVR	Concatenation	3.688	5.0453	0.0248
		Average	3.8708	5.656	0.0263
		Concatenation & Average	4.8996	6.8135	0.0326
	RF	Concatenation	3.2529	4.6786	0.0221
		Average	3.5005	5.1412	0.0236
		Concatenation & Average	4.4201	6.7139	0.0303
	GRU	Concatenation	3.4084	4.9157	0.0231
		Average	3.3609	4.897	0.0227
		Concatenation & Average	3.4073	5.0156	0.023
	LSTM	Concatenation	3.3893	4.9334	0.0229
		Average	**3.1581**	**4.6506**	**0.0222**
		Concatenation & Average	3.7319	5.2966	0.025
LTC	SVR	Concatenation	2.6739	3.7483	0.0368
		Average	2.6514	3.8228	0.0365
		Concatenation & Average	3.6028	5.3563	0.0489
	RF	Concatenation	2.1955	3.1742	0.0305
		Average	2.3249	3.3525	0.0325
		Concatenation & Average	3.002	4.1614	0.0413
	GRU	Concatenation	2.2263	3.1472	0.031
		Average	2.2218	3.1633	0.03076
		Concatenation & Average	2.8083	3.9511	0.0384
	LSTM	Concatenation	2.2012	3.1379	0.0305
		Average	**2.1813**	**3.1166**	**0.0302**
		Concatenation & Average	2.5489	3.6261	0.035

In the 5-day window prediction, the LSTM + average-based feature fusion method showed the best performance in terms of MAE, RMSE, and MAPE. Compared to SVR and RF, both GRU and LSTM are more suitable for handling time series data and effectively capturing cryptocurrency price fluctuations. While LSTM has an advantage over GRU in capturing long-term dependencies, its complex structure leads to higher computational costs and longer training times. By using the average fusion technique, noise is reduced, and the feature data is smoothed, which helps compensate for LSTM's efficiency drawbacks, thus enhancing its prediction performance and stability.

In summary, the LSTM + Average-based data fusion method significantly improves prediction accuracy and stability by combining LSTM's ability to process time series data with the smoothing effect of average data fusion. This method performed the best in short-term (5-day) prediction tasks for all cryptocurrencies, effectively handling market volatility, making it the most suitable predictive model.

4.4.2 Results of 7-day Period Prediction

The results of predictive models for 7-day period prediction can be presented as Table 4.5.

Table 4.5 The results for 7-day prediction

Crypto	Algorithms	Data Fusion Technique	MAE	RMSE	MAPE
BTC	SVR	Concatenation	600.9205	882.2418	0.0264
		Average	714.6294	1043.5299	0.0314
		Concatenation & Average	576.6435	791.0632	0.0253
	RF	Concatenation	454.6314	680.3421	0.0199
		Average	471.9496	715.1413	0.0208
		Concatenation & Average	802.074	1197.0317	0.0355
	GRU	Concatenation	453.7366	669.9489	0.0198
		Average	460.1362	683.9973	0.0202
		Concatenation & Average	453.7604	676.479	0.0199
	LSTM	Concatenation	451.3077	**673.3674**	0.0198
		Average	**447.0709**	677.2853	**0.0196**
		Concatenation & Average	468.8841	696.6986	0.0206
ETH	SVR	Concatenation	53.735	77.4009	0.036
		Average	61.9952	86.9766	0.0414
		Concatenation & Average	55.277	74.06	0.0363
	RF	Concatenation	42.4921	59.9478	0.028
		Average	43.5892	62.6818	0.0291
		Concatenation & Average	68.8802	97.166	0.0464
	GRU	Concatenation	43.1776	60.4969	0.02855
		Average	42.4719	59.9407	0.0281
		Concatenation & Average	41.9609	59.3846	0.0279
	LSTM	Concatenation	41.0845	58.8866	0.0273
		Average	**40.1942**	**58.7056**	**0.0268**
		Concatenation & Average	47.0934	65.8033	0.0311
XRP	SVR	Concatenation	0.0129	0.0196	0.0311
		Average	0.0137	0.0202	0.0333
		Concatenation & Average	0.0165	0.0257	0.0391
	RF	Concatenation	0.011	0.016	0.0269
		Average	0.0111	0.0161	0.0268
		Concatenation & Average	0.015	0.0218	0.0366
	GRU	Concatenation	0.0109	0.0162	0.0267
		Average	0.0105	0.0155	0.0256
		Concatenation & Average	0.0148	0.655	0.0349
	LSTM	Concatenation	0.0108	0.0159	0.0264
		Average	**0.0102**	**0.0153**	**0.025**
		Concatenation & Average	0.0138	0.0219	0.0329

Table 4.5 (Cont')

Crypto	Algorithms	Data Fusion Technique	MAE	RMSE	MAPE
XMR	SVR	Concatenation	4.0391	5.6256	0.0272
		Average	4.5506	6.7496	0.0313
		Concatenation & Average	4.6066	6.1424	0.0308
	RF	Concatenation	3.2529	4.6786	0.0221
		Average	3.5301	5.1612	0.024
		Concatenation & Average	4.9167	7.5776	0.0341
	GRU	Concatenation	3.2785	4.7603	0.0224
		Average	3.3068	4.7712	0.0225
		Concatenation & Average	3.358	4.7758	0.0228
	LSTM	Concatenation	3.3436	4.8792	0.0227
		Average	**3.275**	**4.7572**	**0.0223**
		Concatenation & Average	4.0036	5.4131	0.0268
LTC	SVR	Concatenation	2.9666	4.113	0.041
		Average	3.0881	4.196	0.0425
		Concatenation & Average	3.212	4.4734	0.044
	RF	Concatenation	2.2497	3.1728	0.0312
		Average	2.3527	3.3693	0.0328
		Concatenation & Average	3.3345	4.476	0.0461
	GRU	Concatenation	2.2229	3.1276	0.0303
		Average	2.1658	3.1258	0.03
		Concatenation & Average	2.1867	3.1378	0.0302
	LSTM	Concatenation	2.1936	**3.095**	0.0303
		Average	**2.1589**	3.1371	**0.0299**
		Concatenation & Average	3.026	4.2764	0.0413

In the 7-day window prediction results, the LSTM + average-based feature fusion method still showed the lowest errors in MAE, RMSE, and MAPE, demonstrating its strong predictive capability. However, in the predictions for BTC and LTC, the RMSE of the LSTM + Concatenation model was slightly lower than that of the LSTM + average-based feature fusion method. This phenomenon may be due to the characteristics of the concatenation method, which directly merges the features of different cryptocurrencies, preserving the unique information of each currency. This approach is particularly helpful in predicting markets with high volatility (such as BTC and LTC), as it better captures short-term fluctuations.

In conclusion, the LSTM + average-based feature fusion method performs best in most metrics, providing the most stable and accurate predictions, and the LSTM + Concatenation method also demonstrates advantages in certain cases, particularly when capturing market volatility.

4.4.3 Results of 24-day Period Prediction

The results of predictive models for 24-day period prediction can be presented as Table 4.6.

Table 4.6 The results for 24-day prediction

Crypto	Algorithms	Data Fusion Technique	MAE	RMSE	MAPE
BTC	SVR	Concatenation	1533.0117	1986.1527	0.0687
		Average	1541.6335	2004.6217	0.0693
		Concatenation & Average	655.8719	924.7917	0.0289
	RF	Concatenation	466.9816	690.3513	0.0211
		Average	581.3517	810.0493	0.0259
		Concatenation & Average	1539.4163	1998.6082	0.069
	GRU	Concatenation	452.2644	649.3874	0.02
		Average	**440.8092**	**637.4319**	**0.0197**
		Concatenation & Average	448.5807	643.4527	0.0201
	LSTM	Concatenation	455.3163	668.8547	0.0201
		Average	448.9357	674.029	0.0201
		Concatenation & Average	452.3023	661.1657	0.0201
ETH	SVR	Concatenation	128.2142	156.853	0.0872
		Average	128.7885	157.9273	0.0877
		Concatenation & Average	63.581	86.6771	0.0423
	RF	Concatenation	41.9219	58.355	0.0277
		Average	45.7287	65.352	0.0304
		Concatenation & Average	128.6242	157.4256	0.0874
	GRU	Concatenation	40.5968	58.166	0.0271
		Average	**39.8004**	**56.9062**	**0.0264**
		Concatenation & Average	49.287	60.2487	0.0327
	LSTM	Concatenation	42.734	60.5849	0.0285
		Average	45.3876	64.5981	0.0303
		Concatenation & Average	49.7209	70.3906	0.0328
XRP	SVR	Concatenation	0.0268	0.035	0.0629
		Average	0.0267	0.0334	0.0636
		Concatenation & Average	0.0227	0.0316	0.0524
	RF	Concatenation	0.0112	0.0162	0.0269
		Average	0.0123	0.018	0.0299
		Concatenation & Average	0.0267	0.0335	0.0635
	GRU	Concatenation	0.0114	0.0167	0.0272
		Average	**0.0105**	**0.0156**	**0.0255**
		Concatenation & Average	0.0254	0.4546	0.058
	LSTM	Concatenation	0.0111	0.0162	0.0267
		Average	0.0117	0.0169	0.0285
		Concatenation & Average	0.0269	0.0384	0.0615

Table 4.6 (Cont’)

Crypto	Algorithms	Data Fusion Technique	MAE	RMSE	MAPE
XMR	SVR	Concatenation	8.7565	11.6669	0.0607
		Average	8.8437	11.8903	0.0616
		Concatenation & Average	5.5724	7.6105	0.0377
	RF	Concatenation	3.0754	4.3558	0.0209
		Average	3.8289	5.1055	0.0261
		Concatenation & Average	8.8118	11.8021	0.0612
	GRU	Concatenation	**3.0208**	**4.3462**	**0.02059**
		Average	3.1155	4.4702	0.0212
		Concatenation & Average	3.3212	4.7152	0.0226
	LSTM	Concatenation	3.2269	4.6348	0.022
		Average	3.3874	4.8275	0.0232
		Concatenation & Average	5.5886	7.7667	0.0377
LTC	SVR	Concatenation	5.4226	6.7433	0.0712
		Average	5.2436	6.4784	0.0699
		Concatenation & Average	4.6062	6.1923	0.063
	RF	Concatenation	2.4648	4.1868	0.0338
		Average	2.7972	3.8994	0.0385
		Concatenation & Average	5.271	6.5174	0.07
	GRU	Concatenation	2.3991	3.3717	0.0327
		Average	**2.246**	**3.3153**	**0.0306**
		Concatenation & Average	2.2484	3.2531	0.0305
	LSTM	Concatenation	2.3366	3.3426	0.0319
		Average	2.4604	3.5866	0.0334
		Concatenation & Average	5.1253	7.0034	0.0684

In the 24-day window period prediction, the GRU + Average-based data fusion method performed well for BTC, ETH, XRP, and LTC. This is primarily due to GRU’s simplified structure and computational efficiency, allowing for more stable processing of long-window data while avoiding the overfitting issues that may arise with LSTM. Additionally, the Average-based method smooths the data and reduces the impact of short-term fluctuations, enabling the model to capture long-term trends more effectively. In contrast, XMR exhibits high volatility, low liquidity, strong privacy features, and high sensitivity to news, making the Concatenation method more effective in preserving critical features and adapting to sudden market changes. As a result, GRU + Concatenation performed better in predicting XMR.

Overall, in the 24-day period, the GRU + Average-based data fusion method stands out and is suitable for predicting most cryptocurrencies. Its efficiency and stable performance over long term period make it an ideal choice.

4.4.4 Results of 30-day Period Prediction

The results of predictive models for 30-day period prediction can be presented as Table 4.7.

Table 4.7 The results for 30-day prediction

Crypto	Algorithms	Data Fusion Technique	MAE	RMSE	MAPE
BTC	SVR	Concatenation	1676.9881	2107.4675	0.0744
		Average	1678.7766	2109.9289	0.0745
		Concatenation & Average	789.7404	1060.3095	0.0345
	RF	Concatenation	438.0184	625.6394	0.0195
		Average	643.1016	866.8952	0.0283
		Concatenation & Average	1678.9353	2110.0225	0.0745
	GRU	Concatenation	430.1802	**624.5816**	0.0191
		Average	**428.4483**	634.4250	**0.0191**
		Concatenation & Average	443.3407	638.4558	0.0196
	LSTM	Concatenation	444.631	650.8871	0.0196
		Average	468.3718	695.3241	0.0209
		Concatenation & Average	452.2236	648.7263	0.0202
ETH	SVR	Concatenation	137.5478	166.3264	0.0923
		Average	137.2165	166.0925	0.0918
		Concatenation & Average	70.5421	92.757	0.0468
	RF	Concatenation	40.5123	56.6872	0.0266
		Average	53.4566	71.738	0.0346
		Concatenation & Average	137.3837	166.2311	0.092
	GRU	Concatenation	42.8291	60.6097	0.0281
		Average	**39.8**	**57.6416**	**0.0262**
		Concatenation & Average	52.1109	62.3291	0.034
	LSTM	Concatenation	43.6858	61.957	0.0288
		Average	43.4867	62.7998	0.0288
		Concatenation & Average	52.6227	73.0835	0.0343
XRP	SVR	Concatenation	0.029	0.0375	0.0675
		Average	0.029	0.0356	0.0685
		Concatenation & Average	0.0239	0.0329	0.055
	RF	Concatenation	0.0109	0.016	0.0264
		Average	0.0128	0.0186	0.0306
		Concatenation & Average	0.029	0.0355	0.0685
	GRU	Concatenation	0.0113	0.0166	0.0268
		Average	**0.0107**	**0.0157**	**0.0258**
		Concatenation & Average	0.0314	0.4548	0.0707
	LSTM	Concatenation	0.011	0.0161	0.0265
		Average	0.0111	0.0162	0.0269
		Concatenation & Average	0.031	0.0437	0.0699

Table 4.7 (Cont')

Crypto	Algorithms	Data Fusion Technique	MAE	RMSE	MAPE
XMR	SVR	Concatenation	9.223	11.8375	0.0626
		Average	9.2581	11.9111	0.063
		Concatenation & Average	6.3214	8.3127	0.0423
	RF	Concatenation	2.9935	4.3091	0.0202
		Average	4.3177	5.6485	0.0292
		Concatenation & Average	9.2645	11.9252	0.0631
	GRU	Concatenation	**2.9551**	**4.2898**	**0.0199**
		Average	3.1351	4.5247	0.0211
		Concatenation & Average	3.2774	4.5402	0.0221
	LSTM	Concatenation	3.3847	4.7878	0.0229
		Average	3.7824	5.1085	0.0255
		Concatenation & Average	6.2106	8.6706	0.0416
LTC	SVR	Concatenation	5.8843	7.3249	0.0763
		Average	5.6584	6.9563	0.0744
		Concatenation & Average	4.937	6.6667	0.0669
	RF	Concatenation	2.1872	3.1743	0.0294
		Average	3.054	4.3306	0.0413
		Concatenation & Average	5.6511	6.9442	0.0744
	GRU	Concatenation	**2.1748**	**3.1730**	**0.0294**
		Average	2.2457	3.3039	0.0302
		Concatenation & Average	2.3244	3.3273	0.0312
	LSTM	Concatenation	2.3746	3.4095	0.0319
		Average	2.5187	3.6559	0.0341
		Concatenation & Average	5.4504	7.6222	0.0721

In the 30-day window period prediction, the market volatility of BTC, ETH, and XRP is relatively stable, with strong trends. The GRU + Average method smooths the data and reduces the interference from sharp fluctuations, enabling better capture of long-term market trends. In contrast, XMR and LTC exhibit higher volatility, influenced by factors such as privacy protection demand and LTC's role as a "lite" version of Bitcoin, resulting in more complex price changes. The GRU + Concatenation method effectively preserves the unique information of each cryptocurrency, allowing it to better capture the dramatic fluctuations and market characteristics of these coins.

After experimenting with all models, we observed from the results that the unified model using LSTM + Average-based data fusion performs best for short-term prediction (5 days and 7 days), while the unified model using GRU + Average-based data fusion is most suitable for long-term prediction (24 days and 30 days). Based on this, these two models are proposed as the unified models in this study, with prediction results shown in Figure 4.1 to Figure 4.4.

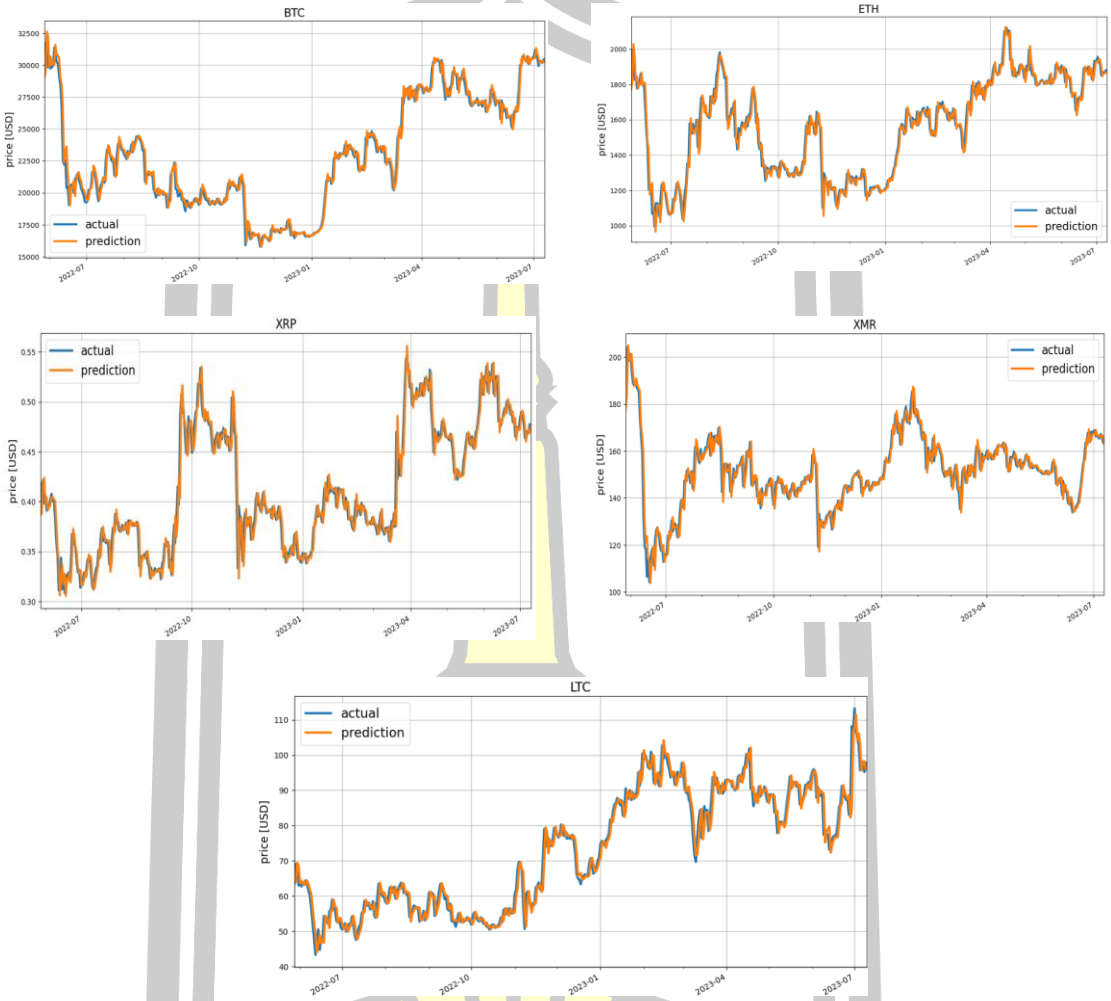


Figure 4.1 LSTM + Average-based data fusion model for BTC, ETH, XRP, XMR, and LTC prediction in 5-day period

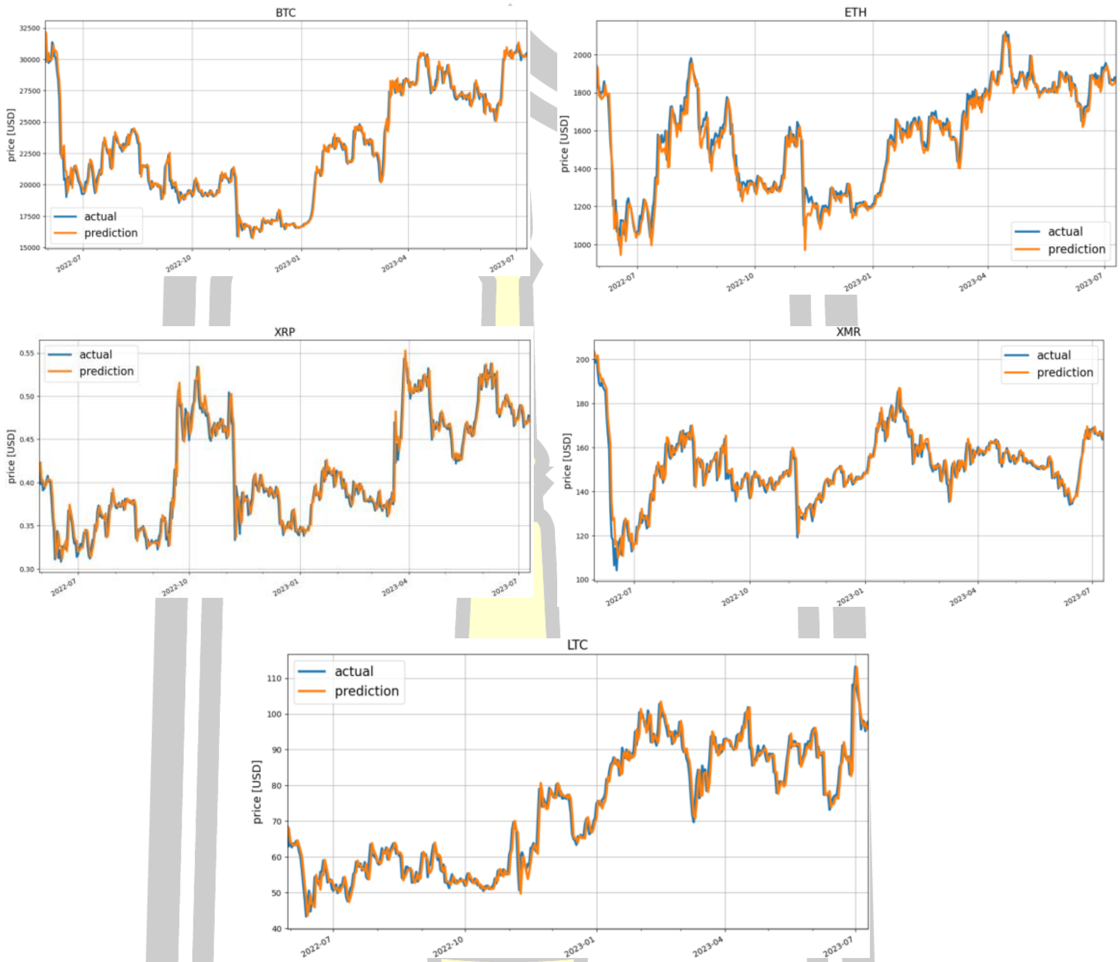


Figure 4.2 LSTM + Average-based data fusion model for BTC, ETH, XRP, XMR, and LTC prediction in 7-day period

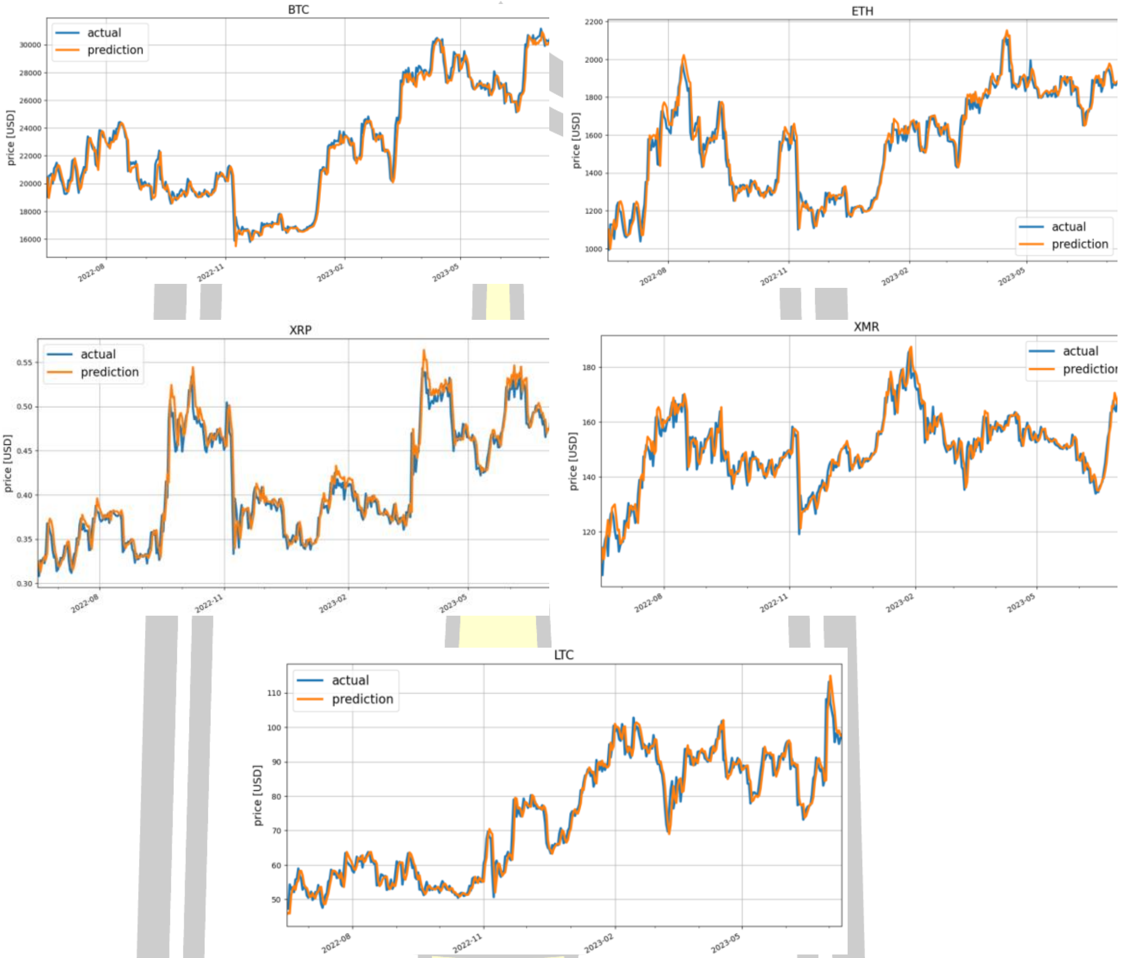


Figure 4.3 GRU+ Average-based data fusion model for BTC, ETH, XRP, XMR, and LTC prediction in 24-day period

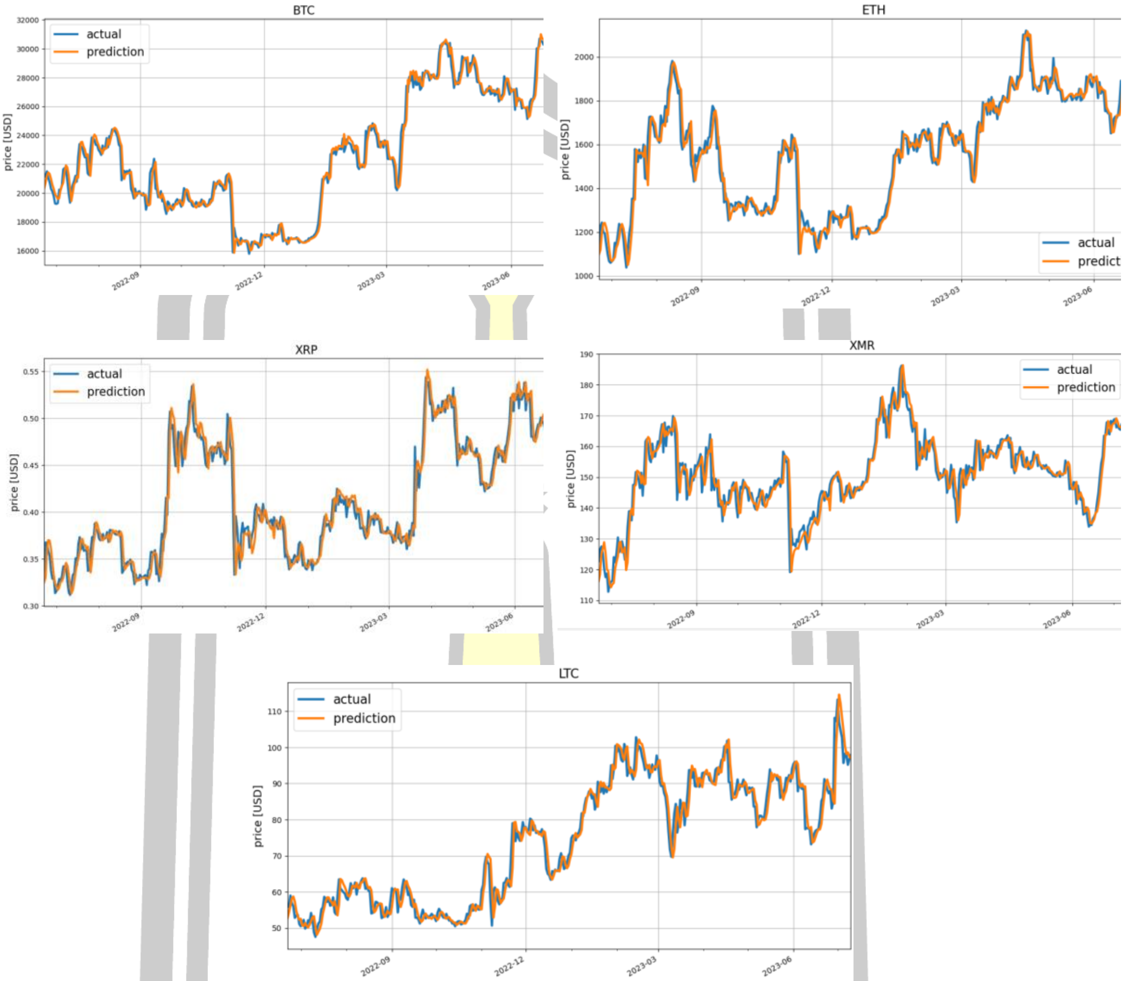


Figure 4.4 GRU+ Average-based data fusion model for BTC, ETH, XRP, XMR, and LTC prediction in 30-day period

The following chapters will compare the results of the proposed unified models with the baseline models and provide an analysis based on different window periods.

4.5 Results of the Baselines

The baseline models designed for individual cryptocurrencies refer to models developed specifically for a single cryptocurrency. For example, if BTC is used as the dataset, the model would be designed exclusively to predict closing prices based on BTC data. This section presents the results of the baseline models for both short-term and long-term periods.

Table 4.8 The results for baseline models in short-term periods

Cryptocurrencies used for study		Baselines	5-Day			7-Day		
Train Set	Test Set		MAE	RMSE	MAPE	MAE	RMSE	MAPE
BTC	BTC	SVR	842.0395	1215.3150	0.0369	848.4917	1190.4752	0.0372
		RF	699.7695	1037.5279	0.0307	801.9857	1196.6652	0.0355
		GRU	481.1279	697.7675	0.0210	458.3757	678.4239	0.0201
		LSTM	**462.0444**	**684.8737**	**0.0201**	**449.3799**	**677.4787**	**0.0198**
ETH	ETH	SVR	70.6952	95.2088	0.0463	75.0912	102.2256	0.0496
		RF	60.4450	85.6342	0.0404	68.8783	97.4716	0.0464
		GRU	42.9865	60.8509	0.0284	43.0340	60.7438	0.0285
		LSTM	**41.7654**	**59.2752**	**0.0275**	**40.5528**	**58.2431**	**0.0269**
XRP	XRP	SVR	0.0195	0.0280	0.0475	0.0201	0.0287	0.0490
		RF	0.0135	0.0200	0.0329	0.0150	0.0219	0.0364
		GRU	0.0130	0.0192	0.0317	0.0112	0.0166	0.0274
		LSTM	**0.0123**	**0.0179**	**0.0299**	**0.0106**	**0.0158**	**0.0260**
XMR	XMR	SVR	6.4466	8.9819	0.0434	5.5205	7.6043	0.0374
		RF	4.4214	6.6964	0.0303	4.9163	7.5722	0.0340
		GRU	3.4142	**4.9942**	0.0231	3.2859	**4.7168**	0.0223
		LSTM	**3.3660**	5.0191	**0.0229**	**3.2808**	4.8327	**0.0223**
LTC	LTC	SVR	3.8569	5.1995	0.0541	3.7831	5.0497	0.0521
		RF	3.0177	4.1704	0.0415	3.3535	4.4891	0.0461
		GRU	3.1621	4.5877	0.0430	2.1710	**3.1084**	0.0301
		LSTM	**2.3247**	**3.4318**	**0.0321**	**2.1689**	3.2011	**0.0300**

It can be observed from Table 4.8 that LSTM outperforms other models, as it effectively captures complex patterns in time series data, thereby improving prediction accuracy. GRU follows closely behind, and while it is slightly less effective than LSTM in capturing intricate dependencies, its simpler structure and higher computational efficiency still contribute to strong prediction performance. On the other hand, the results of SVR and RF are significantly inferior to those of LSTM and GRU. The specific reason is that SVR, when handling time series data, fails to effectively capture temporal features, relying mainly on global features, which leads to poor performance in time series prediction. Although RF can handle nonlinear relationships, it lacks the ability to model the dependencies between time steps in the time series, which results in its performance being far behind that of LSTM and GRU in time series prediction.

Table 4.9 The results for baseline models in long-term periods

Cryptocurrencies used for study		Baselines	24-Day			30-Day		
Train Set	Test Set		MAE	RMSE	MAPE	MAE	RMSE	MAPE
BTC	BTC	SVR	619.8348	849.1199	0.0273	584.0126	796.2550	0.0260
		RF	1539.1438	1997.9350	0.0690	1678.6940	2109.8905	0.0745
		GRU	**443.6136**	**652.1644**	**0.0197**	**432.6146**	**638.1039**	**0.0190**
		LSTM	444.2926	653.4323	0.0199	445.4531	645.0431	0.0198
ETH	ETH	SVR	53.7599	75.1120	0.0359	54.5601	76.2142	0.0362
		RF	128.7657	157.8385	0.0877	137.6989	166.5233	0.0924
		GRU	**40.0215**	**57.4802**	**0.0265**	**39.3998**	**56.7796**	**0.0258**
		LSTM	40.3173	58.2687	0.0268	40.5931	57.7544	0.0267
XRP	XRP	SVR	0.0154	0.0222	0.0371	0.0149	0.0214	0.0354
		RF	0.0269	0.0353	0.0630	0.0291	0.0379	0.0674
		GRU	**0.0108**	**0.0160**	**0.0261**	**0.0112**	**0.0164**	**0.0269**
		LSTM	0.0109	0.0161	0.0263	0.0115	0.0166	0.0273
XMR	XMR	SVR	3.8739	5.2471	0.0264	3.7272	5.0385	0.0251
		RF	8.7729	11.6970	0.0608	9.2219	11.8324	0.0626
		GRU	**3.1573**	**4.5180**	**0.0215**	3.3299	4.6209	0.0224
		LSTM	3.1619	4.5369	0.0216	**3.1671**	**4.5110**	**0.0213**
LTC	LTC	SVR	2.8709	4.1907	0.0387	2.8244	4.0207	0.0380
		RF	5.4554	6.7897	0.0715	5.9236	7.3880	0.0767
		GRU	**2.2552**	**3.2217**	**0.0306**	2.3105	3.3126	0.0310
		LSTM	2.3975	3.3869	0.0326	**2.2631**	**3.2740**	**0.0303**

By analyzing the baseline model results for long-term periods in Table 4.9, it can be concluded that GRU outperforms the other models. Compared to LSTM, as the window period increases and data redundancy emerges, LSTM tends to overfit. In contrast, GRU, with its simpler structure, is better able to adapt to changes in the data, resulting in superior performance in long-term window periods.

In summary, based on the experimental results of the baseline models for short-term and long-term window periods, it is concluded that the LSTM model performs best in short-term windows, while the GRU model is the most suitable for long-term windows. This conclusion is consistent with the proposed unified model. The following section will compare the results of the best baseline models and the proposed unified models for both short-term and long-term windows.

4.6 Comparison of the Proposed Models with the Best Baselines

This section provides a detailed comparison and improvement analysis of the best baseline models and the proposed unified models across four prediction periods. We not only compare the prediction results of both models but also analyze the performance improvement percentage of the unified model relative to the baseline model. The improvement effect of the unified model is quantified through the enhancement rates of MAE, RMSE, and MAPE metrics. The comparison results and improvement analysis are presented in Table 4.10 to Table 4.13, with Table 4.14 summarizing the overall performance improvement across different prediction periods.

Table 4.10 5-day: Baseline vs Unified model performance and improvement analysis

Period	Cryptocurrency	Metric	Baseline Model using LSTM	Unified Model using LSTM + Average-based data fusion	Improvement (%)
5-Day	BTC	MAE	462.0444	458.6837	0.73%
		RMSE	684.8737	690.9105	-0.88%
		MAPE	0.0201	0.02	0.50%
	ETH	MAE	41.7654	41.0975	1.60%
		RMSE	59.2752	59.0418	0.39%
		MAPE	0.0275	0.0271	1.45%
	XRP	MAE	0.0123	0.0105	14.63%
		RMSE	0.0179	0.0154	13.97%
		MAPE	0.0299	0.0257	14.05%
	XMR	MAE	3.366	3.3581	0.23%
		RMSE	5.0191	4.8506	3.36%
		MAPE	0.0229	0.0227	0.87%
	LTC	MAE	2.3247	2.1813	6.17%
		RMSE	3.4318	3.1166	9.18%
		MAPE	0.0321	0.0302	5.92%
	Average	MAE	-	-	**4.67%**
		RMSE	-	-	**5.20%**
		MAPE	-	-	**4.56%**

As shown in Table 4.10, in the 5-day prediction period, the unified model demonstrates significant performance improvements for all cryptocurrencies compared to the baseline LSTM model. Particularly for XRP cryptocurrency, all three metrics achieved improvements of over 13%, indicating that the fusion method is especially effective in capturing short-term fluctuations. In terms of average performance, the unified model in the 5-day period achieved improvements of 4.67%, 5.20%, and 4.56% in MAE, RMSE, and MAPE metrics respectively, representing the most significant improvements among all prediction periods.

Table 4.11 7-day: Baseline vs Unified model performance and improvement analysis

Period	Cryptocurrency	Metric	Baseline Model using LSTM	Unified Model using LSTM + Average-based data fusion	Improvement (%)
7-Day	BTC	MAE	449.3799	447.0709	0.51%
		RMSE	681.4787	677.2853	0.62%
		MAPE	0.0198	0.0196	1.01%
	ETH	MAE	40.5528	40.1942	0.88%
		RMSE	58.2431	58.7056	-0.79%
		MAPE	0.0269	0.0268	0.37%
	XRP	MAE	0.0106	0.0102	3.77%
		RMSE	0.0158	0.0153	3.16%
		MAPE	0.026	0.025	3.85%
	XMR	MAE	3.2808	3.275	0.18%
		RMSE	4.8327	4.7572	1.56%
		MAPE	0.0223	0.0223	0.00%
	LTC	MAE	2.1689	2.1589	0.46%
		RMSE	3.2011	3.1371	2.00%
		MAPE	0.03	0.0299	0.33%
	Average	MAE	-	-	**1.16%**
		RMSE	-	-	**1.31%**
		MAPE	-	-	**1.11%**

Table 4.11 presents the comparison results for the 7-day prediction period. Within this period, the performance improvement of the unified model is somewhat reduced compared to the 5-day period, but still maintains a positive enhancement. XRP continues to show the best performance, with MAE improving by 3.77% and MAPE by 3.85%. However, it is worth noting that ETH's RMSE metric shows a slight decrease (-0.79%). On average, the improvements in the 7-day period for MAE, RMSE, and MAPE are 1.16%, 1.31%, and 1.11% respectively, indicating that the unified model maintains a stable prediction advantage through the fusion of multi-source data.

Table 4.12 24-day: Baseline vs Unified model performance and improvement analysis

Period	Cryptocurrency	Metric	Baseline Model using GRU	Unified Model using GRU + Average-based data fusion	Improvement (%)
24-Day	BTC	MAE	443.6136	440.8092	0.63%
		RMSE	652.1644	637.4319	2.26%
		MAPE	0.0197	0.0197	0.00%
	ETH	MAE	40.0215	39.8004	0.55%
		RMSE	57.4802	56.9062	1.00%
		MAPE	0.0265	0.0264	0.38%
	XRP	MAE	0.0108	0.0105	2.78%
		RMSE	0.016	0.0156	2.50%
		MAPE	0.0261	0.0255	2.30%
	XMR	MAE	3.1573	3.1155	1.32%
		RMSE	4.518	4.4702	1.06%
		MAPE	0.0215	0.0212	1.40%
	LTC	MAE	2.2552	2.246	0.41%
		RMSE	3.2217	3.3153	-2.91%
		MAPE	0.0306	0.0306	0.00%
	Average	MAE	-	-	**1.14%**
		RMSE	-	-	**0.78%**
		MAPE	-	-	**0.82%**

In Table 4.12, the results for the 24-day prediction period show that despite the increased difficulty of prediction with longer time windows, Unified Model using GRU + Average-based data fusion still outperforms the baseline GRU model. BTC's RMSE improved significantly by 2.26%, while XRP's metrics continued to maintain stable improvements of over 2%. However, LTC's RMSE showed a decline of -2.91%, indicating that for certain cryptocurrencies, challenges remain in long-term prediction accuracy. Overall, the average improvements in MAE, RMSE, and MAPE for the 24-day window are 1.14%, 0.78%, and 0.82% respectively. Although these improvements are lower compared to short-term periods, they still maintain a positive enhancement.

Table 4.13 30-day: Baseline vs Unified model performance and improvement analysis

Period	Cryptocurrency	Metric	Baseline Model using GRU	Unified Model using GRU + Average-based data	Improvement (%)
30-Day	BTC	MAE	432.6146	432.1297	0.11%
		RMSE	638.1039	637.3059	0.12%
		MAPE	0.019	0.0193	-1.58%
	ETH	MAE	39.3998	39.5285	-0.33%
		RMSE	56.7796	57.2407	-0.81%
		MAPE	0.0258	0.026	-0.78%
	XRP	MAE	0.0112	0.0107	4.46%
		RMSE	0.0164	0.0157	4.27%
		MAPE	0.0269	0.0258	4.09%
	XMR	MAE	3.3299	3.1351	5.85%
		RMSE	4.6209	4.5247	2.08%
		MAPE	0.0224	0.0211	5.80%
	LTC	MAE	2.3105	2.2457	2.80%
		RMSE	3.3126	3.3039	0.26%
		MAPE	0.031	0.0302	2.58%
	Average	MAE	-	-	**2.58%**
		RMSE	-	-	**1.18%**
		MAPE	-	-	**2.02%**

Table 4.13 presents the comparison results for the 30-day long-term prediction period. the performance differences of the unified model across various cryptocurrencies significantly increase. XMR shows the most outstanding performance with improvements of 5.85% and 5.80% in MAE and MAPE respectively; meanwhile, certain metrics for BTC and ETH show slight negative changes, with ETH declining across all three metrics. This may indicate that for cryptocurrencies with higher market capitalization, the influence of market sentiment and external factors in long-term prediction is more complex. Nevertheless, the average improvements for the 30-day period still reach 2.58% for MAE, 1.18% for RMSE, and 2.02% for MAPE, showing an increase compared to the 24-day period.

Table 4.14 Overall improvement: Baselines vs Unified Models

Prediction Window	MAE Average Improvement	RMSE Average Improvement	MAPE Average Improvement
5-Day	4.67%	5.20%	4.56%
7-Day	1.16%	1.31%	1.11%
24-Day	1.14%	0.78%	0.82%
30-Day	2.58%	1.18%	2.02%
Overall Average	**2.39%**	**2.12%**	**2.13%**

From the overall improvement analysis in Table 4.14, it can be seen that the unified models achieved performance improvements across all prediction periods, with average improvement rates of 2.39%, 2.12%, and 2.13% for MAE, RMSE, and MAPE metrics respectively. The short-term prediction period (5-day) shows the most significant improvements, indicating that the fusion method is particularly effective in capturing short-term market dynamics. Although the improvement magnitude fluctuates as the prediction period extends, the unified models still demonstrate stable prediction capability. The experimental results prove the effectiveness and robustness of the proposed unified models in cryptocurrency price prediction.

4.7 Comparison of the Proposed Models with Related Works

The following three related papers were selected for comparison because they provide valuable insights into cryptocurrency price prediction using deep learning models. These studies focus on similar cryptocurrencies and prediction tasks, which allow us to effectively highlight the advantages of our proposed unified models.

1. P.L. Seabe [78]: This study applies deep learning techniques for cryptocurrency price prediction, including BTC, ETH, and LTC, which overlap partially with our dataset. Although it does not incorporate data fusion or a unified model, it compares variations of LSTM, providing a useful benchmark. Comparing our models with this study allows us to emphasize the advantages of our approach over traditional single-asset models, reinforcing the contribution of our research.

2. S. Hansun [79]: This paper is relevant because it evaluates BTC price prediction using LSTM, GRU, and Bi-LSTM models, which aligns with the comparison of our models. While it does not include data fusion or a unified approach, it offers a benchmark to assess deep learning performance in cryptocurrency forecasting, helping to demonstrate the benefits of our method.

3. S.O. Birim [80]: This paper also focuses on cryptocurrency prediction, covering multiple cryptocurrencies (BTC, ETH, XRP and XMR). It provides a relevant benchmark for evaluating deep learning performance in multi-crypto forecasting, supporting the novelty of our study.

Table 4.15 The results of comparison between the proposed models and related works

Authors	**Cryptocurrencies**		**Methods**	**RMSE**	**MAPE**
	Train	**Test**			
P. L. Seabe [78]	BTC	BTC	LSTM Bi-LSTM GRU	1029.362	0.036
	ETH	ETH		83.953	0.124
	LTC	LTC		8.025	0.041
S. Hansun [79]	BTC	BTC	LSTM Bi-LSTM GRU	1777.310	0.035
	ETH	ETH		147.850	0.057
S. O. Birim [80]	BTC	BTC	LSTM Bi-LSTM GRU	1992.880	0.033
	ETH	ETH		168.600	0.042
	XRP	XRP		0.079	0.048
	XMR	XMR		16.5341	0.0413
The proposed method	BTC ETH XRP	BTC ETH XRP XMR LTC	Short-term: Unified Model using LSTM + Average-based data fusion	690.911	0.020
				59.042	0.027
				0.015	0.026
				4.851	0.023
				3.117	0.030
			Long-term: Unified Model using GRU + Average-based data fusion	637.432	0.020
				56.906	0.026
				0.016	0.025
				4.470	0.021
				3.315	0.031

By comparing with the three related studies in Table 4.15, we further validate the strength of the proposed unified models. The unified models demonstrate strong performance, indicating that the proposed unified models are effective, highly efficient, and capable of challenging existing models.

4.8 Summary of the Experiments

The experimental results indicate that, in short-term prediction windows (5-day and 7-day), Unified Model using LSTM + Average-based data fusion achieves the best performance. In contrast, for long-term prediction windows (24-day and 30-day), Unified Model using GRU + Average-based data fusion outperforms the others.

Comparing the performance of the unified models with the baseline models, the findings show that our unified models achieved performance improvements across all prediction periods. In the 5-day short-term period, the improvements were most significant, with average enhancements of 4.67%, 5.20%, and 4.56% for MAE, RMSE, and MAPE metrics, respectively. Across all four prediction windows, the unified models achieved average improvement rates of 2.39%, 2.12%, and 2.13% for MAE, RMSE, and MAPE, demonstrating the effectiveness of the fusion method across different time scales. Moreover, comparative analysis with relevant literature further validates the advantages of the unified models. Our models significantly outperform existing research on multiple cryptocurrencies examined in this study.

These results conclusively demonstrate the effectiveness, efficiency, and versatility of the proposed unified models, providing a reliable solution for multi-cryptocurrency price prediction. The unified models can predict close prices more effectively than specific models for each cryptocurrency because they leverage shared features and market trends across different cryptocurrencies. While specific models focus only on a single asset and may miss broader signals, the unified models benefit from a larger and more diverse dataset. This helps reduce overfitting and improves generalization. Additionally, the models can learn how movements in one cryptocurrency may influence or reflect patterns in others, leading to more accurate and robust predictions.

CHAPTER 5 CONCLUSION AND FUTURE WORK

5.1 Conclusion

The cryptocurrency market is renowned for its high volatility and unpredictability, making price prediction an exceedingly challenging task. Previous research has predominantly focused on the historical data of individual cryptocurrencies to train, validate, and test their respective predictive models. While a few studies have considered the correlations between different cryptocurrencies, these models typically rely on data from parent cryptocurrencies to predict specific types of cryptocurrencies. However, such models are limited to predicting a single cryptocurrency or specific types of cryptocurrencies and lack a unified model capable of fully integrating the features of multiple cryptocurrencies.

This study combines three data fusion techniques (Concatenation, Averaging, and Concat&Ave) with advanced machine learning (SVR, RF) and deep learning (LSTM, GRU) algorithms to develop unified models. These models integrate the features of BTC, ETH, and XRP into a unified feature vector, capturing their complex interdependencies. To validate the generalization capability of the unified models, five different cryptocurrencies were selected for prediction across four forecasting windows (short-term: 5-day, 7-day; long-term: 24-day, 30-day). The selection includes dominant cryptocurrencies (BTC, ETH, XRP), as well as derivative (LTC) and independent (XMR) currencies. The performance of the unified models was evaluated using MAE, RMSE, and MAPE metrics.

The experimental results indicate that, in short-term windows, the Unified Model using LSTM + Average-based data fusion achieves the best performance, whereas in long-term windows, the Unified Model using GRU + Average-based data fusion outperforms the others. Across all prediction windows, the unified models demonstrated significant improvements over baseline models, with average enhancement rates of 2.39%, 2.12%, and 2.13% for MAE, RMSE, and MAPE respectively.

Based on the prediction results of this study, the following conclusions can be drawn:

1) Effectiveness of Data Fusion Techniques

By employing three data fusion methods—Concatenation, Averaging, and Concat&Ave—this study successfully integrates the features of BTC, ETH, and XRP, enhancing the model's ability to capture the price movements of multiple cryptocurrencies. Experimental results indicate that the averaging-based data fusion method delivers stable and superior performance in both short-term and long-term predictions, demonstrating its applicability in multi-cryptocurrency price forecasting.

2) Differences Between Short-Term and Long-Term Predictions

The model exhibits varying performance across different forecasting windows. In short-term predictions (5-day and 7-day), the Unified Model using

LSTM with averaging-based data fusion achieves the best performance, highlighting LSTM’s strong capability in capturing short-term market dynamics. In contrast, for long-term predictions (24-day and 30-day), the Unified Model using GRU with averaging-based data fusion yields the most optimal results, indicating that GRU’s unique architecture offers advantages in handling long-term temporal dependencies.

3) Advantages of the Unified Model

Compared to traditional baseline models (which are based on individual cryptocurrencies), the unified model demonstrates stronger generalization capability. It consistently outperforms single-cryptocurrency models in both short-term and long-term forecasting tasks, particularly in multi-cryptocurrency environments, where it effectively captures market trends and enhances predictive accuracy.

4) Broad Application Potential

The proposed unified model is not only applicable to major cryptocurrencies such as BTC, ETH, and XRP but can also be extended to different types of cryptocurrencies, including derivative (LTC) and independent (XMR) currencies. Additionally, the model delivers stable forecasting results across various time windows, underscoring its practical applicability and robustness in real-world scenarios.

In conclusion, these results conclusively demonstrate the effectiveness of the unified approach in capturing the complex interrelationships among cryptocurrencies and leveraging this information to improve prediction accuracy. The consistent performance across different cryptocurrencies and time scales further validates the robustness and generalizability of the unified models, offering a promising solution for cryptocurrency price forecasting in volatile market conditions.

5.2 Research Limitation

1) Data Source Reliability

This research relies on data provided by a single cryptocurrency exchange platform. Although the data is continuous and complete, discrepancies in data collection methods and publication cycles across different platforms may lead to inconsistencies that could introduce bias into the model's predictions. Moreover, the data may be susceptible to external factors such as black swan events or market manipulation, which could impact its reliability.

2) Limitations in Feature Engineering

While this study employs validated feature fusion techniques to integrate the OHLCV (Open, High, Low, Close, and Volume) features of three major cryptocurrencies (BTC, ETH, and XRP), it assumes that the influence of these three cryptocurrencies is equal, with all feature weight coefficients set to 1. This treatment does not thoroughly analyze the relative importance of these five features. Such an approach may limit the model's ability to effectively capture the complex interactions between different cryptocurrencies, which could, in turn, affect the overall predictive accuracy and performance of the model.

3) Model Complexity

The deep learning models used in this study, such as LSTM and GRU, have relatively simple architectures with only a few hidden layers. While this simplified model reduces computational costs, it may not be sufficient to address the complexity of time series data. Furthermore, the model's generalization ability may be constrained by its architecture, potentially limiting its capacity to capture the non-linear and long-term dependencies in the cryptocurrency market.

5.3 Future Works

1) Enriching Feature Dimensions

On the feature level, future research should consider incorporating more diverse external variables, such as market sentiment data (extracted from social media, news, etc.) and macroeconomic indicators (e.g., interest rates, inflation rates, policy changes). These additional feature variables could help the model better capture nonlinear relationships and the impact of unexpected events on cryptocurrency prices. Feature selection algorithms, such as recursive feature elimination or L1 regularization, could be employed to ensure that the relevance of these additional features is thoroughly explored while avoiding overfitting.

2) Improving Model Accuracy

Although this study achieved good performance in short-term predictions by integrating features from multiple cryptocurrencies, the model's accuracy decreases as the prediction horizon extends. Future work could explore enhancing predictive accuracy through ensemble learning techniques, such as combining LSTM and GRU, or hybrid models that integrate SVR and deep learning algorithms. Additionally, automated hyperparameter tuning techniques, such as Bayesian optimization or grid search, could be employed to further optimize the model's hyperparameters, thereby improving both accuracy and generalization capabilities.

3) Expanding Application Scenarios

In addition to applications within mainstream cryptocurrencies, future studies could extend the model to other areas, such as small-cap cryptocurrencies or other financial markets (e.g., stock markets or foreign exchange markets). This cross-market generalization would not only test the model’s adaptability across different domains but also provide strong theoretical support for further model improvement.

REFERENCES

[1] P. Ciaian, M. Rajcaniova, and D. a. Kancs, "The economics of BitCoin price formation," *Applied economics,* vol. 48, no. 19, pp. 1799-1815, 2016 2016.

[2] L. W. Cong, Z. He, and J. Li, "Decentralized mining in centralized pools," *The Review of Financial Studies,* vol. 34, no. 3, pp. 1191-1235, 2021 2021.

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008 2008.

[4] H. Jang and J. Lee, "An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information," *IEEE access,* vol. 6, pp. 5427-5437, 2017 2017.

[5] H. Treiblmaier and R. Beck, *Business transformation through blockchain*. Springer, 2019.

[6] P. Jay, V. Kalariya, P. Parmar, S. Tanwar, N. Kumar, and M. Alazab, "Stochastic neural networks for cryptocurrency price prediction," *Ieee access,* vol. 8, pp. 82804-82818, 2020 2020.

[7] A. M. Khedr, I. Arif, M. El-Bannany, S. M. Alhashmi, and M. Sreedharan, "Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey," *Intelligent Systems in Accounting, Finance and Management,* vol. 28, no. 1, pp. 3-34, 2021 2021.

[8] H. A. Aalborg, P. Molnár, and J. E. de Vries, "What can explain the price, volatility and trading volume of Bitcoin?," *Finance Research Letters,* vol. 29, pp. 255-265, 2019 2019.

[9] E. Bouri, B. Lucey, and D. Roubaud, "The volatility surprise of leading cryptocurrencies: Transitory and permanent linkages," *Finance Research Letters,* vol. 33, p. 101188, 2020 2020.

[10] H. Fatah, R. A. Anggraini, D. Supriadi, M. W. Pertiwi, A. I. Warnilah, and N. Ichsan, "Data mining for cryptocurrencies price prediction," presented at the Journal of Physics: Conference Series, 2020, 2020.

[11] C. Alexander and M. Dakos, "A critical investigation of cryptocurrency data and analysis," *Quantitative Finance,* vol. 20, no. 2, pp. 173-188, 2020 2020.

[12] E. Akyildirim, A. Goncu, and A. Sensoy, "Prediction of cryptocurrency returns using machine learning," *Annals of Operations Research,* vol. 297, pp. 3-36, 2021 2021.

[13] R. Matkovskyy and A. Jalan, "Effects of economic policy uncertainty shocks on the interdependence between cryptocurrency and traditional financial markets," presented at the Cryptocurrency Research Conference, 2019, 2019.

[14] A. Politis, K. Doka, and N. Koziris, "Ether price prediction using advanced deep learning models," presented at the 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 2021, 2021.

[15] E. Bouri, D. Roubaud, and S. J. H. Shahzad, "Do Bitcoin and other cryptocurrencies jump together?," *The Quarterly Review of Economics and Finance,* vol. 76, pp. 396-409, 2020 2020.

[16] W. Bolt and M. R. Van Oordt, "On the value of virtual currencies," *Journal of Money, Credit and Banking,* vol. 52, no. 4, pp. 835-862, 2020.

[17] R. Auer, G. Cornelli, and J. Frost, "Rise of the central bank digital currencies: drivers, approaches and technologies," 2020 2020.

[18] T. Adrian and T. Mancini-Griffoli, "The rise of digital money," *Annual Review of*

Financial Economics, vol. 13, no. 1, pp. 57-77, 2021 2021.

[19] Z. H. Munim, M. H. Shakil, and I. Alon, "Next-day bitcoin price forecast," *Journal of risk and financial management,* vol. 12, no. 2, p. 103, 2019 2019.

[20] G. Giudici, A. Milne, and D. Vinogradov, "Cryptocurrencies: market analysis and perspectives," *Journal of Industrial and Business Economics,* vol. 47, pp. 1-18, 2020 2020.

[21] S. Quamara and A. K. Singh, "A systematic survey on security concerns in cryptocurrencies: State-of-the-art and perspectives," *Computers & Security,* vol. 113, p. 102548, 2022 2022.

[22] R. Ali and N. Narula, "Redesigning digital money: What can we learn from a decade of cryptocurrencies," *Digital Currency Iniative (DCI). MIT Media Lab,* 2020 2020.

[23] U. W. Chohan, "Cryptocurrencies: A brief thematic review," *Available at SSRN 3024330,* 2022 2022.

[24] S. Bhatt, M. Ghazanfar, and M. Amirhosseini, "Machine learning based cryptocurrency price prediction using historical data and social media sentiment," *Computer Science & Information Technology (CS & IT),* vol. 13, no. 10, pp. 1-11, 2023 2023.

[25] N. R. Pokhrel *et al.*, "Predicting nepse index price using deep learning models," *Machine Learning with Applications,* vol. 9, p. 100385, 2022 2022.

[26] D. I. Okorie and B. Lin, "Crude oil price and cryptocurrencies: Evidence of volatility connectedness and hedging strategy," *Energy economics,* vol. 87, p. 104703, 2020 2020.

[27] T. L. D. Huynh, R. Ahmed, M. A. Nasir, M. Shahbaz, and N. Q. A. Huynh, "The nexus between black and digital gold: evidence from US markets," *Annals of Operations Research,* pp. 1-26, 2024 2024.

[28] D. G. W. Birch, "What Does Cryptocurrency Mean for the New Economy?," in *handbook of digital currency*: Elsevier, 2024, pp. 487-499.

[29] N. Thampanya, M. A. Nasir, and T. L. D. Huynh, "Asymmetric correlation and hedging effectiveness of gold & cryptocurrencies: From pre-industrial to the 4th industrial revolution☆," *Technological Forecasting and Social Change,* vol. 159, p. 120195, 2020 2020.

[30] F. Castanedo, "A review of data fusion techniques," *The scientific world journal,* vol. 2013, no. 1, p. 704504, 2013 2013.

[31] M. Huang, Z. Liu, and Y. Tao, "Mechanical fault diagnosis and prediction in IoT based on multi-source sensing data fusion," *Simulation Modelling Practice and Theory,* vol. 102, p. 101981, 2020 2020.

[32] K. Xie, Y. Luo, W. Li, Z. Chen, N. Zhang, and C. Liu, "Deep learning with multisource data fusion in electricity internet of things for electricity price forecast," *Wireless Communications and Mobile Computing,* vol. 2022, no. 1, p. 3622559, 2022 2022.

[33] M. W. Nadeem, H. G. Goh, V. Ponnusamy, I. Andonovic, M. A. Khan, and M. Hussain, "A Fusion-Based Machine Learning Approach for the Prediction of the Onset of Diabetes. Healthcare 2021, 9, 1393," 2021 2021.

[34] H. Haleh, B. A. Moghaddam, and S. Ebrahimijam, "A new approach to forecasting stock price with EKF data fusion," *International Journal of Trade, Economics and Finance,* vol. 2, no. 2, p. 109, 2011 2011.

[35] Y. Yonghui, "Research on house price prediction based on multi-dimensional data fusion," *Associate Editor-in-Chief,* p. 1, 2020 2020.

[36] Y. Guo, "Stock trading based on principal component analysis and clustering analysis," in *IOP Conference Series: Materials Science and Engineering*, 2020, vol. 740, no. 1: IOP Publishing, p. 012129.

[37] P. Zhang *et al.*, "A data-level fusion model for unsupervised attribute selection in multi-source homogeneous data," *Information Fusion,* vol. 80, pp. 87-103, 2022.

[38] X. Zhang and L. Zhang, "[Retracted] Forecasting Method of Stock Market Volatility Based on Multidimensional Data Fusion," *Wireless Communications and Mobile Computing,* vol. 2022, no. 1, p. 6344064, 2022 2022.

[39] N. Kardani, A. Zhou, M. Nazem, and S.-L. Shen, "Improved prediction of slope stability using a hybrid stacking ensemble method based on finite element analysis and field data," *Journal of Rock Mechanics and Geotechnical Engineering,* vol. 13, no. 1, pp. 188-201, 2021 2021.

[40] D.-H. Cho, S.-H. Moon, and Y.-H. Kim, "Genetic feature selection applied to KOSPI and cryptocurrency price prediction," *Mathematics,* vol. 9, no. 20, p. 2574, 2021 2021.

[41] C. Phasook, J. Polpinij, and Others, "Comparative Study for Cryptocurrency Price Prediction Methods," Ph.D. Thesis, Mahasarakham University, 2023.

[42] S. Cavalli and M. Amoretti, "CNN-based multivariate data analysis for bitcoin trend prediction," *Applied Soft Computing,* vol. 101, p. 107065, 2021 2021.

[43] D. Haryadi, A. R. Hakim, D. M. U. Atmaja, and S. N. Yutia, "Implementation of support vector regression for polkadot cryptocurrency price prediction," *JOIV: International Journal on Informatics Visualization,* vol. 6, no. 1-2, pp. 201-207, 2022.

[44] G. Lin, A. Lin, and D. Gu, "Using support vector regression and K-nearest neighbors for short-term traffic flow prediction based on maximal information coefficient," *Information Sciences,* vol. 608, no. 1-2, pp. 517-531, 2022 2022.

[45] D. Parbat and M. Chakraborty, "A python based support vector regression model for prediction of COVID19 cases in India," *Chaos, Solitons & Fractals,* vol. 138, p. 109942, 2020 2020.

[46] D. A. Puspitasari and Z. Rustam, "Application of SVM-KNN using SVR as feature selection on stock analysis for Indonesia stock exchange," presented at the AIP Conference Proceedings, 2018, 2018.

[47] M. S. Ferdous, M. J. M. Chowdhury, and M. A. Hoque, "A survey of consensus algorithms in public blockchain systems for crypto-currencies," *Journal of Network and Computer Applications,* vol. 182, p. 103035, 2021 2021.

[48] A. Elbeltagi *et al.*, "Prediction of meteorological drought and standardized precipitation index based on the random forest (RF), random tree (RT), and Gaussian process regression (GPR) models," *Environmental Science and Pollution Research,* vol. 30, no. 15, pp. 43183-43202, 2023 2023.

[49] J. H. Jeong *et al.*, "Random forests for global and regional crop yield predictions," *PloS one,* vol. 11, no. 6, p. e0156571, 2016.

[50] A. Dutta, S. Kumar, and M. Basu, "A gated recurrent unit approach to bitcoin price prediction," *Journal of risk and financial management,* vol. 13, no. 2, p. 23, 2020 2020.

[51] M. J. Hamayel and A. Y. Owda, "A novel cryptocurrency price prediction model

using GRU, LSTM and bi-LSTM machine learning algorithms," *Ai,* vol. 2, no. 4, pp. 477-496, 2021 2021.

[52] I. Nasirtafreshi, "Forecasting cryptocurrency prices using recurrent neural network and long short-term memory," *Data & Knowledge Engineering,* vol. 139, p. 102009, 2022 2022.

[53] J. Polpinij *et al.*, "Predicting the Close-price of Cryptocurrency Using the Kernel Regression Algorithm," presented at the 2023 Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C), 2023, 2023.

[54] M. Köhler, A. Schindler, and S. Sperlich, "A review and comparison of bandwidth selection methods for kernel regression," *International Statistical Review,* vol. 82, no. 2, pp. 243-274, 2014 2014.

[55] D. Sahoo, S. C. H. Hoi, and B. Li, "Large scale online multiple kernel regression with application to time-series prediction," *ACM Transactions on Knowledge Discovery from Data (TKDD),* vol. 13, no. 1, pp. 1-33, 2019 2019.

[56] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International journal of forecasting,* vol. 22, no. 4, pp. 679-688, 2006 2006.

[57] J. M. González-Sopeña, V. Pakrashi, and B. Ghosh, "An overview of performance evaluation metrics for short-term statistical wind power forecasting," *Renewable and Sustainable Energy Reviews,* vol. 138, p. 110515, 2021 2021.

[58] R. Ahmed, V. Sreeram, Y. Mishra, and M. D. Arif, "A review and evaluation of the state-of-the-art in PV solar power forecasting: Techniques and optimization," *Renewable and sustainable energy reviews,* vol. 124, p. 109792, 2020 2020.

[59] D. Qin, "Rise of VAR modelling approach," *Journal of Economic Surveys,* vol. 25, no. 1, pp. 156-174, 2011.

[60] S. Garg and Others, "Autoregressive integrated moving average model based prediction of bitcoin close price," presented at the 2018 international conference on smart systems and inventive technology (ICSSIT), 2018, 2018.

[61] I. Sadia, A. Mahmood, L. B. M. Kiah, and S. R. Azzuhri, "Analysis and Forecasting of Blockchain-based Cryptocurrencies and Performance Evaluation of TBATS, NNAR and ARIMA," presented at the 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), 2022, 2022.

[62] I. M. Wirawan, T. Widiyaningtyas, and M. M. Hasan, "Short term prediction on bitcoin price using ARIMA method," presented at the 2019 International Seminar on Application for Technology of Information and Communication (iSemantic), 2019, 2019.

[63] V. Derbentsev, A. Matviychuk, and V. N. Soloviev, "Forecasting of cryptocurrency prices using machine learning," *Advanced studies of financial technologies and cryptocurrency markets,* pp. 211-231, 2020 2020.

[64] S. McNally, J. Roche, and S. Caton, "Predicting the price of bitcoin using machine learning," presented at the 2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP), 2018, 2018.

[65] Z. Chen, C. Li, and W. Sun, "Bitcoin price prediction using machine learning: An approach to sample dimension engineering," *Journal of Computational and*

Applied Mathematics, vol. 365, p. 112395, 2020 2020.

[66] D. C. A. Mallqui and R. A. S. Fernandes, "Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques," *Applied Soft Computing,* vol. 75, pp. 596-606, 2019 2019.

[67] M. Poongodi *et al.*, "Prediction of the price of Ethereum blockchain cryptocurrency in an industrial finance system," *Computers & Electrical Engineering,* vol. 81, p. 106527, 2020 2020.

[68] X. Sun, M. Liu, and Z. Sima, "A novel cryptocurrency price trend forecasting model based on LightGBM," *Finance Research Letters,* vol. 32, p. 101084, 2020 2020.

[69] T. Zoumpekas, E. Houstis, and M. Vavalis, "Eth analysis and predictions utilizing deep learning," *Expert Systems with Applications,* vol. 162, p. 113866, 2020 2020.

[70] M. Mudassir, S. Bennbaia, D. Unal, and M. Hammoudeh, "Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach," *Neural computing and applications,* pp. 1-15, 2020 2020.

[71] C.-H. Wu, C.-C. Lu, Y.-F. Ma, and R.-S. Lu, "A new forecasting framework for bitcoin price with LSTM," presented at the 2018 IEEE international conference on data mining workshops (ICDMW), 2018, 2018.

[72] S. Marne, S. Churi, D. Correia, and J. Gomes, "Predicting Price of Cryptocurrency--A Deep Learning Approach," *NTASU-9 (3),* pp. 140-157, 2020 2020.

[73] W. Kristjanpoller and M. C. Minutolo, "A hybrid volatility forecasting framework integrating GARCH, artificial neural network, technical analysis and principal components analysis," *Expert Systems with Applications,* vol. 109, pp. 1-11, 2018 2018.

[74] Y. Li and W. Dai, "Bitcoin price forecasting method based on CNN-LSTM hybrid neural network model," *The journal of engineering,* vol. 2020, no. 13, pp. 344-347, 2020 2020.

[75] Q. Guo, S. Lei, Q. Ye, and Z. Fang, "MRC-LSTM: A hybrid approach of multi-scale residual CNN and LSTM to predict bitcoin price," presented at the 2021 International Joint Conference on Neural Networks (IJCNN), 2021, 2021.

[76] I. A. Hashish, F. Forni, G. Andreotti, T. Facchinetti, and S. Darjani, "A hybrid model for bitcoin prices prediction using hidden Markov models and optimized LSTM networks," presented at the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2019, 2019.

[77] M. M. Patel, S. Tanwar, R. Gupta, and N. Kumar, "A deep learning-based cryptocurrency price prediction scheme for financial institutions," *Journal of information security and applications,* vol. 55, p. 102583, 2020 2020.

[78] P. L. Seabe, C. R. B. Moutsinga, and E. Pindza, "Forecasting cryptocurrency prices using LSTM, GRU, and bi-directional LSTM: a deep learning approach," *Fractal and Fractional,* vol. 7, no. 2, p. 203, 2023 2023.

[79] S. Hansun, A. Wicaksana, and A. Q. M. Khaliq, "Multivariate cryptocurrency prediction: comparative analysis of three recurrent neural networks approaches," *Journal of Big Data,* vol. 9, no. 1, p. 50, 2022 2022.

[80] S. O. Birim, M. Comert, and A. E. Şimşek, "An Analysis for Cryptocurrency Price Prediction Using LSTM GRU and the Bi-Directional Implications," in

Developments in Financial and Economic Fields at the National and Global Scale: Gazi Kitabevi, 2022, pp. 377-392.

[81] Z. Guo, H. Wang, Q. Liu, and J. Yang, "A feature fusion based forecasting model for financial time series," *PloS one,* vol. 9, no. 6, p. e101113, 2014 2014.

[82] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data," *PloS one,* vol. 14, no. 2, p. e0212320, 2019 2019.

[83] S. Tanwar, N. P. Patel, S. N. Patel, J. R. Patel, G. Sharma, and I. E. Davidson, "Deep learning-based cryptocurrency price prediction scheme with inter-dependent relations," *IEEE Access,* vol. 9, pp. 138633-138646, 2021 2021.

[84] N. P. Patel *et al.*, "Fusion in cryptocurrency price prediction: A decade survey on recent advancements, architecture, and potential future directions," *IEEE Access,* vol. 10, pp. 34511-34538, 2022 2022.

[85] N. Maleki, A. Nikoubin, M. Rabbani, and Y. Zeinali, "Bitcoin price prediction based on other cryptocurrencies using machine learning and time series analysis," *Scientia Iranica,* vol. 30, no. 1, pp. 285-301, 2023 2023.

BIOGRAPHY

NAME	YUAN CHEN
DATE OF BIRTH	01-11-1986
PLACE OF BIRTH	China
ADDRESS	Room 1709, Ladder 1, Building 1, Yipin Jiayuan, No. 1030, WenXian East Road, Zhenhai Street, Licheng District, Putian City, Fujian Province
POSITION	full-time lecture
PLACE OF WORK	None
EDUCATION	2022-2025 Doctor of Philosophy Program in Faculty of Informatics - Computer Science, Mahasarakham University 2014-2017 Master's Degree in School of Economics and Management - Business Administration Program, Fujian Agriculture and Forestry University 2004-2008 Bachelor's Degree in Department of Telecommunications - Computer Science and Applications Program, Putian University