KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY, KUMASI

COLLEGE OF SCIENCE

FACULTY OF PHYSICAL AND COMPUTATIONAL SCIENCES

DEPARTMENT OF STATISTICS AND ACTUARIAL SCIENCE

PREDICTING BITCOIN PRICES USING LSTM WITH RANDOMIZED SIGMOID ACTIVATION FUNCTION

GEORGE ROOSEVELT JEFFERSON	9350019
ABIGAIL OFORI TAWIAH	9352119
NATHANIEL SACKEY	9354419
ANASTASIA BOAMPONG	9347419

DEDICATION

We dedicate this report to the Almighty God, who has been with us throughout the entire process, guiding and providing us with strength.We also extend our heartfelt gratitude to our dedicated lecturers, especially Dr. George Awiakye Marfo, for generously sharing their knowledge and taking time out of their busy schedules to mentor and support us in completing this project.

ACKNOWLEDGEMENT

We express our sincere gratitude to Almighty God for granting us the wisdom, insight, and understanding required to conduct this study. We are deeply thankful to Dr. George Awiakye Marfo, our dedicated supervisor, whose guidance, patience, and profound knowledge were invaluable throughout our research journey.

Furthermore, we extend our heartfelt appreciation to our friends and colleagues at Kwame Nkrumah University of Science and Technology, whose contributions and support significantly contributed to the success of this research endeavor.

Contents

List of Figures

Abstract

The recent surge in interest in machine learning and AI-assisted trading has transformed the cryptocurrency market, especially Bitcoin. To extract exceptional profits from this dynamic market, traders and investors have increasingly turned to data-driven strategies.This research focuses on enhancing Bitcoin price prediction models by introducing randomized activation function for the Long Short-Term Memory (LSTM) model.The closing price of Bitcoin was flitted using both standard and randomised LSTM models,and a five day forecast made. The randomised LSTM model produced a lower Mean Absolute Error (MAE) of 0.007620 compared to the standard LSTM model, which had an MAE of 0.0087097.

Chapter 1

INTRODUCTION

1.1 BACKGROUND OF STUDY

Bitcoin, a virtual currency widely embraced for both transactions and investment purposes, operates in a decentralized manner, signifying its independence from single individuals or entities. This unique characteristic of Bitcoin stems from its detachment from any specific nation or government authority, making it remarkably accessible and versatile. Investing in Bitcoin is facilitated through Bitcoin exchanges, offering a seamless platform for individuals to buy and sell Bitcoin using a variety of fiat currencies. (Dhanya, N. M, 2021; Farokhmanesh, F., and Sadeghi, M. T., 2019) .Notably, the cryptocurrency landscape witnessed significant growth, with the launch of 170 hedge funds dedicated to cryptocurrencies by January 2017. This surge in cryptocurrency-oriented hedge funds played a pivotal role in driving up the demand for Bitcoin, both as a trading instrument and as a tool for hedging future uncertainties. Amidst this growth, numerous conspiracy theories emerged, attempting to explain the factors behind the cryptocurrency's high volatility. These theories also speculated that the value of cryptocurrencies, including Bitcoin, would continue to exhibit fluctuations in the foreseeable future.In light of these dynamics, some individuals have turned to automated Bitcoin trading as an alternative approach to engaging with this evolving financial landscape (S. Yogeshwaran et al., 2019; F. Andrade de Oliveira et al., 2011) the deep learning model could be supported by Django, and a

graphical web app may be created. However, the convolutional Neural Network was originally created to research and was made to specifically forecast a succession of photos to identify numbers. Although it was inadequate with a 5 percent buffer. It was still able to compete with the LSTM model in terms of amount. The claim was made that forecasting models are going to become more sophisticated and efficient as a result of the expansion of data collection and the creation of more powerful data analysis techniques. The sole potential factor is the demand for increased computational power is what's holding us back.Fredman (2008) argued that complete realism is unattainable, and that the question of whether a theorem (model) is realistic enough can only be settled by saying whether it yields price predictions that are good enough for a purpose in hand or that are better than price predictions from alternative theories (models). According to the National Bureau of Economic and International Monetary Fund research, the global financial crisis of 2007-2009 was the most severe crisis over the last few decades. The consequences within the year 2007-2009, a peak to trough of 18 months were severe in most aspects of life including the economy (investment, productivity, jobs and real income), social (inequality, poverty, and social tensions), leading in the long run to political instability and the need for further economic reforms.In an attempt to think ahead and evade financial institutions and government controls and manipulation, a pseudonymous person named Satoshi Nakamoto invented the first cryptocurrency, Bitcoin, in the early year of 2009. Cryptocurrency does not have a central issuing or regulating authority, instead uses a decentralized system to record transactions and issue new units.Satoshi Nakamoto proposed Bitcoin which is an electronic cash allowing online payments, where the double-spending problem was elegantly solved using a novel purely peer-to-peer decentralized blockchain along with cryptographic hash function as a proof of work. There are other types of cryptocurrency after Bitcoin which are Tether, Etherum, Binance coin, Litcoin, Ripple et al available.Mathematical analysis has a well-established place in the finance industry for evaluating expected return of a stock of a given performance of an entire portfolio. However, Machine learning and Deep learning literature is lacking verification of whether the stock techniques are valid for

the cryptocurrency, and if so, how they can be modified.That is, what features to be removed or added as a basis for price prediction, whether current machine learning and deep learning algorithms work for cryptocurriencies, and which approach yields the best results. Both individuals and large financial firms are attracted to cryptocurrencies because of the transparency and anonymity that they provide to their users, as well as their resistance to fraud due to the distributed nature of the ledger records.A cryptocurrency market has a relatively short history as compared to the stock market. A cryptocurrency price data has arbitrary characteristics as time series data, but the end has higher volatility and their prices fluctuate heavily. The cryptocurrency market is quite different from the existing financial markets and it has new or updated features. Thus, in addition to the existing stock market prediction techniques, it is sufficient and necessary to apply updated prediction techniques suitable for the cryptocurrency market. Although numerous related studies have been conducted in the field of financial stock prices, there a few studies on cryptocurrency price prediction.

1.2 PROBLEM STATEMENT

Bitcoin's price volatility is driven by various factors, making accurate price predictions challenging yet crucial for traders and investors. This research employs the Recurrent Neural Networks (RNNs) with Long-Short Term Memory (LSTM).We propose a randomized activation function for the RNN with LSTM model to enhance the prediction accuracy of the LSTM model. The successful outcome of this research will provide valuable tools for cryptocurrency market participants, improving decision-making and contributing to our understanding of digital asset markets.

1.3 OBJECTIVES

At the conclusion of our research, we hope to achieve the following:

- To fit an LSTM model to Bitcoin prices.

- To propose a randomised activation function for the LSTM model.
- To utilize the proposed model to predict Bitcoin prices.

1.4 SIGNIFICANCE OF THE STUDY

The significance of this study lies in its potential to address the pressing need for accurate Bitcoin price predictions. As the cryptocurrency market gains prominence and attracts a diverse range of participants, including traders and investors, the ability to make well-informed decisions becomes crucial. This research leverages advanced machine learning techniques, particularly the LSTM model with a randomized activation function, to enhance our capacity to forecast Bitcoin prices with precision.By achieving more accurate predictions, this study can empower cryptocurrency market participants to navigate the volatile landscape with greater confidence, potentially leading to more profitable trading and investment decisions. Moreover, the innovative approach of introducing randomness through the beta adds adaptability to the model, which is vital in a rapidly evolving market.Beyond individual traders and investors, the implications of this research extend to the broader financial industry, as cryptocurrencies continue to integrate into traditional portfolios. Therefore, the success of this study could contribute valuable insights and methodologies that advance our understanding of cryptocurrency markets, fostering greater confidence and success for all stakeholders.

1.5 RESEARCH QUESTION

"How can the application of Recurrent Neural Networks (RNNs) with Long-Short Term Memory (LSTM) architecture, augmented with a randomized activation function parameter, improve the precision and confidence of Bitcoin price predictions, and what are the broader implications of such advancements for cryptocurrency market participants and the financial industry?" This research question addresses the core focus of our study, which involves enhancing Bitcoin price predictions using advanced

machine learning techniques and evaluating the wider impact on stakeholders in the cryptocurrency market and the financial industry.

1.6 ORGANIZATION OF THE STUDY

There are five chapters in this research. The **Chapter 1** of this study discusses the background information and the foundations for our research. Our problem statement and a summary of the study's objectives are also included. **Chapter 2** provides a concise review of existing literature related to cryptocurrency, specifically Bitcoin, price prediction, and compares various prediction models. **Chapter 3** delves into the details of the proposed Long Short-Term Memory (LSTM) model, offering insights into techniques for optimizing model parameters. **Chapter 4** covers the data collection and preprocessing methods employed for cryptocurrency price time series data. It also explains how the training and target data are prepared for Recurrent Neural Network (RNN) with LSTM models. Additionally, this chapter presents data visualization and an evaluation of the results achieved by the model. In conclusion and recommendation, **Chapter 5** summarizes the findings,implications of research and provides recommendations for future research.

Chapter 2

LITERATURE REVIEW

In this chapter, we review the relevant literature pertaining to cryptocurrency trading, Bitcoin price prediction, and the application of Recurrent Neural Networks (RNN) with Long-Short Term Memory (LSTM) models in this context.

2.1 Cryptocurrency Trading and Its Challenges

Cryptocurrency trading has gained significant popularity, driven by the emergence of online platforms known as cryptocurrency exchanges. These platforms enable users to engage in buying, selling, and trading various cryptocurrencies using fiat currencies or other cryptocurrencies. The trading process encompasses several key steps, including account creation, funding, cryptocurrency selection, order placement, trade execution, and fund withdrawal.It is essential to highlight the high degree of risk and volatility associated with cryptocurrency trading. Users are advised to conduct thorough research and exercise caution before entering this market. Moreover, cryptocurrency regulations and restrictions vary widely across countries and jurisdictions, adding another layer of complexity to this form of trading.One distinctive feature of the cryptocurrency market is its decentralization and 24/7 availability, contrasting with traditional stock exchanges that operate within specific trading hours. This continuous accessibility allows traders to respond promptly to market events, potentially yielding profits without intermediary involvement. However, it also demands constant online presence, as any absence can result in missed price fluctuations.

Cryptocurrencies, particularly Bitcoin, exhibit remarkable volatility compared to traditional stock markets. The influence of brokers on price manipulation further exacerbates this volatility, leading to sudden and significant price fluctuations. Additionally, cryptocurrency investors must take responsibility for securing their assets, which can be challenging due to the evolving nature of storage solutions.

2.2 Overview of Cryptocurrency

Cryptocurrency is traded on various online platforms known as cryptocurrency exchanges. These exchanges allow users to buy, sell, and trade cryptocurrencies using different fiat currencies or other cryptocurrencies. The trading process generally involves the following: Creating an account: Users need to create an account on a cryptocurrency exchange by providing their personal information and verifying their identity.Funding the account: Users can fund their account with fiat currency or cryptocurrencies through various payment methods, such as bank transfers, credit/debit cards, or digital wallets. Choosing a cryptocurrency: Once the account is funded, users can choose the cryptocurrency they want to trade. Placing an order: Users can place different types of orders, such as market orders, limit orders, or stop-loss orders, to buy or sell the chosen cryptocurrency at a specific price. Executing the trade: When the order is matched with a counterparty, the trade is executed, and the cryptocurrency is transferred to the user's account.Withdrawing funds: Users can withdraw their funds by transferring the cryptocurrency to their digital wallet or fiat currency to their bank account.

It's important to note that cryptocurrency trading involves a high degree of risk and volatility, and users should do their own research and exercise caution before investing. Additionally, cryptocurrency regulations and restrictions vary by country and jurisdiction, so users should also be aware of the legal implications of trading cryptocurrencies in their location.

2.2.1 History of Cryptocurrency

The history of cryptocurrency is a captivating journey that spans several decades and pivotal moments and that traces the development and evolution of digital currencies.In the 1980s, the concept of digital currency took its initial steps. American cryptographer David Chaum introduced the idea of "eCash," which was essentially a cryptographic electronic cash system. However, during this period, it did not gain widespread adoption.Moving into the 1990s, David Chaum founded a company called DigiCash with the ambitious goal of creating a digital currency system. DigiCash's eCash represented one of the earliest attempts at digital cash. Despite its innovative approach, DigiCash faced challenges and eventually filed for bankruptcy in 1998. The true breakthrough in the world of cryptocurrency occurred in 2008 when an individual or group operating under the pseudonym "Satoshi Nakamoto" published the Bitcoin whitepaper titled "Bitcoin: A Peer-to-Peer Electronic Cash System." In January 2009, Nakamoto mined the first-ever Bitcoin block, often referred to as the "genesis block." This marked the birth of the Bitcoin blockchain. Bitcoin began to gain recognition, and its first recorded commercial transaction took place in May 2010 when a programmer named Laszlo Hanyecz exchanged a staggering 10,000 Bitcoins for two pizzas.The success of Bitcoin inspired the creation of numerous alternative cryptocurrencies, often referred to as "altcoins." One of the earliest altcoins, Litecoin, was launched in October 2011 by Charlie Lee. In late 2013, Bitcoin experienced a significant surge in its price, reaching over 1,000 Dollars per Bitcoin. This surge in value led to the rapid expansion of the cryptocurrency ecosystem. Exchanges, wallets, and businesses emerged to support the adoption of cryptocurrencies. Ethereum, proposed by Vitalik Buterin in late 2013 and developed in 2014, introduced a groundbreaking concept known as "smart contracts." Ethereum went live in 2015, enabling self-executing contracts with terms written directly into code. This innovation paved the way for decentralized applications (DApps).The year 2017 witnessed the explosive growth of Initial Coin Offerings (ICOs). Blockchain startups used ICOs to raise funds by issuing their tokens, some of which raised millions of dollars within minutes. However, this period also saw substantial price volatility in the cryptocurrency market.

From 2018 to 2020, governments and regulators around the world began addressing cryptocurrency regulations, impacting exchanges, ICOs, and taxation. Stablecoins, like Tether (USDT), gained prominence for providing stable values pegged to traditional currencies. The 2020s ushered in the era of decentralized finance (DeFi) platforms, offering decentralized lending, trading, and yield farming. Non-fungible tokens (NFTs) gained popularity, enabling the ownership and trading of unique digital assets, including art, collectibles, and virtual real estate. In the same decade, large institutional investors and companies, such as Tesla and Square, began adding Bitcoin to their balance sheets, signaling mainstream acceptance of cryptocurrencies. Cryptocurrency continues to evolve, with ongoing research focused on scalability, security, and usability. The development of central bank digital currencies (CBDCs) and regulatory frameworks remains a focus for governments and central banks. Unlike traditional stock exchanges, cryptocurrency markets operate 24/7 and respond instantly to events, offering both opportunities and challenges. Cryptocurrencies, especially Bitcoin, are highly volatile, with prices subject to influence from Bitcoin brokers. Additionally, storing Bitcoin securely is crucial, as self-custody of assets comes with its own vulnerabilities.

2.3 Deep Learning Predictive Models for Cryptocurrency Prices

Efforts to predict Bitcoin prices have led to the exploration of various predictive models and machine learning techniques. Researchers have employed algorithms such as Long Short-Term Memory (LSTM), Support Vector Regression (SVR), Adaptive Neuro-Fuzzy Inference Systems (ANFIS),Artificial Neural Networks (ANNs), Convolutional Neural Networks(CNNs),Bayesian neural networks (BNNs),Generalized Autoregressive Conditional Heteroskedasticity(GARCH) and Autoregressive Integrated Moving Average (ARIMA).In the following paragraphs below we review some existing literature concerning predictive models for Bitcoin prices.

Yiying and Yeze's (2019)study delved into the dynamic nature of cryptocurrency

prices, specifically focusing on Bitcoin, Ethereum, and Ripple. Their objective was to uncover and comprehend the factors influencing the valuation of these digital assets. Their dataset spanned 1030 trading days, encompassing opening, high, low, and closing prices. Through experimental analysis, they demonstrated the superior effectiveness of LSTM models compared to traditional Artificial Neural Networks (ANNs), indicating LSTM's proficiency in extracting insights from historical data. Furthermore, the authors posited that LSTM's success may stem from its emphasis on short-term dynamics, whereas ANNs tend to rely more on long-term historical trends. However, they noted that with ample historical data, ANNs can attain comparable accuracy to LSTM networks.

McNally et al.(2018) explored Bitcoin price prediction using two deep learning models: a Bayesian-optimized Recurrent Neural Network and an LSTM network. They analyzed data spanning from August 2013 to July 2016, which included information on Bitcoin's open, high, low, and close prices, along with block difficulty and hash rate data. Their performance assessment revealed that the LSTM network outshone the Bayesian-optimized Recurrent Neural Network and even surpassed the accuracy of the traditional statistical approach,AutoRegressive Integrated Moving Average(ARIMA), making it the most effective model for Bitcoin price prediction.

Miura et al.(2019) undertook an analysis of high-frequency Bitcoin data with 1-minute intervals using a combination of machine learning and statistical forecasting models. Given the substantial volume of data, they opted to aggregate realized volatility values over 3-hour intervals. Furthermore, they noted that these values exhibited a weak correlation concerning the extent between high and low prices within these 3-hour intervals.In their rigorous experimental analysis, they explored the effectiveness of several models, including various types of Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), Ridge Regression, and the Heterogeneous Auto-Regressive Realized Volatility model. The outcomes of their research revealed that Ridge Regression consistently outperformed the other models, while SVM exhibited notably poorer performance. These findings emphasize the robustness of Ridge Re-

gression in handling the complexities of Bitcoin's high-frequency data.

Chen et al.(2020a) initited an innovative approach to predicting Bitcoin prices. They incorporated a diverse range of variables, including Bitcoin attributes, public attention metrics from sources like Google Trends and Twitter, as well as economic categorical variables, into their predictive models. One of the key architectural choices was LSTM, a recurrent neural network known for its proficiency in capturing sequential patterns within time-series data.

Jang and Lee's (2018) study sheds light on the growing interest in Bitcoin, which stems from its unique fusion of encryption technology and monetary concepts. This interdisciplinary appeal has drawn attention from the realms of economics, cryptography, and computer science. Their research delves into the Bitcoin process, particularly focusing on the utilization of Bayesian neural networks (BNNs) to analyze its dynamics.To improve the accuracy of Bitcoin price predictions, Jang and Lee meticulously select vital components from the intricate Blockchain data ecosystem, closely intertwined with Bitcoin's supply and demand dynamics. Their empirical study goes further by conducting a thorough comparison of the Bayesian neural network against other benchmark models, both linear and non-linear, in terms of simulating and forecasting Bitcoin price movements.The results of their empirical research showcase the effectiveness of the Bayesian neural network in forecasting Bitcoin price time series and providing insights into the significant price volatility observed in the cryptocurrency market. This research stands out as it not only utilizes advanced neural network techniques but also incorporates relevant Blockchain data, making it a valuable contribution to the field of cryptocurrency price prediction.

Ji et al. (2019) evaluated the prediction performance on Bitcoin price of various deep learning models such as LSTM networks, convolutional neural networks, deep neural networks, deep residual networks and their combinations. The data used in their research, contained 29 features of the Bitcoin blockchain from 2590 days (from 29 November 2011 to 31 December 2018). They conducted a detailed experimental procedure considering both classification and regression problems, where the former

predicts whether or not the next day price will increase or decrease and the latter predicts the next day's Bitcoin price. The numerical experiments illustrated that the deep neural DNN-based models performed the best for price ups-and-downs while the LSTM models slightly outperformed the rest of the models for forecasting Bitcoin's price.

Dyhrberg et al.(2016) employed Generalized Autoregressive Conditional Heteroskedasticity(GARCH) models to assess Bitcoin's potential as a financial asset. Their initial model drew intriguing parallels between Bitcoin, gold, and the U.S. dollar, hinting at its hedging capabilities and potential as a medium of exchange. The results of their asymmetric GARCH analysis suggested that Bitcoin could be a valuable asset for risk-averse investors looking to mitigate exposure to market downturns. In summary, their research indicated that Bitcoin occupies a unique position in financial markets and portfolio management, falling somewhere between gold and the U.S. dollar on a spectrum that spans from pure medium of exchange advantages to pure store of value advantages.

Nakano et al.(2018) examined the performance of ANNs for the prediction of Bitcoin intraday technical trading. The authors focused on identifying the key factors which affect the prediction performance for extracting useful trading signals of Bitcoin from its technical indicators. For this purposed, they conducted a series of experiments utilizing various ANN models with shallow and deep architectures and datasets structures The data utilized in their research regarded Bitcoin time-series return data at 15-min time intervals. Their experiments illustrated that the utilization of multiple technical indicators could possibly prevent the prediction model from overfitting of non-stationary financial data, which enhances trading performance. Moreover, they stated that their proposed methodology attained considerably better performance than the primitive technical trading and buy-and-hold strategies, under realistic assumptions of execution costs.

Nakamoto (2008) introduced a groundbreaking solution to the double-spending problem by proposing a peer-to-peer network. This network achieved security by hash-

ing transactions into an immutable chain of proof-of-work, effectively timestamping transactions and preventing modification without redoing the work. The network's resilience is derived from the longest chain, which not only establishes the order of events but also represents the most significant CPU power source. This decentralized approach allows nodes to freely enter and exit the network, and messages are distributed with best-effort broadcasting, with the longest proof-of-work chain serving as the trusted record of network events, making it a robust and efficient solution.

Liveries et al.(2020) made a significant contribution by integrating three well-established ensemble learning techniques—ensemble averaging, bagging, and stacking—with advanced deep learning models to forecast hourly values of major cryptocurrencies. They utilized contemporary deep learning models such as Long Short-Term Memory (LSTM), bi-directional LSTM, and convolutional layers as building blocks within their ensemble models. These ensemble models underwent rigorous evaluation for their ability to predict cryptocurrency prices for the next hour (regression) and classify price movements relative to the current price. Additionally, their analysis of autocorrelation in prediction errors provided insights into the accuracy and reliability of each forecasting model. This comprehensive research highlights the potential synergy between deep learning and ensemble learning, yielding robust and precise cryptocurrency price forecasting models.

Wu et al. (2018), They suggested a brand-new framework for forecasting the price of bitcoin using two different LSTM models (standard LSTM model and LSTM with AR(2) model). The outcomes supported the suggested model's good forecasting performance using AR(2). For the purpose of predicting the price of bitcoin, the test mean squared error (MSE), root mean square error (RMSE), mean absolute percentage error (MAPE), and mean absolute error (MAE), respectively, are used. Their suggested LSTM with AR(2) model fared better than the LSTM model as a whole.

Kumar and Rath (2020)focused on forecasting the trends of Etherium prices utilizing machine learning and deep learning methodologies. They conducted an experimental analysis and compared the prediction ability of LSTM neural networks and Multi-

Layer perceptron (MLP). They utilized daily, hourly, and minute based data which were collected from the CoinMarket and CoinDesk repositories. Their evaluation results illustrated that LSTM marginally outperformed MLP but not considerably, although their training time was significantly high.

2.4 Future Directions and Research Gaps

While progress has been made in predicting Bitcoin prices using various predictive models, there remain several research gaps and opportunities. As the cryptocurrency market continues to evolve, future research can focus on:

- Enhancing the accuracy and interpretability of predictive models, especially LSTM-based models.
- Exploring the impact of external factors such as macroeconomic events, regulatory changes, and technological advancements on Bitcoin price prediction.
- Investigating the relevance and effectiveness of different machine learning techniques beyond LSTM, including Convolutional Neural Networks (CNNs), Gated Recurrent Units (GRUs), and Deep Neural Networks (DNNs).
- Assessing the practical implications of accurate Bitcoin price predictions on cryptocurrency trading strategies, investor behavior, and market stability.
- Addressing the challenges posed by the unique characteristics of cryptocurrencies, such as decentralization and lack of regulation.

In conclusion, the field of Bitcoin price prediction is dynamic and offers numerous opportunities for further research and innovation. As the cryptocurrency market matures, predictive models and analytical techniques will play a crucial role in assisting traders, investors, and policymakers in navigating this evolving financial landscape. The literature presented here forms the basis for the development and evaluation of our proposed LSTM model for Bitcoin price prediction.

Chapter 3

METHODOLOGY

3.1 INTRODUCTION

The aim of this research paper is to predict the future price of Bitcoin using the Recurrent Neural Network (RNN) with Long-Short Term Memory (LSTM). This study utilizes time series data, specifically the price history of Bitcoin, to forecast and model future prices through a deep learning approach. The quantitative data required for this research was sourced from Yahoo Finance. This dataset covers the price fluctuations of Bitcoin from January 2018 to March 2023. Our main goal is to improve cryptocurrency price predictions and gain insights into market dynamics. Through statistical analysis, we aim to reduce prediction errors, ultimately benefiting traders and investors with more accurate Bitcoin price forecasts. Our objectives include applying LSTM models, introducing randomized activation functions, and using them to predict Bitcoin prices. The research outcomes have the potential to enhance decision-making in the cryptocurrency market.

3.2 AN OVERVIEW OF NEURAL NETWORKS

Neural networks are among the most widely used machine learning algorithms, renowned for their superior performance in terms of both accuracy and speed when compared to other algorithms.

3.2.1 GENERAL UNDERSTANDING OF NEURAL NETWORKS

A Neural Network is a key concept in artificial intelligence, designed to enable computers to process data in a manner inspired by the human brain. It operates through deep learning, utilizing interconnected nodes or neurons organized in layers, akin to the human brain's structure.

Neural Networks comprise various layers interlinked, collaborating to define a system's structure and function. These networks learn from extensive datasets and employ intricate algorithms for training. The fundamental layers encompass an input layer, one or more hidden layers, and an output layer. Information flows sequentially, starting with the input layer, passing through hidden layers, and ultimately generating an output. Figure 3.1 shows a simple neural network.

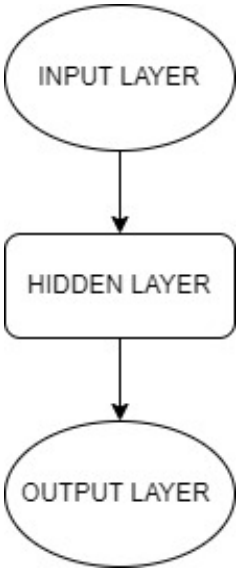

Figure 3.1: A Simple Neural Network

3.2.2 TYPES OF NEURAL NETWORK

Several neural networks can help solve different problems. Some include: **Feed Forward Network:**An artificial neural network with a circular network of nodes is called a feed forward neural network. In contrast to recurrent neural networks, feed forward neural networks cycle some routes. The feed forward model is a basic type of neural

network because the input is only processed in one direction.It is the traditional neural network use for general regression and classification. Figure 3.2 shows the various stages in the feed forward neural network.

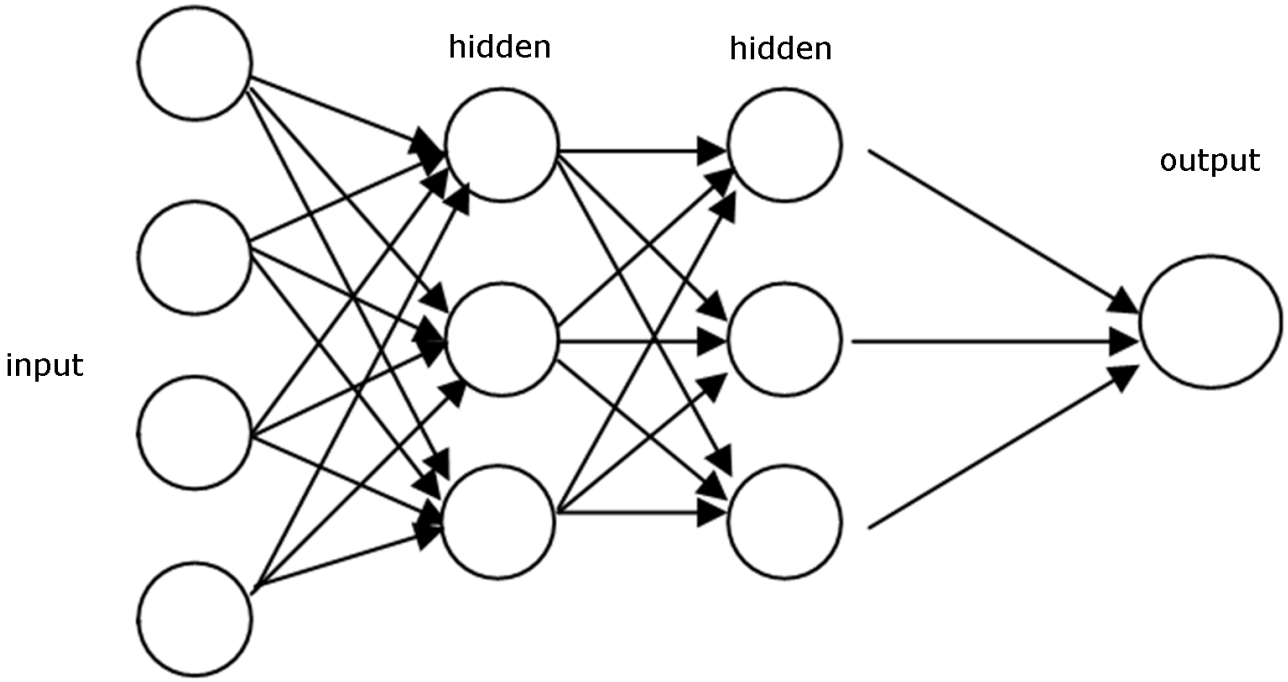

Figure 3.2: Feed Forward Neural Network

Convolutional Neural Network: In deep learning, a convolutional neural network (CNN) is a class of artificial neural network most commonly applied to analyze visual imagery. CNNs use a mathematical operation called convolution in place of general matrix multiplication in at least one of their layers. They are specifically designed to process pixel data and are used in image recognition and processing. They have applications in image and video recognition, recommender systems,image classification, image segmentation, medical image analysis, natural language processing,brain–computer interfaces, and financial time series. CNNs are also known as Shift Invariant or Space Invariant Artificial Neural Networks (SIANN), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation-equivariant responses known as feature maps. Counter-intuitively, most convolutional neural networks are not invariant to translation, due to the downsampling operation they apply to the input.

Mathematical formula for the CNN.

Convolution with Bias and Activation:

$$(I * K)(x, y) = \sigma \left(\sum_{i=0}^{n} \sum_{j=0}^{m} (I(x+i, y+j) \cdot K(i, j) + b) \right)$$

where,

σ = Activation function (tanh, sigmoid).

b = Bias term added to the convolution result.

The figure below represents the convolutional neural network.

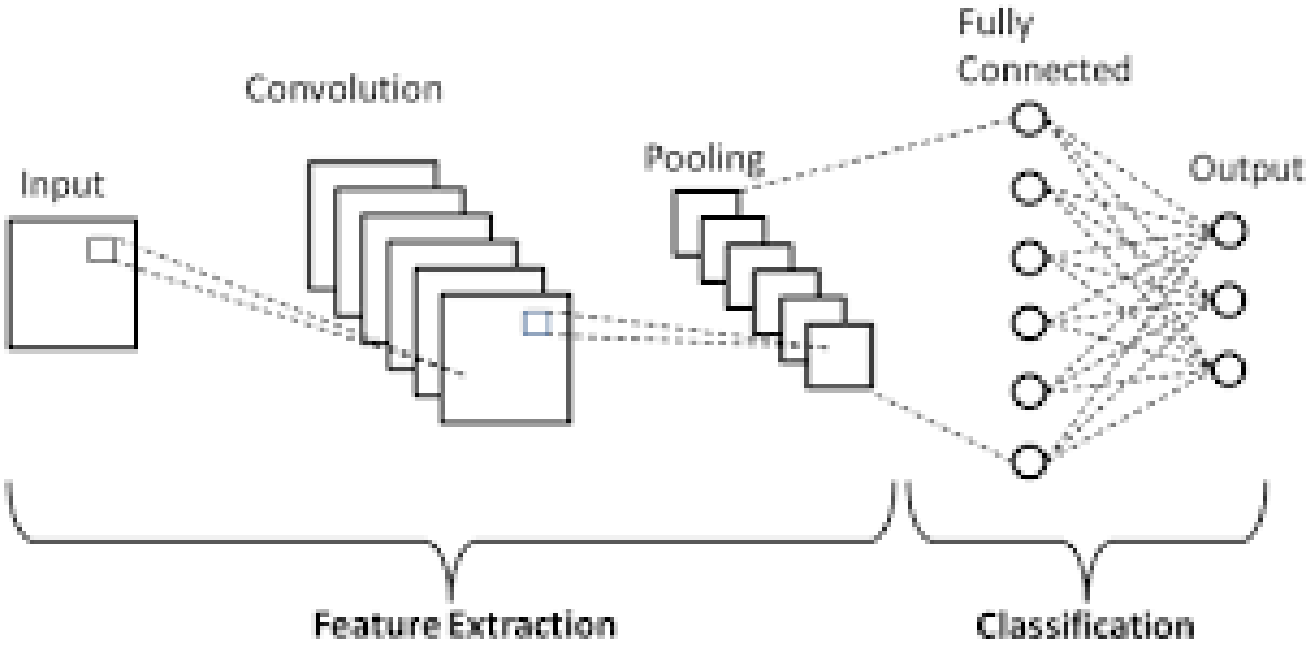

Figure 3.3: Convolutional Neural Network

Recurrent Neural Network (RNN) Used for speech recognition, voice recognition, time series prediction, and natural language processing.The traditional neural network that is, Feed forward neural networks is not designed for sequences (time series data) hence the results with time series is bad. It does not have a memory model which makes them unsuitable for time series analysis. RNN captures information of the sequences of the time series data. They can take variable size inputs and give us variable size output and they work really well with time series data.
RNN works on the principle of saving the output of a particular layer and feeding this back to the input to predict the output of the layer.

3.3 RECURRENT NEURAL NETWORK

A recurrent neural network (RNN) is a type of artificial neural network which uses sequential data or time series data. These deep learning algorithms are commonly used for ordinal or temporal problems, such as language translation, natural language processing (nlp), speech recognition, and image captioning; they are incorporated into popular applications such as Siri, voice search, and Google Translate. Like feedforward and convolutional neural networks (CNNs), recurrent neural networks

utilize training data to learn. They are distinguished by their "memory" as they take information from prior inputs to influence the current input and output. While traditional deep neural networks assume that inputs and outputs are independent of each other, the output of recurrent neural networks depend on the prior elements within the sequence.

3.3.1 WHY RECURRENT NEURAL NETWORK?

RNN were created because there were a few issues in the feed-forward neural network:

- Cannot handle sequential data.
- Considers only the current input.
- Cannot memorize previous inputs

The Feed forward neural network (FNN) does not use previous information to affect later ones.The solution to these issues is the RNN. An RNN can handle sequential data, accepting the current input data, and previously receive inputs. RNNs can memorize previous inputs due to their internal memory (sequential memory).
An RNN has a loop mechanism that can pass previous information forward. The loop mechanism allows information to flow from one step to the next. This information is known as the hidden state which represents previous input passed through the activation function.
Fiure 3.4 shows the various layers in a recurrent neural network.

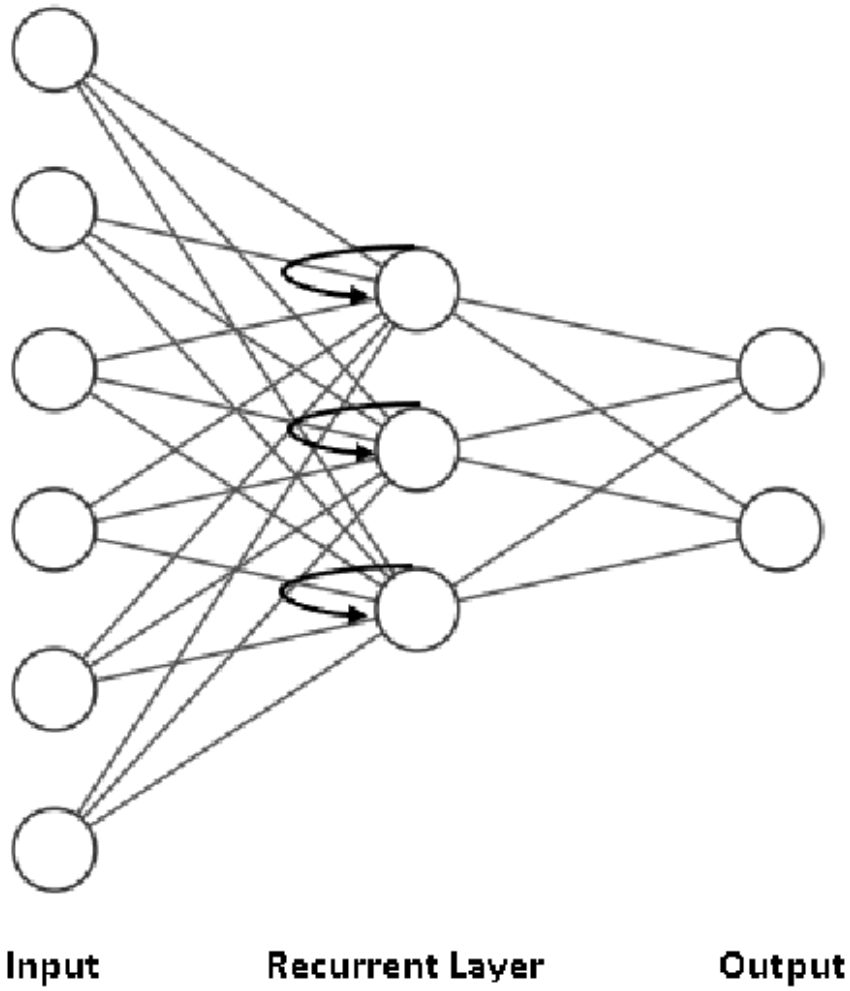

Figure 3.4: RNN Model

TYPES OF RNN MODEL

There are four types of Recurrent Neural Networks:

1. One to One
2. One to Many
3. Many to One
4. Many to Many

One to One RNN (1:1)

This type of neural network is known as the Vanilla Neural Network. It's used for general machine learning problems, which has a single input and a single output.

One to Many RNN (1:N)

This type of neural network has a single input and multiple outputs. An example of this is the image caption.

$$\text{Input Sequence: } x = (x_1, x_2, \ldots, x_T) \tag{3.3.1}$$

$$\text{Output Sequence: } y = (y_1, y_2, \ldots, y_N) \tag{3.3.2}$$

$$\text{Description: } y_t = f(x) \text{ for each } t \tag{3.3.3}$$

$$\text{where } f \text{ can be any function.} \tag{3.3.4}$$

Many to One RNN (N:1)

This RNN takes a sequence of inputs and generates output. Sentiment analysis is a good example of this kind of network where a given sentence can be classified as expressing positive or negative sentiments. Another example is the **time series prediction.**

$$\text{Input Sequence: } x = (x_1, x_2, \ldots, x_T) \tag{3.3.5}$$

$$\text{Output: } y \tag{3.3.6}$$

$$\text{Description: } y = f(x), \text{ where } f \text{ processes the entire input sequence to produce a single output.} \tag{3.3.7}$$

The figure below represents a many to one RNN model.

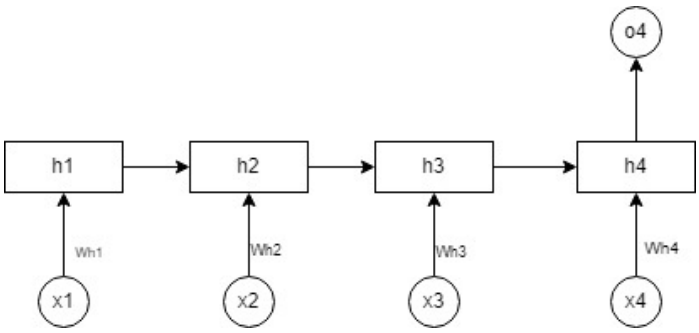

Figure 3.5: Many to one RNN

Many to Many RNN (N:N)

This RNN takes a sequence of inputs and generates a sequence of outputs, making it suitable for tasks such as machine translation.

$$\text{Input Sequence: } x = (x_1, x_2, \ldots, x_T) \tag{3.3.8}$$

$$\text{Output Sequence: } y = (y_1, y_2, \ldots, y_N) \tag{3.3.9}$$

$$\text{Description: } y_t = f(x_t), \text{ for each } t \tag{3.3.10}$$

Figure 3.6: Many to Many RNN Diagram

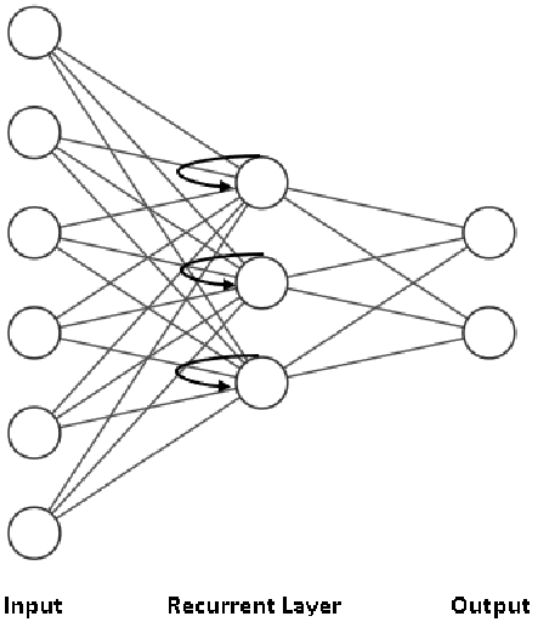

Figure 3.7: RNN Model

3.3.2 ACTIVATION FUNCTION IN RNN

In a recurrent neural network (RNN), the activation function plays a crucial role in determining the output of the hidden state and, consequently, the overall output of the network. The activation function of an RNN takes the weighted input of the current time step, the previous hidden state, and possibly an input from the previous time step, and applies a nonlinear transformation to produce the hidden state of the current time step. The choice of activation function can significantly impact the performance of the RNN model. Some commonly used activation functions in RNNs include the sigmoid function, hyperbolic tangent function, and Rectified Linear Unit (ReLU) function. The sigmoid function is often used as it produces outputs in the range of $[0, 1]$, which can be interpreted as probabilities.

Proof of Sigmoid Function Range

As $x \to \infty$ and $x \to 0$:

Upper Bound:

As $x \to \infty$, e^{-x} approaches 0. Therefore, the denominator of the sigmoid function approaches 1, and the numerator remains as 1. Hence, as $x \to \infty$, the sigmoid function approaches 1:

$$\lim_{x \to \infty} \sigma(x) = \frac{1}{1+0} = 1 \tag{3.3.11}$$

This shows that the sigmoid function has an upper bound of 1.

Lower Bound:

As x approaches $-\infty$, e^{-x} approaches infinity. Therefore, the denominator of the sigmoid function approaches infinity, and the numerator remains as 1. Hence, as x goes to negative infinity, the sigmoid function approaches 0:

$$\lim_{x \to -\infty} \sigma(x) = \frac{1}{1+\infty} = 0 \tag{3.3.12}$$

This shows that the sigmoid function has a lower bound of 0. Combining the upper and lower bounds, we can conclude that the sigmoid function has a range of $(0, 1)$.

A typical sigmoid activation function used in a Recurrent Neural Network (RNN) is given as equation 3.4.13.

$$\sigma(x) = \left(\frac{1}{1+e^{-x}}\right) \tag{3.3.13}$$

In particular, we choose a uniforrm distribution for beta [$\beta \sim U(0,1)$] to make β a learnable parmater to modify the shape of the sigmoid function is given as equation 3.4.14.

$$\sigma(x) = \left(\frac{1}{1+e^{-\beta x}}\right) \tag{3.3.14}$$

Since we want to optimize the parameter beta β, we choose to define a custom activation function.

In addition, the RNN has two extra activation functions as follows: **The hyperbolic tangent function**is also a popular choice as it produces outputs in the range of $[-1, 1]$, which can help with the stability of the network during training. **The ReLU function** is a simpler and computationally efficient activation function that is commonly used in other types of neural networks, but it can also be used in RNNs.

3.3.3 UNDERSTANDING LAYERS OF RNN

Recurrent Neural Networks (RNNs) prove highly effective in time series prediction tasks due to their ability to accommodate sequences of varying lengths and capture temporal dependencies between data points. To gain insight into the utilization of RNNs for time series prediction, consider the following general overview: **Input layer**: The first layer of an RNN takes input vector which represents the elements of a sequence.

For example, in prediction, each input vector might represent a data point(price), **x(t)**.

$$i(t) = (w_i * h(t-1) + w_i * x(t) + b) \tag{3.3.15}$$

The candidate value ($\tilde{C}_t$) is computed using the hyperbolic tangent (tanh) activation function to ensure values are within -1 and 1.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3.3.16}$$

Cell State Update The cell state is updated by combining the scaled previous cell state and the scaled candidate value from the input gate.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{3.3.17}$$

The output gate computes the new hidden state (h_t) using the sigmoid activation function and the tanh activation function.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{3.3.18}$$

The output hidden state h_t based on the updated cell state is obtained by multiplying equation 3.3.18 by equation.3.3.17

$$h_t = o_t \cdot \tanh(C_t) = y_t \tag{3.3.19}$$

In this thesis, we denote w_h as the weight matrix responsible for mapping the input vector $x(t)$ to the hidden state vector $h(t)$.

Where:

- wh is the weight matrix responsible for mapping the input vector $x(t)$ to the hidden state vector $h(t)$.
- wh is the weight matrix for the hidden state.
- $x(t)$ is the input vector at time t.
- $h(t)$ is the hidden state vector at time t.
- wi is determined by the input size.
- wi represents a weight matrix.
- wi is determined by the input size.

Where o is output gate,h is hidden state, σ is Activation function, W and U are weight matrix, and t is time. The structure of RNN nodes can be described as shown in Figure 3.11

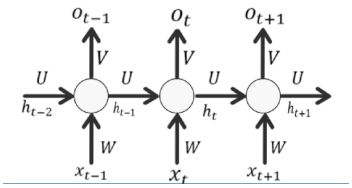

Figure 3.8: Structure of nodes in RNN

W=Weight=$\begin{bmatrix} W_i \\ W_o \end{bmatrix}$, U=Weight=$\begin{bmatrix} U_i \\ U_o \end{bmatrix}$, V=Weight=$\begin{bmatrix} V_i \\ V_o \end{bmatrix}$, b=bias=$\begin{bmatrix} b_i \\ b_o \end{bmatrix}$, h=$\begin{bmatrix} W_{i_h} \\ W_{o_h} \end{bmatrix}$, and g=$\begin{bmatrix} i_t \\ o_t \end{bmatrix}$.

3.3.4 BIAS TERM IN RNN

The bias term in RNN is associated with each unit (neuron) within the RNN cell. It's added to the weighted sum of inputs to introduce a shift or offset to the activations of the units. In RNN, the bias term is typically added to the weighted sum of the input and recurrent connections in each cell of the network.

The formula for the bias term in an RNN cell can be written as follows:

$$b = W_b * x + U_b * h \tag{3.3.20}$$

where:

b is the bias vector

W_b is the weight matrix for the bias term in the input layer

x is the input vector

U_b is the weight matrix for the bias term in the recurrent layer

h is the output of the previous cell in the sequence (i.e., the previous hidden state)

The bias term allows the RNN to model data with different means and to adjust the output of each cell based on a constant value.

3.4 LONG SHORT-TERM MEMORY (LSTM)

LSTM is a specialized type of recurrent neural network architecture designed to address the limitations of traditional RNNs, particularly the vanishing gradient problem. Vanishing gradients occur when the gradient signal that is backpropagated through the network during training becomes too small to update the weights effectively, leading to slow or stalled learning.

LSTMs use a more complex architecture than traditional RNNs, incorporating a set of memory cells and gates that control the flow of information through the network. These gates allow the LSTM to selectively remember or forget information from previous time steps, while the memory cells enable it to store and retrieve information for longer periods of time.

LSTMs have several advantages over traditional RNNs, including:

1. The ability to control long-term dependencies in the input sequence
2. Reduced risk of vanishing gradients, enabling more stable and efficient training
3. Flexibility in the types of input and output data that can be used
4. The ability to learn more complex temporal patterns in the input data.

   These features make LSTMs well-suited for a wide range of applications, including speech recognition, natural language processing, and **time series analysis**

The figure below represents the scematic diagram of RNN and LSTM.

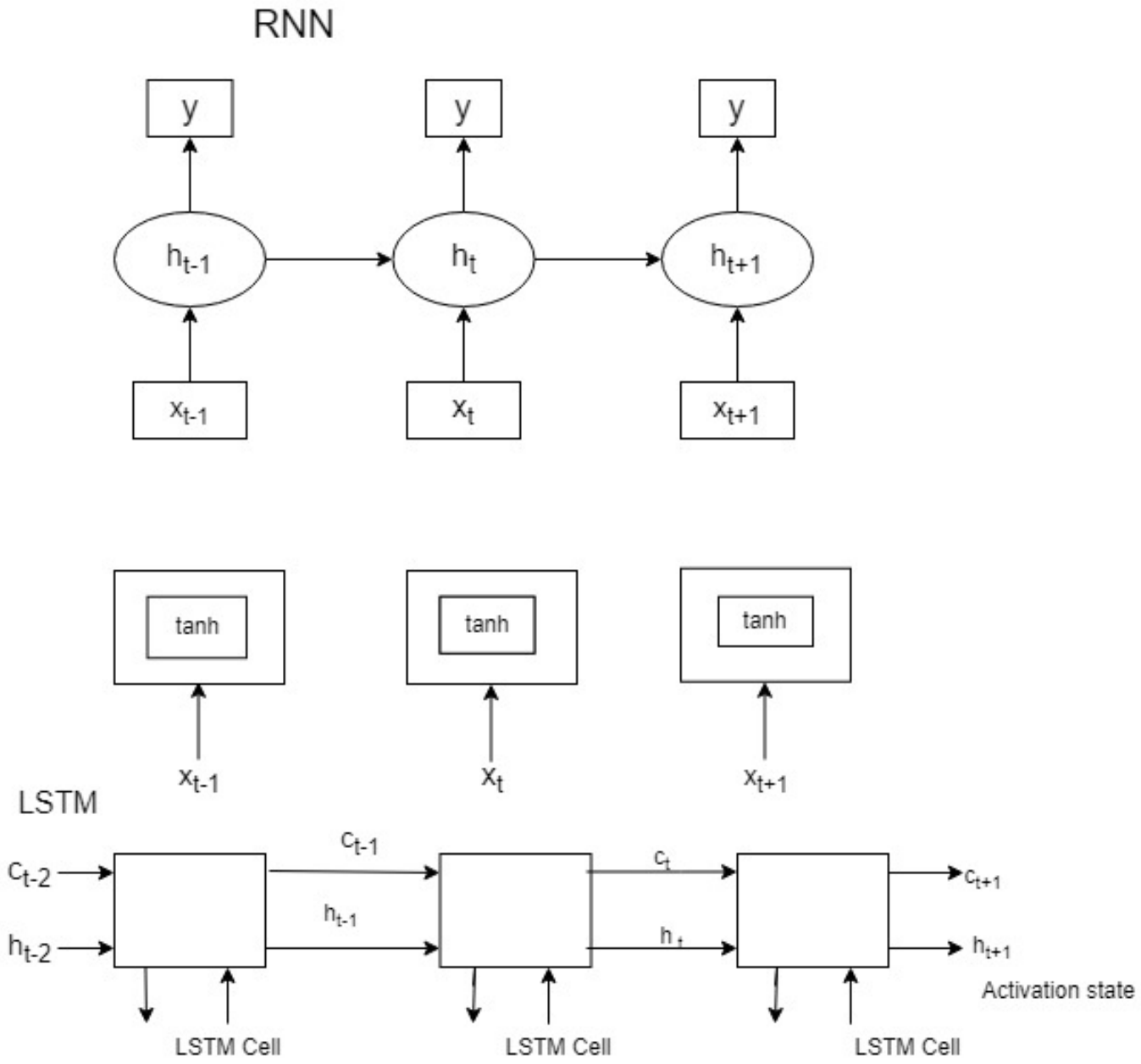

Figure 3.9: RNN and LSTM

3.4.1 THEORY OF LONG-SHORT TERM MEMORY

Long-Short Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to address the vanishing gradient problem in traditional RNNs.Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. LSTM networks are particularly well-suited for tasks involving sequential data, such as time series analysis, natural language processing, and speech recognition. The theory behind LSTM revolves around its unique cell structure and gating mechanisms that allow it to capture long-term dependencies in sequences.

The key theoretical components of LSTM:

- **Cell State:** The central concept of LSTM, serving as a "memory" of past information. This state can be modified, read from, and written to, allowing the network to selectively remember or forget information over time, enhancing prediction accuracy.

- **Three Gates:** LSTM introduces three gates to control the flow of information within the cell: the Forget Gate, Input Gate, and Output Gate.

- **Gating Mechanisms:** Each gate is controlled by a sigmoid activation function, outputting values between 0 and 1. These values scale the information flowing through the gates, deciding what to keep (values close to 1) and what to discard (values close to 0).

- **Cell State Update:** The cell state is updated through a combination of the Forget Gate, Input Gate, and an intermediate update calculation. This results in a refined cell state that retains important information and discards irrelevant details.

In the context of our Bitcoin price prediction, LSTM can be employed to analyze historical price data, learn patterns and dependencies, and make forecasts based on the captured information. The network's ability to retain relevant information from the past while discarding noise makes it a powerful tool for time series forecasting.

3.4.2 ACTIVATION FUNCTION IN LSTM

In Long Short-Term Memory (LSTM) networks, the activation functions are fundamental in controlling information flow and memory retention within the network. The primary activation functions in an LSTM cell are the sigmoid and hyperbolic tangent (tanh) functions. These functions play essential roles in the gating mechanisms of the LSTM, allowing it to capture long-range dependencies in sequential data.

3.4.3 Hyperbolic Tangent (tanh) Activation Function

The hyperbolic tangent function, denoted as $\tanh(x)$, is another essential activation function in LSTM networks. It transforms its input to a value between -1 and 1. The tanh function is defined as:

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{3.4.1}$$

In LSTMs, the tanh function is primarily used to compute the new candidate cell state ($\tilde{C}_t$) and to determine which values will be updated in the current cell state. The tanh activation function is valuable because it allows the LSTM to capture both positive and negative information, making it effective for learning complex patterns in sequential data.These activation functions, in conjunction with the LSTM architecture's various gates and memory cells, enable LSTMs to excel in tasks that involve sequential data, such as natural language processing, speech recognition, and time series forecasting.

3.4.4 Standard Sigmoid Activation Function

The sigmoid activation function, denoted as $\sigma(x)$, is a key component of the LSTM cell's gating mechanism. It takes an input x and transforms it to a value between 0 and 1. The sigmoid function is expressed as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.4.2}$$

3.4.5 Randomisation of Beta in Sigmoid Activation Function

LSTM gates with varying parameter beta, the mathematical formula for the sigmoid function with beta varying in the range (0, 1) is as follows:

$$\sigma(x) = \left(\frac{1}{1 + e^{-\beta X}}\right) \tag{3.4.3}$$

In particular we choose a uniform distribution for β [$\beta \sim U(0, 1)$] to modify the shape of the sigmoid function in the equation above, the optimal beta is a constant value sampled from a uniform distribution in the range (0, 1). This optimal beta value can be different for each gate (input gate, forget gate, and output gate) in the LSTM cell.For values of beta between 0 and 1, the sigmoid function will be scaled, Smaller values of beta (closer to 0) will result in weaker gating, allowing more information to pass through, while larger values of beta (closer to 1) will lead to stronger gating, retaining less information. Figure 3.9 below shows the architecture of the LSTM model.

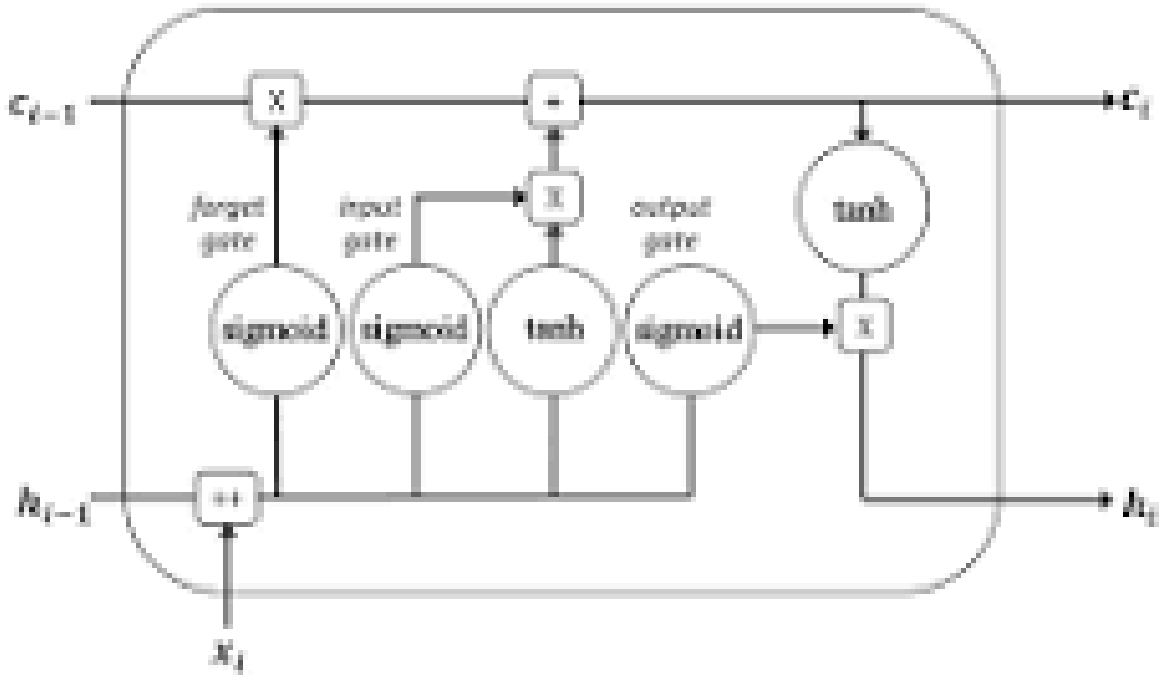

Figure 3.10: LSTM Architecture

3.4.6 LSTM Equations with randomised Beta

The LSTM Gates equations :

$$\mathbf{i}_t = \sigma_\beta(\mathbf{W}_{ix}\mathbf{x}_t + \mathbf{W}_{ih}\mathbf{h}_{t-1} + \mathbf{b}_i) \tag{3.4.4}$$

$$\mathbf{f}_t = \sigma_\beta(\mathbf{W}_{fx}\mathbf{x}_t + \mathbf{W}_{fh}\mathbf{h}_{t-1} + \mathbf{b}_f) \tag{3.4.5}$$

$$\mathbf{o}_t = \sigma_\beta(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{W}_{oh}\mathbf{h}_{t-1} + \mathbf{b}_o) \tag{3.4.6}$$

Substituting into the Sigmoid Activation Function in the LSTM is given as below.

$$\mathbf{i}_t = \left(\frac{1}{1 + e^{\beta(\mathbf{W}_{ix}\mathbf{x}_t + \mathbf{W}_{ih}\mathbf{h}_{t-1} + \mathbf{b}_i)}}\right) \tag{3.4.7}$$

$$\mathbf{f}_t = \left(\frac{1}{1 + e^{\beta(\mathbf{W}_{fx}\mathbf{x}_t + \mathbf{W}_{fh}\mathbf{h}_{t-1} + \mathbf{b}_f)}}\right) \tag{3.4.8}$$

$$\mathbf{o}_t = \left(\frac{1}{1 + e^{\beta(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{W}_{oh}\mathbf{h}_{t-1} + \mathbf{b}_o)}}\right) \tag{3.4.9}$$

The remaining LSTM cell equations remain unchanged, providing control over the sigmoid activation function's shape.

noindent W=Weight= $\begin{bmatrix} W_f \\ W_i \\ W_g \\ W_o \end{bmatrix}$ b=bias= $\begin{bmatrix} b_f \\ b_i \\ b_g \\ b_o \end{bmatrix}$, h= $\begin{bmatrix} W_{f_h} \\ W_{i_h} \\ W_{g_h} \\ W_{o_h} \end{bmatrix}$, and g= $\begin{bmatrix} f_t \\ i_t \\ g_t \\ o_t \end{bmatrix}$

Where i is input gate, f is forget gate, o is output gate, c is cell state, h is hidden state, σ is Activation function, W is weight matrix assigned, g is various gates, and t is time.

3.4.7 Bias Term in LSTM

The bias terms in the LSTM are essential components that help control the flow of information through the LSTM cell. These bias terms are incorporated into the larger equations governing LSTM cell operations. For formula for the bias term for the input gate may be expressed as:

$$b_i = \sigma(W_{bi} \cdot h_{t-1} + U_{bi} \cdot x_t)$$

Where:

b_i is the bias term for the input gate.

W_{bi} and U_{bi} are weight matrices associated with the input gate.

h_{t-1} is the previous hidden state.

x_t is the input at time step t.

3.4.8 HOW THE LSTM WORKS

At its core, an LSTM model consists of a cell, which contains a hidden state and a memory unit, and three gates: input gate, forget gate, and output gate.

Input Sequence: LSTM takes an input sequence, which could be a sequence of words in a sentence, time series data, or any other sequential data.

Cell State: It maintains a cell state, which serves as a memory unit that can store and retrieve information over long sequences. The cell state runs horizontally through the entire network.

Forget Gate: The forget gate uses a sigmoid activation function to decide what information from the previous cell state should be retained and what should be discarded. It helps in selectively removing less relevant information from the cell state.

$$f(t) = \sigma(W_{f_h} \cdot h(t-1) + W_{f_x} \cdot x(t) + b_f) \tag{3.4.10}$$

where $x = W_{f_h} \cdot h(t-1) + W_{f_x} \cdot x(t) + b_f$

Substitute x into the sigmoid to get equation 3.4.11

$$f(t) = \left(\frac{1}{1 + e^{-\beta(W_{f_h} \cdot h(t-1) + W_{f_x} \cdot x(t) + b_f)}} \right) \tag{3.4.11}$$

Input gate: regulates the information allowed to enter the current cell state. It takes input from the current time step and the previous hidden state and produces a vector of values between 0 and 1 for each element of the input vector. These values determine which information is added to the cell state.

$$i(t) = \sigma(W_i \cdot h(t-1) + W_i \cdot x(t) + b_i) \tag{3.4.12}$$

where $x = W_{i_h} \cdot h(t-1) + W_{i_x} \cdot x(t) + b_i$

Substitute x into the sigmoid to get equation 3.4.13

$$i(t) = \left(\frac{1}{1 + e^{-\beta(W_i \cdot h(t-1) + W_i \cdot x(t) + b_i)}} \right) \tag{3.4.13}$$

Output Gate: The output gate controls the information that is outputted from the cell. It takes the input at the current time step, the previous hidden state, and the current cell state as input and produces a vector of values between 0 and 1 for each element of the current hidden state.

$$o(t) = \sigma(W_{o_h} * h(t-1) + W_{o_h} * x(t) + b_o) \tag{3.4.14}$$

where $x = W_{o_h} \cdot h(t-1) + W_{o_x} \cdot x(t) + b_o$

Substitute x into the sigmoid to get equation 3.4.15 as follow,

$$o(t) = \left(\frac{1}{1 + e^{-\beta(W_{o_h} * h(t-1) + W_{o_h} * x(t) + b_o)}} \right) \tag{3.4.15}$$

The LSTM model works by processing sequential input data one time step at a time. At each time step, the input is fed through the input gate, and the previous hidden state and memory state are passed through the forget gate. The resulting information is combined in the cell to create a new memory state. This new memory state is then passed through the output gate to create the new hidden state, which is the output of the LSTM model for that time step.

The output hidden state h_t (equation 3.4.16) based on the updated cell state is obtained by multiplying the output gate (equation 3.4.14) with the tanh function substituted with the cell state C_t(equation 3.3.16).

$$h_t = o_t \cdot \tanh(C_t) = y_t \tag{3.4.16}$$

Substituting equation 3.4.14 into equation 3.4.16 you obtain equation 3.4.17.

$$h_t = \left(\frac{1}{1 + e^{-\beta(W_{o_h} \cdot h(t-1) + W_{ox} \cdot x(t) + b_o)}} \right) \cdot \tanh(C_t) = y_t \tag{3.4.17}$$

where the hidden state for the output ht is the output y_t

The key advantage of LSTMs is their ability to capture and retain important information over long sequences while selectively forgetting less relevant details, making them effective for a wide range of sequential data tasks.

The memory state of an LSTM allows it to store information for longer periods of time, making it useful for processing long sequences of data. The input, forget, and output gates of an LSTM allow it to selectively remember or forget information, mak-

ing it more powerful than traditional RNNs. W=Weight= $\begin{bmatrix} W_f \\ W_i \\ W_g \\ W_o \end{bmatrix}$, U=Weight= $\begin{bmatrix} U_f \\ U_i \\ U_g \\ U_o \end{bmatrix}$,
V=Weight= $\begin{bmatrix} V_f \\ V_i \\ V_g \\ V_o \end{bmatrix}$, b=bias= $\begin{bmatrix} b_f \\ b_i \\ b_g \\ b_o \end{bmatrix}$, h= $\begin{bmatrix} W_{f_h} \\ W_{i_h} \\ W_{g_h} \\ W_{o_h} \end{bmatrix}$, and g= $\begin{bmatrix} f_t \\ i_t \\ g_t \\ o_t \end{bmatrix}$

Where i is input gate, f is forget gate, o is output gate, c is cell state, h is hidden state, σ is Activation function, W and U are weight matrix, and t is time. The structure of LSTM nodes can be described as shown in Fig The structure of LSTM nodes can be described as shown in Figure below

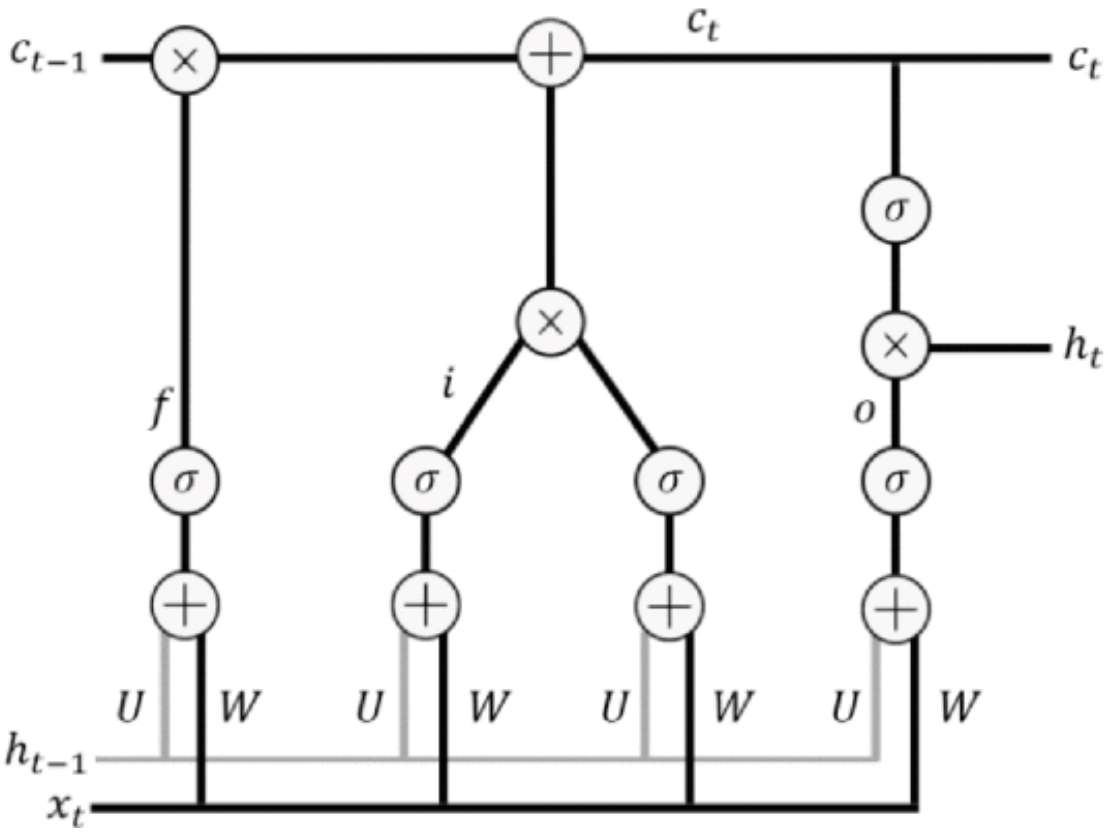

Figure 3.11: Structure of nodes in LSTM and How it works

3.5 Model Evaluation Accurency

To select the best approach for Bitcoin price prediction, we evaluated the performance of both the baseline LSTM model and the LSTM model with the randomized activation function (β). Evaluation metrics will include:

- Mean Absolute Error (MAE)
- Accuracy

The formula for MAE is given by:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{Y}_i - Y_i|$$

Where:

n is the number of data points.

$\hat{Y}_i$ represents the predicted Bitcoin price.

Y_i represents the actual Bitcoin price.

Accuracy was calculated based on a predefined threshold for price direction prediction.

3.5.1 Replication and Average optimal Beta, β Calculation

To further assess model performance and achieve an average optimal β value of 0.5, we will conduct simulations as follows:

- Replicate,10,20,50,and 100 iterations of Bitcoin price predictions using the LSTM model with randomized β values.
- Calculate the average β value across these iterations using the formula:

$$\text{Average } \beta = \frac{1}{N}\sum_{i=1}^{N}\beta_i$$

Where:

N is the number of iterations.

β_i represents the β value for the i-th iteration.

This methodology outlines the steps for model development, evaluation, and simulations necessary to select the best approach for Bitcoin price prediction while maintaining an average β of 0.5.

3.5.2 Replication and Average Forecast Values Calculation

In order to comprehensively evaluate model performance and determine the average forecast values for various sigmoid activation functions and the standard LSTM, we will conduct simulations as outlined below:

- We simulated Bitcoin price predictions using the LSTM model for different scenarios, specifically 5,10,20,50 and 100 iterations. Each iteration will involve uniform distribution of β values.$\beta \sim U(0, 1)$

The Average Forecast Value (AFV) for each scenario can be calculated using the following formula:

$$AFV = \frac{1}{n} \sum_{i=1}^{n} P_i \tag{3.5.1}$$

Where:

AFV = Average Forecast Value for the given scenario.

n = Total number of iterations (5,10,20,50,and 100).

P_i = Predicted Bitcoin price for the i-th iteration.

This formula allows us to determine the average forecasted Bitcoin price for each scenario, providing valuable insights into the model's price future price.

Chapter 4

DATA ANALYSIS

4.1 INTRODUCTION

In this project we perform data analysis and use Long Short-Term Memory (LSTM) with varying beta for predicting Bitcoin. We will explore the impact of different beta values on the LSTM model's predictive performance and identify the optimal beta for the best predictions.

This study aims to make a short run prediction for Bitcoin from January 2018 to March 2023 for the close price of the day of the extracted data.The data for this study is downloaded from the online repository Yahoo Finance. Before using information for the training process,it is vital to take a pre-processing step. We employ data cleaning, which is the process of detecting and correcting inaccurate or irrelevant parts of the data and then replacing, modifying or deleting the dirty data.Indeed, as an important point,to prevent the effect of the large value of an indicator on the smaller ones the data is normalized. Data normalization refers to rescaling actual numeric features into a 0 or 1 range and is employed in machine learning to create attaining model less sensitive to the scale of variable.This data was analyzed using the Statistical softwares python and jupyter notebook ,employing the approach of Machine Learning using the RNN with LSTM as discussed in the Methodology.

4.2 EXPLORATORY DATA ANALYSIS

Exploratory data analysis provides valuable insights into the key statistics of the dataset. In our analysis, we focus on Bitcoin data spanning from January 2018 to March 2023.

The table contains the mean, standard deviation(std),minimum value,maximum and percentiles for the various variables of a bitcoin price data. The mean values (mean of close approximately 20,444.65) provide insights into the central tendency of Bitcoin prices during the specified period. The mean figures help to understand the typical price levels. It is essential to consider the standard deviation (e.g., Std Close of approximately 16,610.89) to gauge the extent of price variability. Higher standard deviations indicate greater price volatility, which is a crucial aspect to analyze in this context. Minimum and maximum values showcase the range of price fluctuations, emphasizing Bitcoin's potential for substantial gains and losses. Percentiles like the 25th, 50th, and 75th percentiles help understand price distribution and trends.The trading volume reveals liquidity and market activity levels, essential for analyzing market efficiency and sentiment.

Table 4.1: Basic Statistics for Bitcoin Prices

	Open	High	Low	Close	Adj Close	Volume
Mean	20444.65	20944.28	19888.05	20449.75	20449.75	2.66e+10
Std	16614.98	17049.43	16112.56	16610.89	16610.89	1.95e+10
Min	3236.27	3275.38	3191.30	3236.76	3236.76	2.92e+09
25%	7915.75	8135.36	7696.53	7912.79	7912.79	1.33e+10
50%	11480.90	11786.16	11219.94	11485.86	11485.86	2.44e+10
75%	32136.51	33259.25	30943.07	32123.12	32123.12	3.54e+10
Max	67549.73	68789.63	66382.06	67566.83	67566.83	3.51e+11

sectionDATA PREPARATION TThe data is put together and randomized. This helps make sure that data is evenly distributed, and the ordering does not affect the learning process. The data is cleaned to remove unwanted data, missing values, rows, and columns, duplicate values, data type conversion, etc. The data is visualized to understand how it is structured and understand the relationship between various variables and prices present. The data is split into two sets - a training set and a testing set. The training is when the model the model learns the sequences and

patterns of the dataset. The testing is used to check the accuracy of your model after training.

4.3 MODEL TRAINING

The training phase is crucial in machine learning, especially for models like Recurrent Neural Networks (RNN) with Long-Short Term Memory (LSTM).80% of the dataset is used. During this phase, historical Bitcoin data is provided to the model, allowing it to learn patterns and improve predictions. As the model processes data over time, it gets better at forecasting. In this case, activation functions is used and experiment with beta values for optimization. The data then goes through the RNN with LSTM, undergoing transformations.The goal is to fine-tune the model to predict Bitcoin prices accurately.Through training, we aim to enhance its forecasting abilities.

4.4 MODEL TESTING AND EVALUATION

After training, the model's performance is evaluated using a 20% of the dataset. Testing with new data is crucial because using the same data as training would lead to inflated accuracy. Testing on unseen data provides an accurate measure of the model's performance and computational speed for real-time applications. The model's ability to predict the Bitcoin Close Price is assesed, a key metric for evaluating its effectiveness in forecasting cryptocurrency prices.In short, testing and evaluation validate the practicality and reliability of our trained model in real-world scenarios.

4.5 PARAMETER TUNING

In the "Parameter Tuning" section, optimizing the RNN with LSTM model to enhance its accuracy in predicting Bitcoin prices is the focus. Parameters are key variables that influence the model's performance, and finding the best values for these parameters is crucial.Parameter tuning is a critical aspect of this thesis as it directly impacts our model's predictive power. By identifying optimal beta values, we can sig-

nificantly improve the accuracy of our Bitcoin price predictions.This chapter delves into the process of parameter tuning, where these variables are carefully adjusted to achieve the best results. Replication plays a central role, laying the groundwork for the detailed analysis and ultimately leading to more precise predictions.In essence, parameter tuning is a vital step toward accomplishing this thesis's main goal: achieving accurate and reliable Bitcoin price forecasts.

4.6 IMPLEMENTATION STEPS

Step 1: Data Collection The Bitcoin data was collected from the Yahoo Finance website.

Step 2: Pre-processing This step incorporates the following:

1. Data Discretization: Part of data reduction, especially crucial for numerical data.

2. Data Integration: Integration of data files. After transforming the dataset into a clean format, it is divided into "training and testing" sets for evaluation. We create a data structure with timesteps of 1 and 1 output

. 3. Data Transformation: Normalization.

Step 3: Feature Selection: In this step, data attributes chosen to be fed to the neural network. For this study, Date and Close Price are chosen as selected features.

Step 4: Train the LSTM model The LSTM model is trained by feeding the training dataset. The model is initialized using random weights and biases. The proposed LSTM model consists of a sequential input layer followed by 3 LSTM layers and then a dense layer with activation. The output layer again consists of a dense layer with a linear activation function.

Step 5: Output Generation The RNN-generated output is compared with the target values, and the error difference is calculated. The Backpropagation algorithm is used to minimize the error difference by adjusting the biases and weights of the neural network.

Step 6: Test Dataset Update Step 2 is repeated for the test data set.

Step 7: Error and Companies' Net Growth Calculation By calculating deviation, we

check the percentage error of our prediction with respect to the actual price.

4.7 PREDICTION AND VISUALISATION

Once the LSTM model is trained using the optimal beta value, it is applied to predict the closing prices on the testing dataset. These predicted prices are then visually compared to the actual prices, providing an intuitive assessment of the model's accuracy.The results of these predictions are presented graphically, allowing for a clear and insightful analysis. This visualization helps in drawing conclusions regarding the influence of varying beta values within the LSTM model for stock price prediction.By identifying the optimal beta value that consistently produces the most accurate predictions, The model can be refined further, enhancing its predictive performance and reliability Figure 4.1 shows the output curves for sigmoid function. Figure 4.1 shows the output curves for sigmoid function.

Figure 4.1 shows the output curves for sigmoid function.

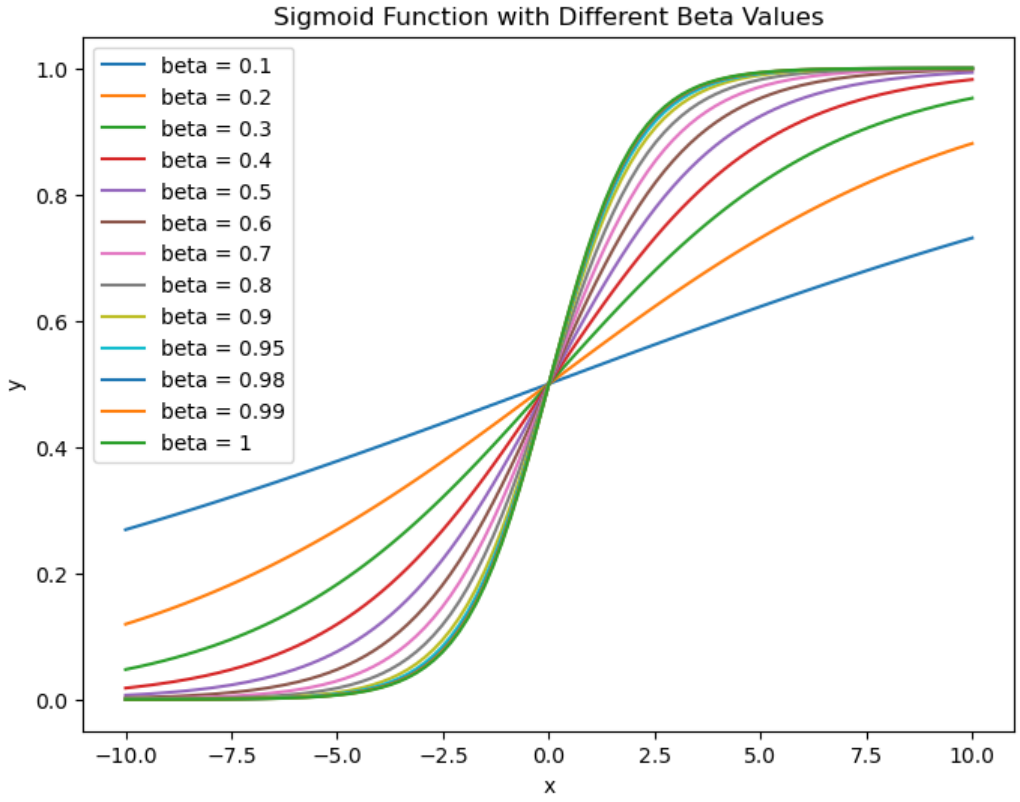

Figure 4.1: Sigmoid function with different beta

A comprehensive analysis of historical Bitcoin close price data is undertaken, spanning from January 1, 2018, to March 30, 2023. The analysis includes the visualization of Bitcoin's close price trends over this period, providing valuable insights into the cryptocurrency's price behavior and market dynamics.

Figure 4.2 shows a time series graph of the close price(target variable) of the Bitcoin data

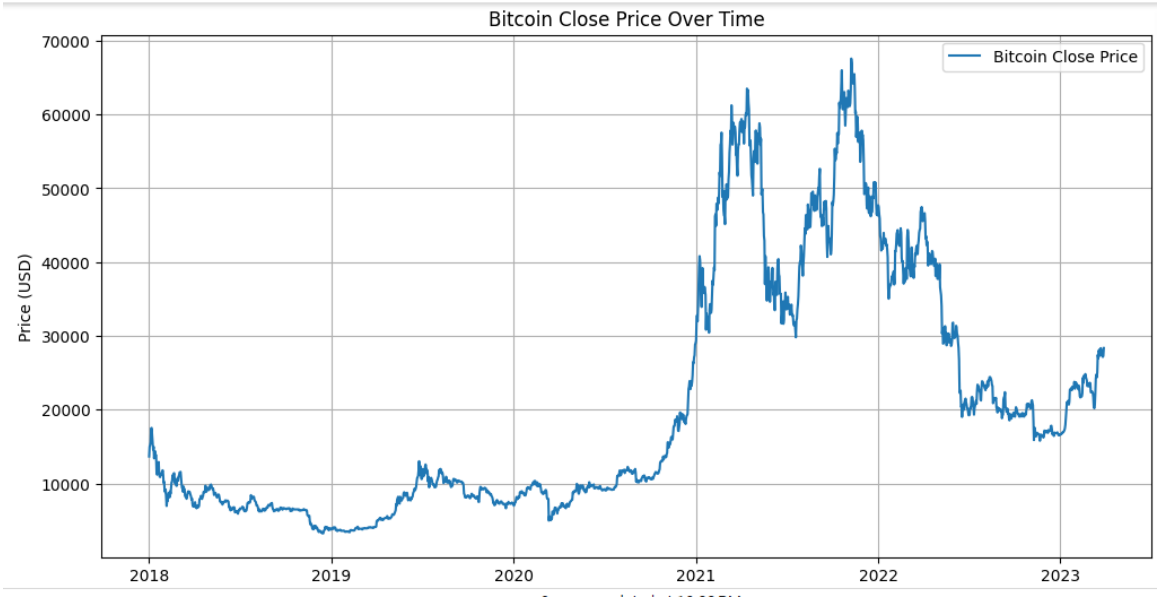

Figure 4.2: Time Series Graph of Bitcoin Close Price

The comprehensive visualization of the LSTM model's performance and an analysis of the predictions made by both the randomised sigmoid LSTM and the Standard LSTM model are as follows. Their performance is also compared based on Mean Absolute Error (MAE).

Figure 4.3 shows visualisation of the standard LSTM.It shows the plot of the actual bitcoin prices and the predicted prices by the standard LSTM. The graph indicates the predicted prices are close to the actual prices, The MAE for the standard LSTM model was 0.0087097.

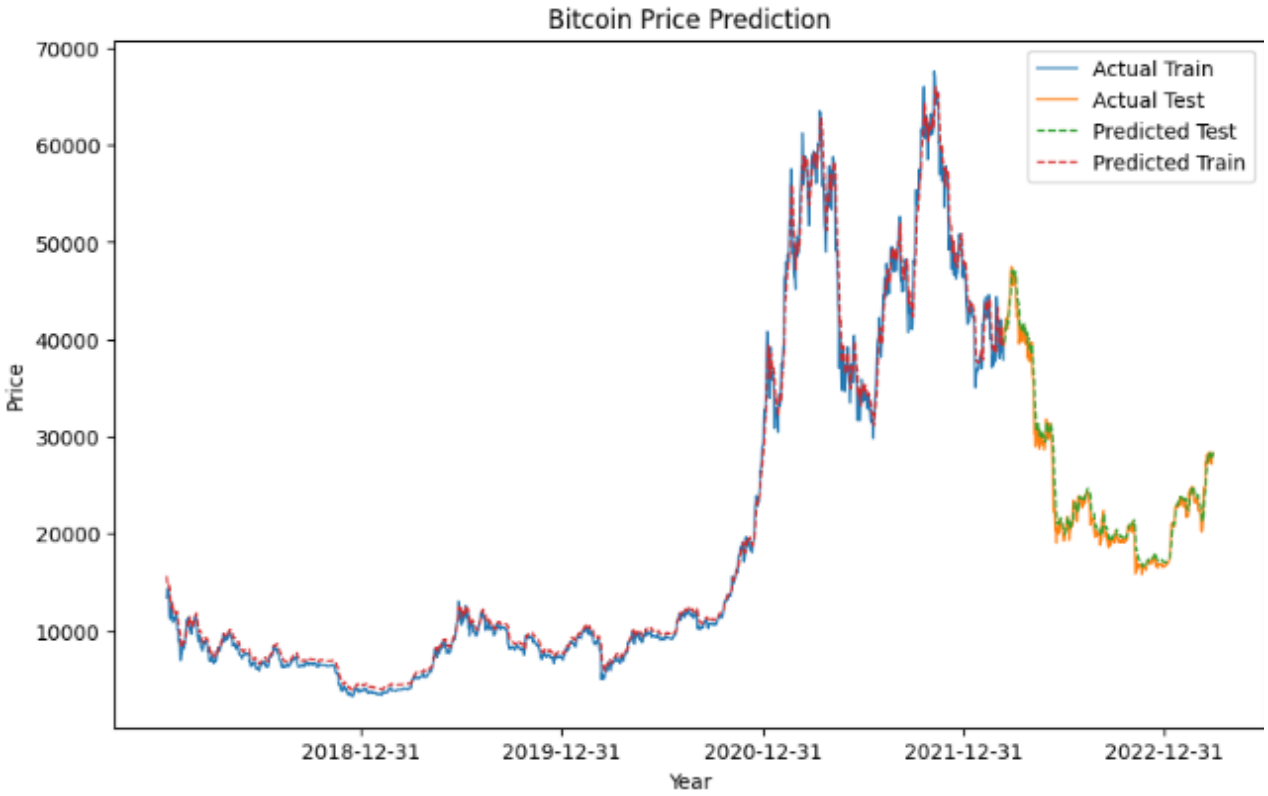

Figure 4.3: Actual vs. Predicted Prices

Figure 4.4 shows the visualisation of the randomised LSTM model. The plot shows the actual prices of bitcoin prices and the predicted prices.The MAE for this model was 0.0076270 indicating a better model than the standard LSTM.

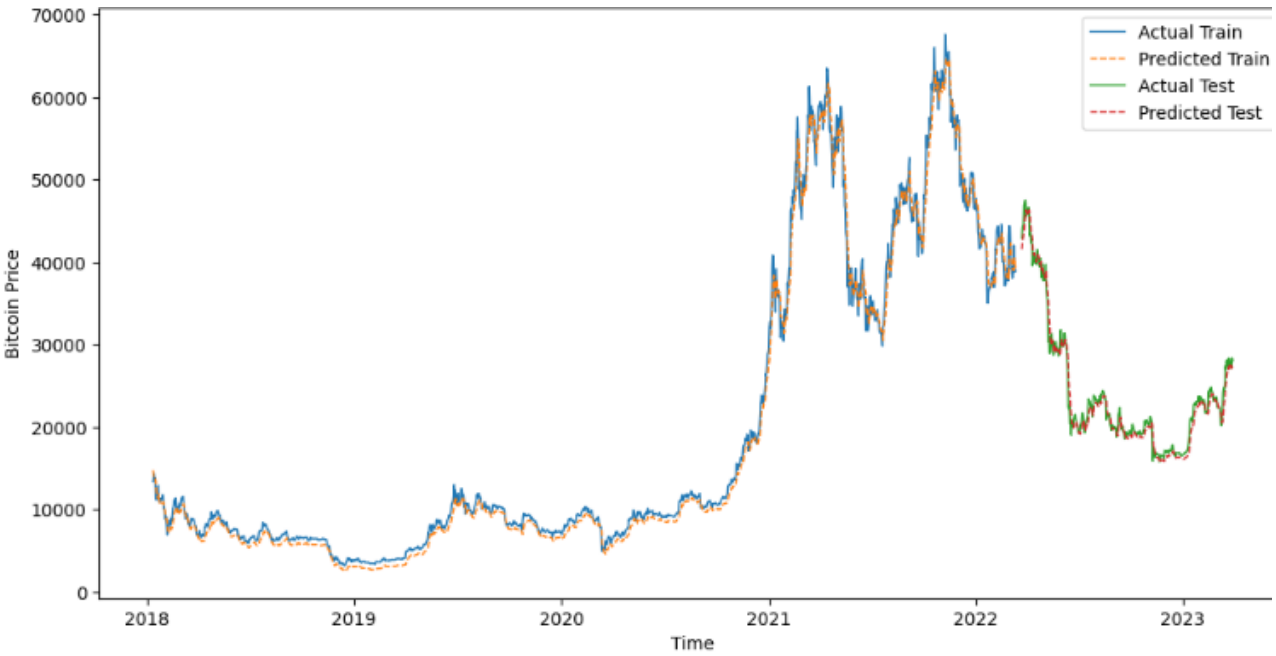

Figure 4.4: Actual vs.Standard LSTM Prediction

4.8 REPLICATION OF THE MODEL

This section outlines the pivotal role of simulations in our research. Through a series of carefully designed simulations, the optimal beta is identified for our model. The table below summarizes the outcomes of these simulations, highlighting the number of iterations conducted and the average optimal beta values derived.

With 5 replications the average beta was 0.65, for 10 replications it was 0.48, for 20 0.51499, for 50 replications 0.51890 was the best beta and 0.51990 for 100 replications. As the replication increased the best beta converges to 0.52.

Number of replication	Average best beta
5	0.6500
10	0.4800
20	0.51499
50	0.51890
100	0.51990

Table 4.2: Replication Results

4.9 FORECAST AND METRICS

In this section, we delve into the heart of our research by presenting a comparative analysis of actual Bitcoin prices (from March 30th to April 3rd, 2023), forecasted prices generated by the standard LSTM model, and average forecasted prices obtained through an array of simulations. These simulations encompass varying LSTM iterations (5, 10, 20, 50, and 100), allowing us to gain valuable insights into the predictive performance of our model. The comprehensive results are neatly summarized in Table 4.3

Table 4.3 provides a comprehensive view of the actual Bitcoin prices and the corresponding predictions made by both the standard LSTM model and the varied LSTM model. The average forecasted prices from simulations on the varied LSTM are included for comparison. Analyzing this table allows us to assess the accuracy of our models and explore the potential for improving forecasting precision.

Actual Price	LSTM	Varied LSTM(Average Forecast Prices)
28033.56	27771.69	28078.93
28478.48	27509.71	28128.44
28411.04	27247.62	28170.07
28199.31	26985.38	28215.84
27790.22	26723.01	28261.74

Table 4.3: Predicted values and Actual values

To analyze the performance of both models, we calculated the Mean Absolute Error (MAE) for their predictions. A comparison of MAE values and predicted Bitcoin prices for the models showed the MAE for the standard LSTM as 0.0087097 and 0.0076270 for the randomised LSTM which is lower indicating it evaluates well than the standard LSTM.Table 4.4 also provides a comparison of the Mean Absolute Error (MAE) values for the models, the Standard LSTM and the randomised LSTM with an average optimal beta value of 0.52. These MAE values offer insights into the models' performance in predicting Bitcoin prices. A lower MAE indicates a better model fit and, consequently, more accurate predictions.

MODEL	MAE VALUE
Standard LSTM	0.0087097
Varied LSTM((Average Optimal Beta=0.52)	0.0076270

Table 4.4: MAE Values for Different Betas

Chapter 5

CONCLUSION AND RECOMMENDATION

In summary, this study has effectively showcased the substantial influence of diverse sigmoid beta parameters on the performance of the LSTM model. The outcomes of this research consistently point to the superiority of a randomised LSTM model over the conventional LSTM model.

5.1 CONCLUSION

In summary, when assessing both the standard LSTM and the randomized LSTM through visualization, they both demonstrate strong performance. However, when we examine their numerical metrics, it becomes clear that the randomized LSTM outperforms the standard LSTM. Specifically, the randomized LSTM achieves a lower Mean Absolute Error (MAE) value of 0.0076270, while the standard LSTM's MAE is slightly higher at 0.0087097, indicating that the randomized LSTM exhibits superior performance in this regard.

5.2 SUMMARY FINDINGS

In essence, the study has underscored the considerable impact of employing randomised sigmoid beta parameters on the predictive capabilities of LSTM models for

Bitcoin price forecasting. It is evident that consistently, the utilization of a randomised beta parameter outperforms the traditional LSTM model, leading to more precise price predictions.

5.3 IMPLICATION OF THE STUDY

"It is advisable to explore optimal β values for different datasets since their effectiveness can vary based on dataset characteristics. Given the ever-changing nature of cryptocurrency prices, including additional factors like trading volume,blockchain and other features could improve prediction accuracy. Neural networks can exhibit randomness, so it's a good practice to replicate the model multiple times to ensure robust conclusions.

5.4 RECOMMENDATION

Based on the study's findings and implications, the following recommendations are made:

- Researchers and practitioners should explore the application of varied sigmoid beta parameters in LSTM models for other financial and time-series datasets to assess its potential benefits.
- Future studies could investigate the combination of LSTM models with other machine learning techniques or hybrid models to further improve predictive accuracy.
- It is advisable to consider alternative loss functions or evaluation metrics to complement the MAE and provide a more comprehensive assessment of model performance. This research contributes valuable insights into enhancing the predictive capabilities of LSTM models for cryptocurrency price forecasting, and it opens avenues for further exploration in this field.

APPENDIX A

.1 These Are Some Of The Codes Used

.1.1 LIBRARIES USED

```
import numpy as np
import pandas as pd
import yfinance as yf
import matplotlib.pyplot as plt
from scipy import stats
from sklearn.metrics import mean_absolute_error
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.layers import LSTM, Dense, Lambda
from tensorflow.keras.callbacks import EarlyStopping
```

Listing 1: Python Code Libraries

EXPLORATORY DATA ANALYSIS

```
 import yfinance as yf
import numpy as np
from scipy import stats

# Define the date range
start_date = '2018-01-01'
end_date = '2023-03-30'

# Fetch Bitcoin price data from Yahoo Finance
bitcoin_data = yf.download('BTC-USD', start=start_date, end=end_date
    )

```

```
# Calculate statistics for Open, High, Low, Close, and Volume
statistics = {
    'Open': {
        'Mean': np.mean(bitcoin_data['Open']),
        'Std Deviation': np.std(bitcoin_data['Open']),
        'Variance': np.var(bitcoin_data['Open']),
        'Median': np.median(bitcoin_data['Open']),
        'Max': np.max(bitcoin_data['Open']),
        'Min': np.min(bitcoin_data['Open'])
    },
    'High': {
        'Mean': np.mean(bitcoin_data['High']),
        'Std Deviation': np.std(bitcoin_data['High']),
        'Variance': np.var(bitcoin_data['High']),
        'Median': np.median(bitcoin_data['High']),
        'Max': np.max(bitcoin_data['High']),
        'Min': np.min(bitcoin_data['High'])
    },
    'Low': {
        'Mean': np.mean(bitcoin_data['Low']),
        'Std Deviation': np.std(bitcoin_data['Low']),
        'Variance': np.var(bitcoin_data['Low']),
        'Median': np.median(bitcoin_data['Low']),
        'Max': np.max(bitcoin_data['Low']),
        'Min': np.min(bitcoin_data['Low'])
    },
    'Close': {
        'Mean': np.mean(bitcoin_data['Close']),
        'Std Deviation': np.std(bitcoin_data['Close']),
        'Variance': np.var(bitcoin_data['Close']),
        'Median': np.median(bitcoin_data['Close']),
        'Max': np.max(bitcoin_data['Close']),
        'Min': np.min(bitcoin_data['Close'])
    },
    'Volume': {
        'Mean': np.mean(bitcoin_data['Volume']),
```

```
        'Std Deviation': np.std(bitcoin_data['Volume']),
        'Variance': np.var(bitcoin_data['Volume']),
        'Median': np.median(bitcoin_data['Volume']),
        'Max': np.max(bitcoin_data['Volume']),
        'Min': np.min(bitcoin_data['Volume'])
    }
}

# Print the calculated statistics
for price_type, stats in statistics.items():
    print(f"{price_type} Statistics:")
    for stat_name, stat_value in stats.items():
        print(f"{stat_name}: {stat_value:.2f}")
    print("\n")
```

Listing 2: Python Code Statistics

.1.2 CLOSE PRICE PLOT

```
  import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt

# Define the date range
start_date = '2018-01-01'
end_date = '2023-03-30'

# Fetch Bitcoin price data from Yahoo Finance
bitcoin_data = yf.download('BTC-USD', start=start_date, end=end_date
    )

# Calculate and print basic statistics
basic_stats = bitcoin_data.describe()

# Plot the closing prices over time
plt.figure(figsize=(12, 6))
```

```
plt.plot(bitcoin_data['Close'], label='Bitcoin Close Price')
plt.title('Bitcoin Close Price Over Time')
plt.xlabel('Date')
plt.ylabel('Price (USD)')
plt.legend()
plt.grid(True)

# Show the plot
plt.show()

# Print basic statistics
print("Basic Statistics:")
print(basic_stats)
```

Listing 3: Python Code Bitcoin Close Price

.1.3 Sigmoid Function Curves with Varying Beta

```
import numpy as np
import matplotlib.pyplot as plt

# Generate a range of x values
x = np.linspace(-10, 10, 400)

# Define the number of beta values to sample
num_beta_values = 10

# Create subplots for each sigmoid curve
plt.figure(figsize=(12, 6))
for _ in range(num_beta_values):
    # Sample a beta value uniformly between 0 and 1
    beta = np.random.uniform(0, 1)

    # Calculate the sigmoid curve for the sampled beta
    sigmoid_curve = 1 / (1 + np.exp(-beta * x))

```

```
    # Plot the sigmoid curve
    plt.plot(x, sigmoid_curve, label=f'Beta={beta:.2f}')

# Customize the plot
plt.title('Sigmoid Function Curves with Varying Beta (Uniform 0-1)')
plt.xlabel('x')
plt.ylabel('Sigmoid(x)')
plt.legend()
plt.grid(True)

# Show the plot
plt.show()
```

Listing 4: Python Code: Sigmoid Function Curves with Varying Beta

REFERENCES

Dhanya, N. M (2021). An Empirical Evaluation of Bitcoin Price Prediction Using Time Series Analysis and Roll Over." In Inventive Communication and Computational Technologies, pp. 327-339. Springer, Singapore.Journal of Soft Computing Paradigm (JSCP)Vol.03/ No.03Pages:205-217http://irojournals.com/jscp/ DOI:https:doi.org/10.36548/

Farokhmanesh, F., Sadeghi, M. T. (2019, April). Deep feature selection using an enhanced sparse group lasso algorithm. In 2019 27th Iranian Conference on Electrical Engineering (ICEE) (pp. 1549-1552). IEEE.

Yogeshwaran, S., Kaur, M. J., Maheshwari, P. (2019, April). Project based learning: predicting bitcoin prices using deep learning. In 2019 IEEE global engineering education conference (EDUCON) (pp. 1449-1454). IEEE.

Oliveira, J., Lima Rodrigues, L., Craig, R. (2011). Voluntary risk reporting to enhance institutional and organizational legitimacy: Evidence from Portuguese banks. Journal of Financial Regulation and Compliance, 19(3), 271-289.

Nakamoto, S. (2008).A peer-to-peer Electronic Cash System.Bitcoin whitepaper.Available at: https://git.dhimmel.com/bitcoin-whitepaper (Accessed on February 23, 2023)

Bouri, E., Gil-Alana, L. A., Gupta, R., Roubaud, D. (2019). Modelling long memory volatility in the Bitcoin market: Evidence of persistence and structural breaks. International Journal of Finance Economics, 24, 412–426.

Yeze, Z., Yiying, W. (2019, March). Stock price prediction based on information entropy and artificial neural network. In 2019 5th International Conference on Information Management (ICIM) (pp. 248-251). IEEE.

McNally, S., Roche, J., Caton, S. (2018, March). Predicting the price of bitcoin using machine learning. In 2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP) (pp. 339-343). IEEE.

Miura, R., Pichl, L., Kaizoji, T. (2019). Artificial neural networks for realized volatility prediction in cryptocurrency time series. In Advances in Neural Networks–ISNN 2019: 16th Inter-

national Symposium on Neural Networks, ISNN 2019, Moscow, Russia, July 10–12, 2019, Proceedings, Part I 16 (pp. 165-172). Springer International Publishing.

Chen, Wei, Huilin Xu, Lifen Jia, and Ying Gao. (2020a). Machine learning model for Bitcoin exchange rate prediction using economic and technology determinants. International Journal of Forecasting 37(1): 28–43.

Jang, H., Lee, J. (2017). An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. Ieee Access, 6, 5427-5437.

Ji, S., Kim, J., Im, H. (2019). A comparative study of bitcoin price prediction using deep learning. Mathematics, 7(10), 898. Dyhrberg, A. H. (2016). Bitcoin, gold and the dollar–A GARCH volatility analysis. Finance Research Letters, 16, 85-92. https://doi.org/10.1016/j.frl.2015.10.008 Nakano, M., Takahashi, A., Takahashi, S. (2018). Bitcoin technical trading with artificial neural network. Physica A: Statistical Mechanics and its Applications, 510, 587-609. Nakamoto, Satoshi. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Available

online: https://bitcoin.org/bitcoin.pdf (accessed on 7 October 2022) Livieris, I. E., Stavroyiannis, S., Pintelas, E., Pintelas, P. (2020). A novel validation framework to enhance deep learning models in time-series forecasting. Neural Computing and Applications, 32, 17149-17167. Wu, C. H., Lu, C. C., Ma, Y. F., Lu, R. S. (2018, November). A new forecasting framework for bitcoin price with LSTM. In 2018 IEEE International Conference on Data Mining Workshops (ICDMW) (pp. 168-175). IEEE. Kumar, D., Rath, S. K. (2020). Predicting the trends of price for ethereum using deep learning techniques. In Artificial Intelligence and Evolutionary Computations in Engineering Systems (pp. 103-114). Springer Singapore.

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6, 107–116. Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.