Executive Summary

Name: ***SANDHYA S***

Roll Number*: 2214503765*

Programme: ***MBA – Data Science and Analytics***

Semester: 4th

Project Name: Bitcoin Price Prediction using Machine Learning

Project Title: Bitcoin Price Prediction using Machine Learning

Institution: Manipal University, Jaipur

Researcher: Sandhya S

Supervisor: Sowmya AM, Assistant Professor, Sri Sairam College of Engineering

Objective:

This project aims to evaluate the predictive power of the N-BEATS deep learning architecture for forecasting Bitcoin prices using various time series data. The study compares the performance of the N-BEATS model against other popular time series forecasting methods, including Long Short-Term Memory (LSTM) and Autoregressive Integrated Moving Average (ARIMA). The goal is to determine which model provides the most accurate and reliable predictions for different time intervals, ultimately aiding investors and financial analysts in making informed decisions.

Background:

With the increasing relevance of cryptocurrencies in the global financial markets, accurate prediction of Bitcoin prices has become a focal point for investors and researchers. Bitcoin, being the most prominent cryptocurrency, experiences significant price fluctuations, driven by market demand, investor sentiment, and macroeconomic factors. The volatile nature of Bitcoin and its significant price fluctuations present a unique challenge for time series forecasting models. Traditional statistical models often fall short in capturing these complexities. However, recent advancements in deep learning have introduced sophisticated architectures capable of capturing complex patterns in financial data, promising improved prediction accuracy. The N-BEATS model, known for its robustness and flexibility, is particularly well-suited for this task.

Methodology:

The study employs the N-BEATS deep learning architecture to forecast Bitcoin prices using daily, hourly, and minute-by-minute data. The model's performance is measured against traditional forecasting models such as LSTM and ARIMA. The dataset includes historical Bitcoin price data, which is preprocessed to ensure quality and consistency. Features such as trading volume, market capitalization, and global economic indicators are considered to enhance the model's predictive power. Evaluation metrics include Mean Average Percentage Error (MAPE) and Root Mean Squared Error (RMSE), which quantify the prediction accuracy of each model. The models are trained and tested on separate data subsets to prevent overfitting and to validate their generalizability.

Key Findings:

- **Superior Predictive Performance:** The N-BEATS model demonstrated superior predictive performance compared to LSTM and ARIMA models. This suggests that N-BEATS is more effective at capturing the underlying patterns in Bitcoin price data.

- **Handling Non-linear Patterns:** The architecture effectively captured the non-linear patterns and volatility inherent in Bitcoin price movements. This ability to handle complex data patterns is critical for forecasting in volatile markets.
- **Evaluation Metrics:** The evaluation metrics, MAPE and RMSE, indicated lower error rates for the N-BEATS model, suggesting its potential for real-world applications in financial forecasting. These metrics highlight the accuracy and reliability of the model's predictions.
- **Versatility Across Timeframes:** The N-BEATS model's versatility was evident as it performed well across different timeframes, including daily, hourly, and minute-by-minute data. This flexibility is valuable for various trading and investment strategies that operate on different timescales.
- **Robustness to Market Changes:** The model's robustness to sudden market changes and shocks makes it a reliable tool for predicting future prices, even in highly dynamic and unpredictable markets like that of Bitcoin. Its ability to adapt to new data without significant loss of accuracy is a crucial advantage.
- **Feature Importance Analysis:** An analysis of feature importance revealed that certain external factors, such as trading volume and market sentiment indicators, significantly impact the model's predictions. This insight can help investors understand the key drivers of Bitcoin price movements and adjust their strategies accordingly.

Conclusion:

The findings of this study highlight the efficacy of the N-BEATS deep learning architecture in forecasting Bitcoin prices. Its enhanced predictive capabilities over traditional methods make it a promising tool for investors and financial analysts seeking to leverage machine learning for cryptocurrency market analysis. The model's ability to adapt to various timeframes and handle complex patterns makes it suitable for diverse financial applications. Future research could explore the integration of additional financial indicators and external variables to further improve forecasting accuracy. Additionally, examining the model's performance in different market conditions and its applicability to other cryptocurrencies could provide further insights.

Key Words: Bitcoin, machine learning, N-BEATS, time series forecasting, LSTM, ARIMA

Acknowledgments:

The researcher acknowledges the guidance and support of Prof. Sowmya AM and her valuable feedback.

BITCOIN PRICE PREDICTION USING MACHINE LEARNING

Project Report Submitted

in Partial fulfilment of the

requirement for theaward

of Degree of

MASTER OF

BUSINESS

ADMINISTRATI

ON (MBA)

Submitted by

Sandhya S

Reg No:

2214503765

Under the guidance

of

Sowmya AM

Guide Reg No:

MANIPAL UNIVERSITY

JAIPUR (MUJ)

DIRECTORATE OF

ONLINE EDUCATION

JUNE & 2024

ACKNOWLEDGEMENTS

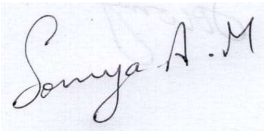
I would like to express my deepest gratitude to **Prof. Sowmya AM**, Assistant Professor, Department of Computer Science and Engineering, Sri Sairam College of Engineering for her invaluable guidance, continuous support, and encouragement throughout the course of this project. Her profound knowledge, insightful feedback, and unwavering commitment have been instrumental in shaping the direction and outcome of this research.

Prof. Sowmya AM mentorship has not only enhanced my academic skills but also inspired me to strive for excellence in my work. Her patience and willingness to assist at every stage of the project have been truly remarkable, and I am immensely thankful for her dedication and effort.

This project would not have been possible without her expert supervision and thoughtful advice. I am sincerely grateful for the opportunity to work under her guidance.

<u>BONAFIDE CERTIFICATE</u>

Certified that this project report titled **<u>“BITCOIN PRICE PREDICTION USING MACHINE LEARNING”</u>** is the bonafide work of “**<u>SANDHYA S</u>**” who carried out the project work under my supervision in the partial fulfilment of the requirements for the award of the MBA degree.



SIGNATURE

Name of the Guide: **SOWMYA AM**

Guide Registration Number:

DECLARATION BY THE STUDENT

I **SANDHYA S** bearing Reg. No **2214503765** hereby declare that this project report entitled **BITCOIN PRICE PREDICTION USING MACHINE LEARNING** has been prepared by me towards the partial fulfilment of the requirement for the award of the Master of Business Administration (MBA) Degree under the guidance of **SOWMYA AM.**

I also declare that this project report is my original work and has not been previously submitted for the award of any Degree, Diploma, Fellowship, or othersimilar titles.

Place: Bangalore

Name: Sandhya S

Date: 29/06/2024

Reg. No: 2214503765

EXECUTIVE SUMMARY

The rise of Bitcoin as a prominent cryptocurrency has spurred significant interest in forecasting its prices. This project explores the application of the N-BEATS (Neural Basis Expansion Analysis) deep learning architecture to predict Bitcoin prices. The study evaluates the predictive power of the N-BEATS model trained on Bitcoin daily, hourly, and minute-level data, comparing its performance with other popular time series forecasting methods such as Long Short-Term Memory (LSTM) and Autoregressive Integrated Moving Average (ARIMA).

Bitcoin's volatile nature presents a unique challenge for time series forecasting, requiring advanced models that can capture its intricate patterns. The N-BEATS model, known for its effectiveness in time series forecasting, is evaluated against traditional and contemporary methods to determine its robustness and accuracy.

The methodology involves data preprocessing, model training, and performance evaluation using metrics such as Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE). The results indicate that the N-BEATS model outperforms LSTM and ARIMA models in terms of predictive accuracy, showcasing its potential as a reliable tool for financial forecasting.

This study contributes to the growing body of research on cryptocurrency price prediction, highlighting the capabilities of deep learning architectures like N-BEATS in handling complex financial time series data. Future work can explore the integration of additional features such as trading volumes and market sentiment to further enhance the model's predictive performance.

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

INTRODUCTION

1.1 Background of Bitcoin

The first widely used cryptocurrency, Bitcoin, was introduced on October 31st, 2008, to a few cryptography enthusiasts via an email blast. The email included an attached paper called "Bitcoin: A Peer-to-Peer Electronic Cash System," which explained what Bitcoin is, how it works, and its architecture

Bitcoin is a purely peer-to-peer electronic payment system based on cryptographic proof instead of trust. It allows any two willing parties to transact directly with each other without the need for a trusted third party such as any financial institution. Bitcoin has emerged as the most popular and demanded cryptocurrency rising in its price as high as 20,000$ per Bitcoin in December 2017. The Bitcoin was quickly followed by many other alternative coins, derivatives of the original concepts, and other block chain-based cryptocurrencies of more or less sophisticated design such as Etherium or Ripple. The current market capitalization of all of the 5075 cryptocurrencies combined is over 340 billion USD (as of September 25th, 2020), with Bitcoin dominating the market and representing 56% of the total market capitalization

The unexpected appearance of such an unknown technology seen as a speculative financial asset with a bubble-like behaviour immediately drew much attention from the media, governments, policymakers, financial institutes, investors, and an academic community. Bitcoin became a topic of countless controversies. There are many ongoing debates concerning even the simplest questions regarding Bitcoin price formation mechanisms or its categorization. Bitcoin can be viewed as a currency, an asset, or even a "digital gold," a commodity sharing the same properties from various people's perspectives.

Nevertheless, cryptocurrencies have become commonly used as an alternative investment for diversifying portfolio risks hedging against stocks, currencies, gold, oil, and other financial assets.

As more people and financial institutions start to invest in the cryptocurrency markets, investors are interested in forecasting future price changes. Because of Bitcoin's highly volatile nature, there is a need for robust predictions to help in risk assessment and investment decisions. However, the research on how to accurately predict Bitcoin prices is still evolving.

Through the examination of past price shifts and market data, they acquire an invaluable understanding of possible trends and market emotions. These insights empower them to make knowledgeable and strategic trading decisions. These techniques have evolved into indispensable instruments in the trader’s repertoire, empowering them to adeptly and confidently traverse the dynamic and constantly shifting terrain of financial markets.

At present, BTC’s 24-hour trading volume stands at $31 billion. The circulating supply of BTC is 19.6 million coins. The 14-day relative strength index currently registers at 57.42.

1.2 Financial Time Series Forecasting

Stock market forecasting is considered as one of the most challenging issues among time-series predictions due to its noise and volatile features. How to predict future stock movements accurately is still an open question concerning modern society’s economic and social organization. This means that temporal relationships in the financial data exist, yet they are challenging to analyze and predict due to the non-linear and chaotic nature of it.

In recent years, machine learning has been gaining popularity among financial analysts and traders. Machine learning has proven to be very efficient in finding non-linear relationships in the time series. Various machine learning models, such as Artificial Neuron Networks (ANN) and the Super Vector Regression (SVR), have been utilized in financial time series forecasting to gain high predictive accuracy. However, in the literature, a recent trend in machine learning and pattern recognition communities considers that deep non-linear architectures should be applied to time series prediction. This new approach improves traditional machine learning models by extracting robust features and capturing relevant information to model complex real-world data.

Deep Learning is a form of machine learning that is inspired by the workings of the human brain and its data processing mechanisms. Recent advances in deep learning have drastically improved computers’ ability to recognize and label images, recognize and translate speech, and play games of skill at a better than human-level performance. Considering the complexity of financial time series, combining deep learning methods with financial market forecasting is a fascinating filed that is still relatively unexplored.

1.4.1 Bitcoin Time Series Forecasting

Although asset price forecasting is an important part of portfolio optimization and hedging, a small number of works have focused on the Bitcoin market. Most of the papers that tried to predict the Bitcoin prices used various statistical methods such as ARIMA. Other papers investigated the predictive power of various price movement indicators such as social media sentiment analysis, volume information, and google trends search data. There is a handful of papers that focused on deep learning methods for Bitcoin time series forecasting. Among those papers, the most popular deep learning architectures employed were SVMs, and LSTM. Many novel state of the art architectures for time series forecasting have been recently introduced, but none were tested on Bitcoin data.

1.3 Objective of the Study

The objective of this study is to enhance the accuracy and robustness of Bitcoin price prediction using machine learning techniques, with a focus on exploring and improving the N-Beats algorithm. N-Beats is a state-of-the-art deep learning model designed for time series forecasting, known for its flexibility and scalability. The study aims to investigate novel improvements and adaptations to the N-Beats algorithm, thereby optimizing its performance in predicting Bitcoin prices.

Specifically, the objectives include:

- Evaluating the current performance of N-Beats in Bitcoin price prediction.
- Identifying potential limitations or shortcomings of the existing N-Beats implementation.
- Proposing and implementing enhancements to the N-Beats algorithm tailored to the characteristics of Bitcoin price data.
- Assessing the effectiveness of the improved N-Beats model compared to baseline models and traditional forecasting methods.
- Providing insights into the factors influencing Bitcoin price movements and the applicability of machine learning techniques in cryptocurrency market analysis.
- By achieving these objectives, this study aims to contribute to the advancement of machine learning-based approaches for Bitcoin price prediction, offering valuable insights for investors, traders, and researchers in the cryptocurrency domain.

By achieving these objectives, this study aims to contribute to the advancement of machine learning-based approaches for Bitcoin price prediction, offering valuable insights for investors, traders, and researchers in the cryptocurrency domain.

1.4 Definitions:

1.4.1 **Cryptocurrency:** Any form of currency that only exists digitally.

1.4.2 **Bitcoin:** The first widely used and most popular digital currency created for use in peer-to-peer online transactions.

1.4.3 **Financial Market:** The aggregation of buyers and sellers of some asset.

1.4.4 **Time Series:** A series of values of a quantity obtained at successive times, often with equal intervals between them.

1.4.5 **Hedge:** An investment position intended to offset potential losses or gains by a companion investment.

1.4.6 **Volatility:** The degree of variation of a trading price series over time

1.4.7 **Machine Learning:** The process by which a computer can improve its performance by continuously incorporating new data into an existing statistical model.

1.4.8 **Deep Learning:** A subfield of machine learning concerned with algorithms inspired by the brain's structure called artificial neural networks**.**

1.5 Problem Definition:

Financial time series forecasting is a univariate point forecasting problem in discrete time. Given a forecast horizon with length-H and an observed series history length-T, then the task is to predict the vector of future values $y \in R^H = [y_{T+1}, y_{T+2}, ..., y_{T+H}]$, where $[y_1, ..., y_T] \in R^T$ [1]. Therefore, the machine learning model's goal is to learn a function that maps a sequence of past observations as input to a predicted observation as an output.

1.6 Motivation:

The financial time series forecasting models are vital for many stakeholders, such as investors, regulatory agencies, and governments. Such models are commonly used in risk assessment, portfolio building, and investment decision making. Nevertheless, a limited amount of works focus on the Bitcoin time series forecasting, which is becoming a common way of diversifying portfolios. Additionally, there are even fewer papers investigating the use of deep learning methods for this problem.

1.7 Thesis Statement

This research aims to utilize the novel N-BEATS deep learning architecture for time series forecasting to assess its predictive capabilities on the Bitcoin price data.

1.8 Outcomes and Contribution

This work evaluates the N-BEATS deep learning architecture's predictive power trained on the historical Bitcoin pricing data. The main contributions of this work are as follows:

- The N-BEATS deep learning model is developed and fine-tuned to forecast the future Bitcoin price data.
- The daily, hourly, and up-to-the-minute data sets are utilized in this study to analyze both the daily and high-frequency predictive capabilities of the architecture.
- The forecasting results are compared with other time series forecasting methods such as LSTM and ARIMA.

1.9 Organization of Thesis

The thesis is organized as follows: the literature review chapter presents prior research conducted in the fields of Bitcoin and machine learning methods in Bitcoin time series forecasting. The methodology chapter describes the details of the methodologies involved in the work. This includes selecting the training data, identifying the best parameters, training the N-BEATS model, and testing the model with outlined performance metrics. The findings chapter highlights the experimental results, including the analysis of these results and potential future works.

2. LITRATURE REVIEW

2.1. Bitcoin Time Series: an Economic Viewpoint

Initially, the literature has studied the Bitcoin only from the technical viewpoint, although recent literature has examined the economic features of the Bitcoin. For example, Lahmiri et al, revealed a significant property of the long-range memory in the Bitcoin time series data. Long-range memory leads to dependencies between distant time series trajectories of the investigated non-linear systems, namely for all Bitcoin markets. As a result, future prices can be predicted using algorithms that could utilize past information to detect the data's non-linear patterns. One of the main differences of the Bitcoin from other financial assets such as stocks or commodities is its high volatility, which makes forecasting future values more challenging. The Bitcoin had daily volatility of 35% on several occasions and ranged drastically in its prices throughout its history, as seen in below Figure

2.2 Bitcoin Time Series Forecasting

In recent years, machine learning gained much recognition in the finance sector. Previous studies reported that advanced machine learning algorithms could predict price changes in financial markets with high accuracy. Support Vector Machines (SVMs) and Artificial Neural Networks (ANNs) are prevalent in financial market forecasting due to their ability to recognize hidden patterns in non-linear, dynamic time-series data.

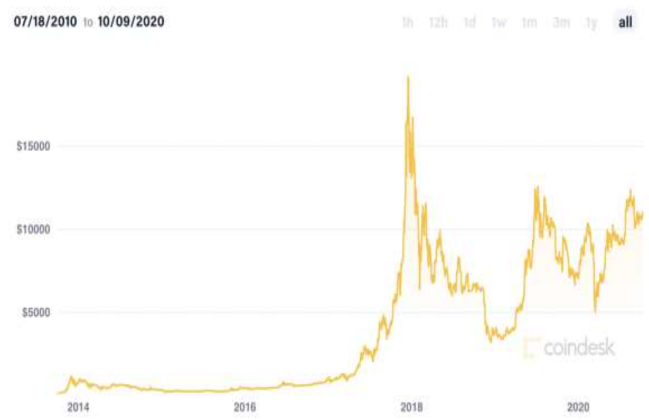


Figure 2-1: Historical Bitcoin Prices

Since the majority of the literature agrees that the Bitcoin market is mostly inefficient and the long-memory exists in the time series [20, 21, 22, 23, and 18] machine learning methods should be capable of detecting hidden dependencies and utilize them to forecast the future prices of the Bitcoin. Compared to traditional financial markets, there is much less literature that focuses on Bitcoin time series forecasting and even less literature investigating the possibilities of employing Machine Learning methods for predictions.

2.1.1 Machine Learning Approaches in Bitcoin Price Forecasting:

As with traditional markets, most of the papers employed either Support Vector Machines or Artificial Neural Networks. Among ANNs, the LSTM model was used in most studies [17, 24, 25, 26, 27, 28]. LSTM neural networks overcome the problem of vanishing gradients by replacing nodes in the RNN with memory cells and gating mechanisms which makes it efficient in memorizing long and short-term temporal information simultaneously.

McNally et al. used Bayesian optimized recurrent neural network (RNN) and LSTM network to forecast the direction of Bitcoin prices in USD. The price data used ranged from August 19th, 2013 until July 19th, 2016, and was sourced from the Bitcoin Price Index. The LSTM achieved the classification accuracy of 52% and an RMSE of 8%. The popular ARIMA model for time series forecasting was also compared to the deep learning models. The non-linear deep learning methods outperformed the ARIMA forecast, which performed poorly

Mudassir et al. used various classification and regression machine learning models for predicting Bitcoin price movements and prices in short and medium terms. The methods employed included artificial neural network (ANN), stacked artificial neural network (SANN), support vector machines (SVM), and LSTM. Technical indicators were used as inputs to the models. The study employed the one, seven, thirty, and ninety days forecast horizons. The study used data from April 1st, 2013, to December 31st, 2019. Different metrics were used to evaluate the performance of regression models: mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE), while F1 score and area under curve (AUC) were used to evaluate the classification models. The resulting models had an up to 65% accuracy for the next-day forecast and up to 64-64% accuracy for the seventh-ninetieth-day forecast. For the daily price forecast, the error percentage (MAPE) was 1.44%, while it varied from 2.88 to 4.10% for horizons of seven to ninety days. The best performing model overall was found to be LSTM. Performance evaluation results showed an improvement in daily closing price forecast and price increase/decrease forecasting.

Chowdhury et al. used an ensemble learning method to forecast daily Bitcoin prices. An ensemble learning technique uses a combination of various machine learning algorithms in order to solve one particular computational intelligence problem. It uses a voting mechanism that helps to overcome the biases and error rates of the individual (weak) models. The study obtained a 92.4% accuracy using the ensemble learning method, which was considered the best among all the models used in this paper.

Chen et al. investigated the forecasting of Bitcoin prices by applying different modeling techniques to samples with various data structures and dimensional features. The study used statistical (Logistic Regression, Linear Discriminant Analysis) and machine learning techniques (Random Forest, XGBoost, Quadratic Discriminant Analysis, Support Vector Machine, Long Short-term Memory). The study used a set of high-dimensional features and the basic trading features acquired from a cryptocurrency exchange for 5-minute interval price prediction. The results showed that the statistical methods performed better for low-frequency data with high-dimensional features, while the machine learning models outperformed statistical methods for high-frequency data. The study was the first to highlight the importance of the sample dimension in machine learning techniques.

De Souza et al. investigated whether Machine Learning methods, namely Support Vector Machines (SVM) and Artificial Neural Networks (ANN), can generate abnormal risk-adjusted returns when applied to the Bitcoin time series. Findings indicated that traders could earn conservative returns on a risk-adjusted basis, even accounting for transaction costs, when using SVM. Furthermore, the study suggested that ANN can explore short-run informational inefficiencies to generate abnormal profits and beat even buy-and-hold during strong bull trends.

Mallqui et al. utilized Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Ensemble algorithms (based on Recurrent Neural Networks and the k-Means clustering method) for price direction predictions. A total of 86 possible input attributes were used in the study. Five attribute selection techniques were used for feature engineering: correlation analysis, relief technique, information gain method, principal component analysis, and correlation-based feature subset selection. Correlation analysis was found to be the best method for feature engineering. The study used two data intervals. The first interval was used to compare the results with the previous studies, while the second interval included up-to-date data. The results showed that the selected attributes and the best machine learning model achieved an improved accuracy 62.91%

using the same period of information. Concerning the maximum, minimum, and closing Bitcoin prices regressions, it was possible to obtain Mean Absolute Percentage Errors between 1% and 2%.

Ji et al. compared various deep learning methods for Bitcoin price prediction: such as a deep neural network (DNN), a long short-term memory (LSTM) model, a convolutional neural network, a deep residual network, and their combinations. The study addressed both regression and classification problems. The former predicts the future Bitcoin price, and the latter predicts whether or not the future price will go up or down. Experimental results showed that LSTM-based prediction models slightly outperformed the other prediction models for Bitcoin price prediction (regression). The DNN-based models performed the best for classification. Besides, a profitability analysis showed that classification models were more effective than regression models for algorithmic trading.

Lamothe-Fernandez et al. used deep recurrent convolution neural network, deep neural decision trees, and deep learning linear support vector machines to predict the future prices of the Bitcoin. A sample of 29 initial features was used including information about the demand and supply, attractiveness, macroeconomic and financial variables. The study used quarterly data from 2011 to 2019 obtained from the IMF's International Financial Statistics (IFS), the World Bank, FRED Sant Louis, Google Trends, Quandl, and Blockchain.info. The research showed high precision results achieving a precision hit range of 92.61–95.27%.

Li et al. proposed a novel attentive LSTM network and an Embedding Network (ALEN) to forecast Bitcoin price fluctuations. In particular, an attentive LSTM network was used to capture the time dependency representation of Bitcoin price. An embedding network was used to capture the hidden representations from related cryptocurrencies. Attentive LSTM is an improvement over standard LSTM created by coupling them to attention processes. Experimental results demonstrated that ALEN achieved state-of-the-art performance among all baselines [28]. Uras et al. used historical data on prices and volumes to forecast the daily closing price series of Bitcoin, Litecoin, and Ethereum cryptocurrencies. They used both statistical and machine learning techniques. The study used two artificial neural networks: Multilayer Perceptron (MLP) and Long, short-term memory (LSTM). The research found that partitioning of datasets into shorter sequences, representing different price 'regimes,' allowed to obtain precise forecast as evaluated in terms of Mean Absolute Percentage Error (MAPE) and relative Root Mean Square Error (relative RMSE). The study was able to obtain a MAPE error of 0.007. Regarding the implemented algorithms, the best results were found with both regression models and the LSTM network.

2.1.2 Recent Developments in Time Series Forecasting

Recently, the famous M4 time series forecasting competition for the first time has included Machine Learning forecasting methods. The M4-Competition is the continuation of three previous ones organized by Spyros Makridakis (known as the M-Competitions), whose purpose was to identify the most accurate forecasting methods for different types of predictions. To get precise and compelling answers, the M4 Competition utilized 100,000 real-life series. It incorporated all major forecasting methods, including those based on Artificial Intelligence (ML) and traditional statistical ones.

The winner of the M4 competition, with a substantial margin, was Smyl Slawek from Uber technologies with a hybrid Exponential Smoothing-Recurrent Neural Networks (ES-RNN) method. It used a mix of hand-coded parts with a black-box recurrent neural network (RNN) forecasting engine. Nevertheless, ElementAI (a startup co-founded by Yoshua Bengio) recently published a paper introducing a pure deep learning method for time-series predictions that beat ES-RNN's score in M4. N-BEATS is a neural-network-based model for univariate time-series forecasting. The architecture has several desirable properties, being interpretable, applicable without modification to a wide array of target domains, and fast to train. The model demonstrated state-of-the-art performance for all the datasets, improving forecast accuracy by 11% over a statistical benchmark and by 3% over last year's winner of the M4 competition. Since it has been introduced very recently, there are no papers that utilized it for Bitcoin time series prediction.

2.3 Testing Accuracy of Forecasting Models

The standard approach to test the forecasting models' accuracy in the literature is to split a time series into two non-overlapping sets for model training and testing. The training set is used to estimate the forecasting model's parameters, and the testing set is used to test the model on previously unseen data. Most papers used 80 percent of data for training, and the remaining 20 percent for testing and validation [17, 24, 25, 26, 27, 28]. The predictions are compared to the target variable's actual values in the test set to measure forecast accuracy. This allows us to compare models in terms of their predictive accuracy on the hold-out data set. Root Mean Squared Error (RMSE) and Mean Average Percentage Error (MAPE) were the most common metrics for evaluating the regression models.

2.4. Parameters in Determining the Best Machine Learning Model

To conclude, we can see that many parameters can affect the accuracy of the Machine Learning model: the machine learning method employed, the forecasting horizon, the market maturity, the usage of technical analysis indicators, the static or dynamic approach, and the model-assessment method all significantly affected the forecast accuracy.

The following chapter introduces the proposed research approach in detail. This includes a detailed description of the N-Beats model, data collection, data preprocessing, feature engineering, model training, metrics, and evaluation.

3. METHODOLOGY

This chapter introduces the proposed research-based model for forecasting the Bitcoin prices in details. This includes the steps for building the forecasting model, performance metrics, and evaluation strategy. Figure 3-1 presents an overall methodology pipeline of this study.

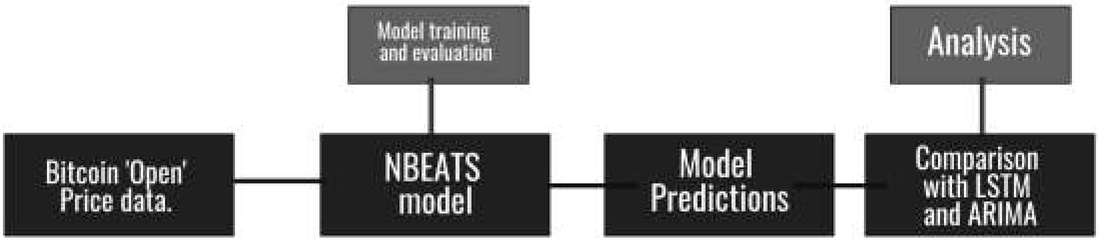


Figure 3-1: Methodology Pipeline

The first step of the pipeline is data preparation based on the Bitcoin 'Open' price values. The following step consists of building and training the N-BEATS model. The model performance is evaluated using several metrics and compared with other selected time series forecasting methods such as LSTM and ARIMA (Autoregressive Integrated Moving Average).

3.1 N-BEATS Architecture

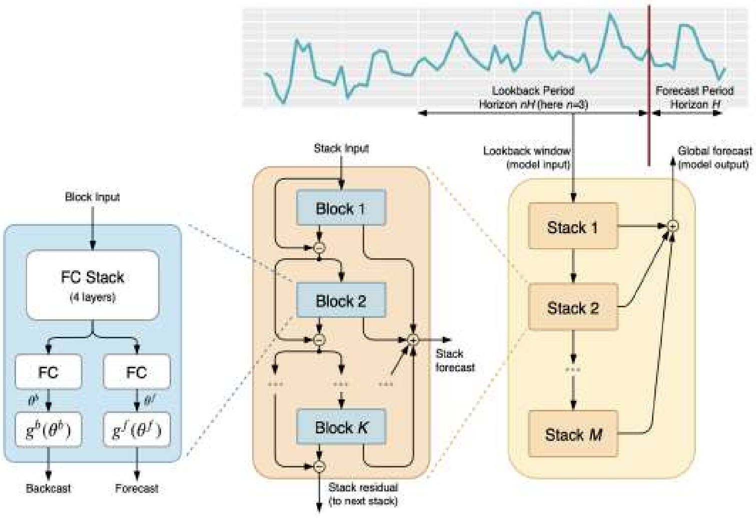


Figure 3-2: N Beats architecture [1]

N-BEATS (Neural Basis Expansion Analysis for interpretable Time Series forecasting) is a deep learning architecture based on backward and forward residual links and a very deep stack of fully-connected layers used for univariate time-series forecasting. The architecture has several desirable properties such as: being interpretable, applicable without modification to a wide array of target domains, and fast to train. Unlike other architectures like LSTM, N-BEATS architecture does not rely on any time-series-specific feature engineering or input scaling. The

N-Beats architecture consists of a stack of stacks where each stack comprises multiple basic blocks. The figure 3-2 depicts the architecture in detail.

Figure 3-3 provides an example of a basic block's detailed internal structure. As we can see from the figure, the basic block has a fork-like architecture. A look back window of length nH from the time series is serving as the model input; n is the number of data points required to forecast one data point into the future. H is the desired forecast horizon, which means how many data points into the future we want to predict. H is equal to 5 in this example; therefore, the look-back period is equal to 15 data points. Thus, 15-dimensional input from the look-back period is passed through a four-layer [FC(Fully Connected)+Relu] stack divided into two parts. Both of those two parts are further passed through another FC layer generating two outputs: a 15-dimensional back cast vector and a 5-dimensional forecast vector (5 data points forecast). Therefore, the basic block predicts the future data points and the input data in the form of the back cast.

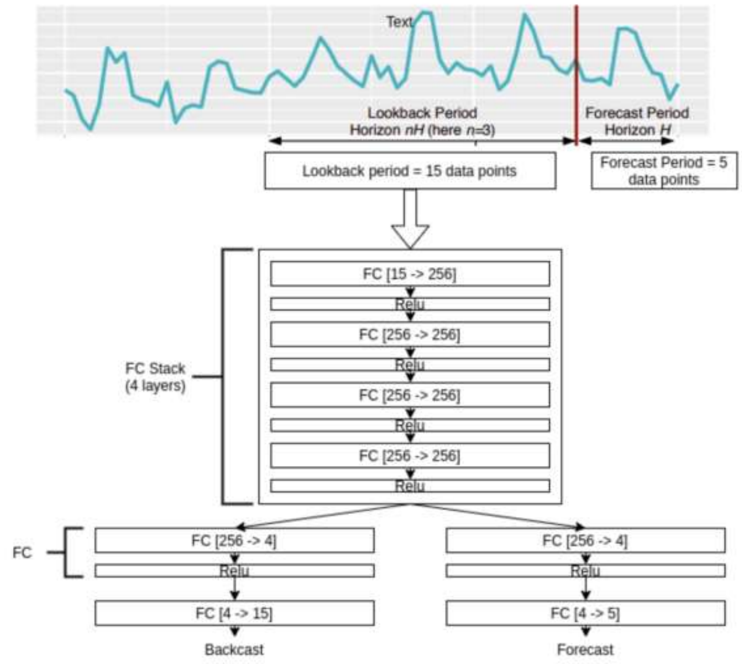


Figure 3-3: NBEATS basic building block [1]

3.3.1 Basic Block Stacking

Multiple N-BEATS basic blocks are combined into a single stack. The basic blocks in the stack are arranged following a double residual stacking manner. It is named double residual stacking because there are two arithmetic operations with the basic block's output (i.e., backcast and forecast).

The backcast and the forecast vectors are the two outputs of the first basic block after it processes input from the look-back period. The backcast is then used to calculate the input to the next block, an element-wise subtraction of backcast with the new look-back input (i.e., backcast - look-back). By performing subtraction of the new look-back input from the backcast, we get a vector that incorporates only things not learned enough by the first block, which will be passed as input to the next block. Therefore an input to every other next block would be a vector made up of element-wise subtraction of the previous block's backcast output and input.

The backcast output of the last block in a stack is called stack backcast output. The forecast outputs of all the blocks in the stack are used to calculate a stack forecast output, which is an element-wise addition of all the outputs.

Each stack is also used to build another more giant stack. The explanation above shows that each stack's input is a backcast output from a previous stack (things not learned by the previous stack).

Stack forecast output from all of the stacks is summed element-wise to yield the final global forecast vector. Loss between real and predicted values is calculated using MSE (Mean Squared Error). At the end of each training cycle, the model gradients are updated based on the loss value.

3.3.2 Meta Learning

From the explanation above, we can see that what the N-BEATS model does is essentially called meta-learning. Meta-learning is the learning process that can be divided into two parts: an inner and an outer training loop. The inner training loop focuses on task-specific knowledge, while the outer loop focuses on the overall across-task knowledge.

In N-BEATS architecture, backcast forecasting θ is responsible for task-specific knowledge, which incorporates the knowledge learned from the most recent look-back period. Gradient descent, which trains the weight matrices that θ depends on, is responsible for learning the bigger picture.

3.3.3 N-Beats Bitcoin Forecasting Model Implementation

The proposed model has been implemented using a PyTorch implementation of the N BEATS model by Philippe Remy.

3.2. Historical Bitcoin Pricing Data

Most studies such as [30] [26] and [27] used only daily frequency for prices in predictions. This study uses three frequencies - daily, hourly, and up-to-the-minute Bitcoin price data from Bitstamp exchange. This allows us to assess the predictive power of the N-BEATS architecture on both daily and high-frequency data. All data sets include data ranging from September 2012 to October 2020.

Each data set in the study features several variables: Date, Open, High, Low, and Close. The

N-BEATS model does not require any data preprocessing or specific feature engineering; therefore, a 1D NumPy array with Bitcoin 'Open' prices was used as the only input to the model.

Each dataset was split into two subsets: training and test datasets, where 80% of each set were used for training and the remaining 20% were used for testing.

3.3 Model Evaluation and Analysis

3.3.1 **Prediction Accuracy Measures:** The Root Mean Squared Error (RMSE) and the Mean Average Percentage Error (MAPE) are used to assess a models' price predictions' adequacy. These measures have also been used by many other studies such as [27] [26], and [4]. The MAPE and RMSE can be calculated according to equations 3.1 and 3.2, respectively.

$$MAPE = \frac{1}{T}\sum_{i=1}^{T}\left|\frac{d_i - \hat{d}_i}{d_i}\right| \times 100 \quad (3.1)$$

$$RMSE = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(d_i - \hat{d}_i)^2} \quad (3.2)$$

Where T is the total number of testing data, while di and ˆdi are the desired and predicted outputs, respectively.

3.3.2 **Model Performance Evaluation:** The N-BEATS model forecasting results are analyzed and compared with LSTM model and a popular statistical method ARIMA, both of which are regarded as the most reliable and widely used forecasting methods for time series. Due to its architecture, the LSTM model can learn and remember long sequences of data without relying on a pre-specified window lagged observation as input. On the other hand, ARIMA is a robust statistical method that tends to perform well on relatively short time-series data.

4. FINDINGS (ANALYSIS AND EVALUATION)

This chapter presents the analysis that was performed to test the effectiveness of the N-BEATS Bitcoin forecasting model. MAPE and RMSE statistical measures were used to evaluate the accuracy of the model on the test sets. The N-BEATS model results were also compared with other time series forecasting models trained on the same training data set. These models include a machine learning model LSTM and a statistical model ARIMA.

4.1 Experimental Models Training

The best performing N-BEATS model for daily Bitcoin price forecasting was found to have the following parameters: a look-back period of 3 days and a forecasting horizon of 1 day. The model consists of two stacks with three basic blocks per one stack and 128 hidden layer units per block. The models for hourly and up-to-theminute Bitcoin price forecasting have the same parameters except for the best lookback periods, which were found to be 6 hours and 6 minutes, respectively. That means that the models performed best using three previous days to forecast the next day price for daily data, six previous hours for hourly data prediction, and six previous minutes for minute data prediction. The models were trained for 25 epochs with an Adam optimizer and a batch size of 128. The MSE was used as a loss function.

The LSTM model consists of an input layer with 50 neurons followed by three hidden layers with 50 neurons each and an output layer. Additional 0.2 dropout units were placed in between all layers for regularization. The models were compiled with an Adam optimizer and an MSE loss function. The training time was set to 100 epochs with a batch size of 32. LSTM models' data was scaled to a 0-1 range and shaped into a format required for the model. The N-BEATS model performed worse with scaled data since the model does not require any data preprocessing. The LSTM model used a rolling forecast with the previous 60 days to forecast the next day's value.

The ARIMA model was built with the lag value equal to 5 for auto-regression, meaning it used an auto-correlation between values 5 data points apart. The difference order of 1 was used to make the time series stationary. A moving average model was set to 0. The model used the rolling forecast to predict the next data point, meaning that it performed parameter re-estimation after adding a new previously forecasted data point from test set into the training set.

4.2 Statistical Evaluation for the Models

This section presents the statistical analysis results to test the N-BEATS models' effectiveness compared to LSTM and ARIMA. The RMSE and MAPE were used to

evaluate the forecasting accuracy for the three models. The daily data forecasting results are presented in the table 4.1. As we can see on the table, the N-BEATS model has a MAPE of 2.261% and an RMSE of 308.859, which is better than the results of the LSTM model that has a MAPE of 2.976% and an RMSE of 370.051, and also slightly better than the results of the ARIMA model with the MAPE of 2.281% and an RMSE of 309.756.
The results for the hourly data are presented in the table 4.2. All models have seen an improvement in accuracy with higher frequency hourly data. The N-BEATS model has an RMSE of 59.303 and a MAPE of 0.388%. LSTM has an RMSE of 211.510 and a MAPE of 1.691%. The ARIMA has an RMSE of 59.307 and a MAPE of 0.386%.

The results for the minute data are presented in the table 4.3. All models have seen another improvement in accuracy with up-to-the-minute data. The N-BEATS model has an RMSE of 13.678 and a MAPE of 0.096%. LSTM has an RMSE of 46.419 and a MAPE of 0.430%. The ARIMA has an RMSE of 13.9 and a MAPE of 0.098%.
All of the results are also visualized using grouped bar plots in figures 4-1, 4-2, 4-3, and 4-4 for better visual understanding.

Table 4.1: Results Daily Data.

Model	**RMSE**	**MAPE**
N-BEATS	**308.859**	**2.261%**
LSTM	370.051	2.976%
ARIMA	309.756	2.281%

Table 4.2: Results Hourly Data.

Model	**RMSE**	**MAPE**
N-BEATS	**59.303**	0.388%
LSTM	211.510	1.691%
ARIMA	59.307	**0.386%**

Table 4.3: Results Minute Data.

Model	**RMSE**	**MAPE**
N-BEATS	**13.678**	**0.096%**
LSTM	46.419	0.430%
ARIMA	13.9	0.098%

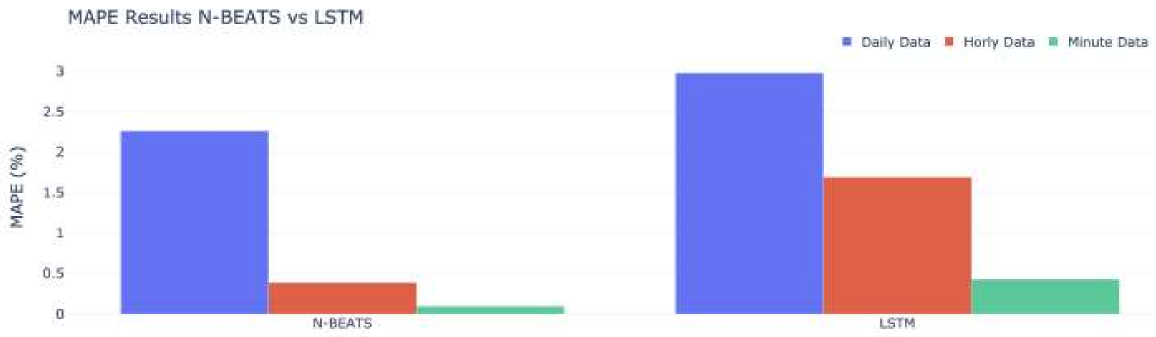


Figure 4-1: MAPE Results: N-BEATS/LSTM

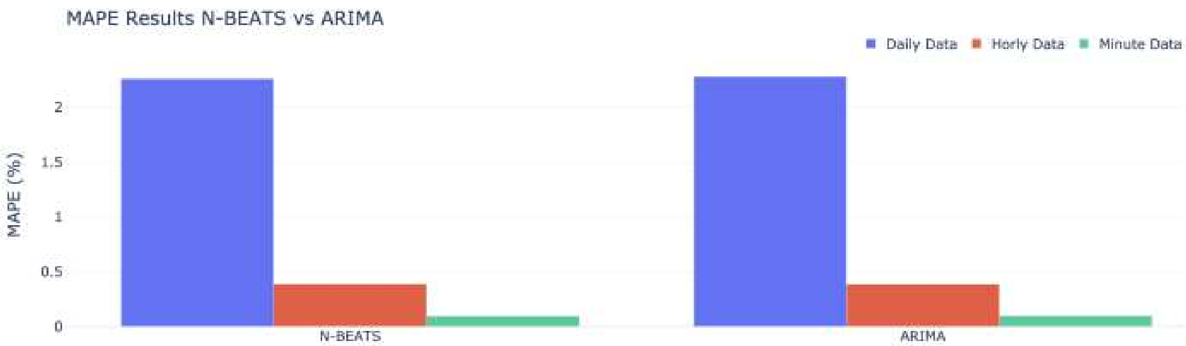


Figure 4-2: MAPE results: N-BEATS/ARIMA

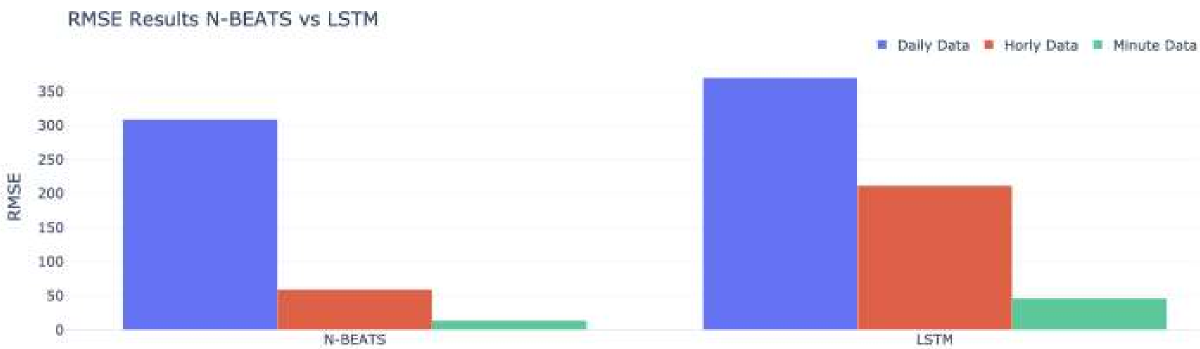


Figure 4-3: RMSE Results: N-BEATS/LSTM

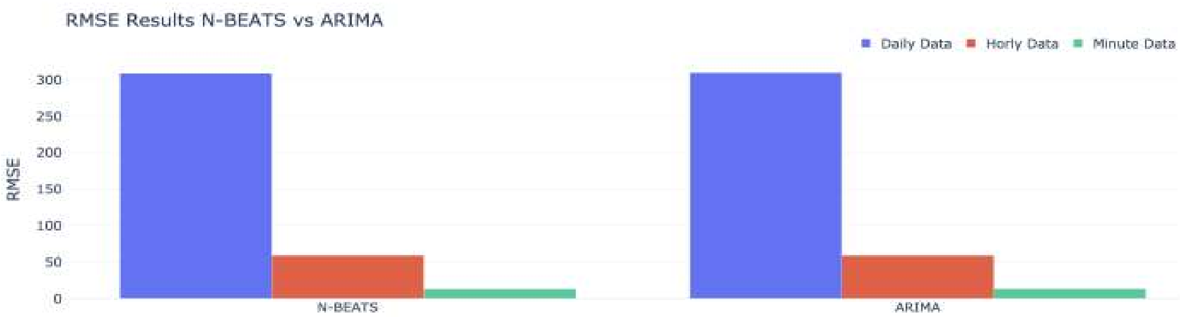


Figure 4-4: RMSE results: N-BEATS/ARIMA

4.3 Discussions

The results show that all models were able to forecast the prices with high accuracy. The individual plots for predictions on daily, hourly, and up-to-the-minute data for each model are presented in figures 4-6 - 4-14. The developed N-BEATS model showed promising results that were slightly better than the ARIMA results and significantly better than the results of LSTM.

2.1 Model Advantages

One of the most significant advantages of N-BEATS architecture compared to other deep learning architectures is interpretability. Any time series can be decomposed into three components trend, seasonality, and noise/residual. The actual observed time series is the result of the sum of three other components. The trend component determines the direction of the time series, where it can either increase or decrease. The seasonality component represents the repeating short-term cycles in the data. The noise represents the amount of random variation in the data. Such trend and seasonality decomposition can be built into the N-BEATS model to enable the results to be human interpretable. This paper utilizes the interpretable two-stack architecture. The interpretable model uses the values of the back stack to perform the decomposition. The first stack is responsible for learning the trend component, and the second stack is learning the seasonality of the data. Figure 4-5 depicts the decomposition of the Bitcoin time series data. As we can see from the graph, the Bitcoin data has a clear upside trend until mid-2018, then a downward trend until 2019, followed by another upward trend. The time series also has a slight seasonality and a high amount of randomness, especially at times of turmoil like the December 2017 price crash.

Another advantage of the N-BEATS is that it uses ensembling of models with different input horizons as a regularization technique to improve the performance. It was found that ensembling yielded better results than using dropout or L2 norm penalty on individual models since different input horizons provide different trend and seasonality representations of the model's data to learn from.

N-BEATS training time is also better compared to other deep learning architectures. The N-BEATS neural network improves training time by early stoppage, and determining the number of batches on the validation set. The GPU-based training of the N-BEATS model on the daily data takes around 1 minute; the hourly data model training takes around 3 minutes, while the minute data model training takes around 20 minutes. In comparison, the LSTM model took around 5 minutes for daily data, 30 minutes for the hourly data, and 6 hours for minute data.

4.4 Limitations

This research showed a significant potential to create models using machine learning to forecast the future financial time series data of Bitcoin prices. As with any methodology, there are some limitations.

- The N-BEATS model has shown promising results with one dimensional 'Open' price input compared to other models such as LSTM. However, the main limitation of the N-BEATS deep learning architecture is that it can accept only one-dimensional input. Therefore, it cannot process any additional features (e.g., volume, buy/sell indicators, moving averages) to improve the model's accuracy.
- The model parameters (such as the number of stacks and number of neurons per layer) could be further fine-tuned for better performance.
- This paper, we investigated the predictive power of the N-BEATS model with a forecasting horizon of one day into the future. It would also be interesting to increase the forecasting horizon to one week or more and compare the results with other models once again.

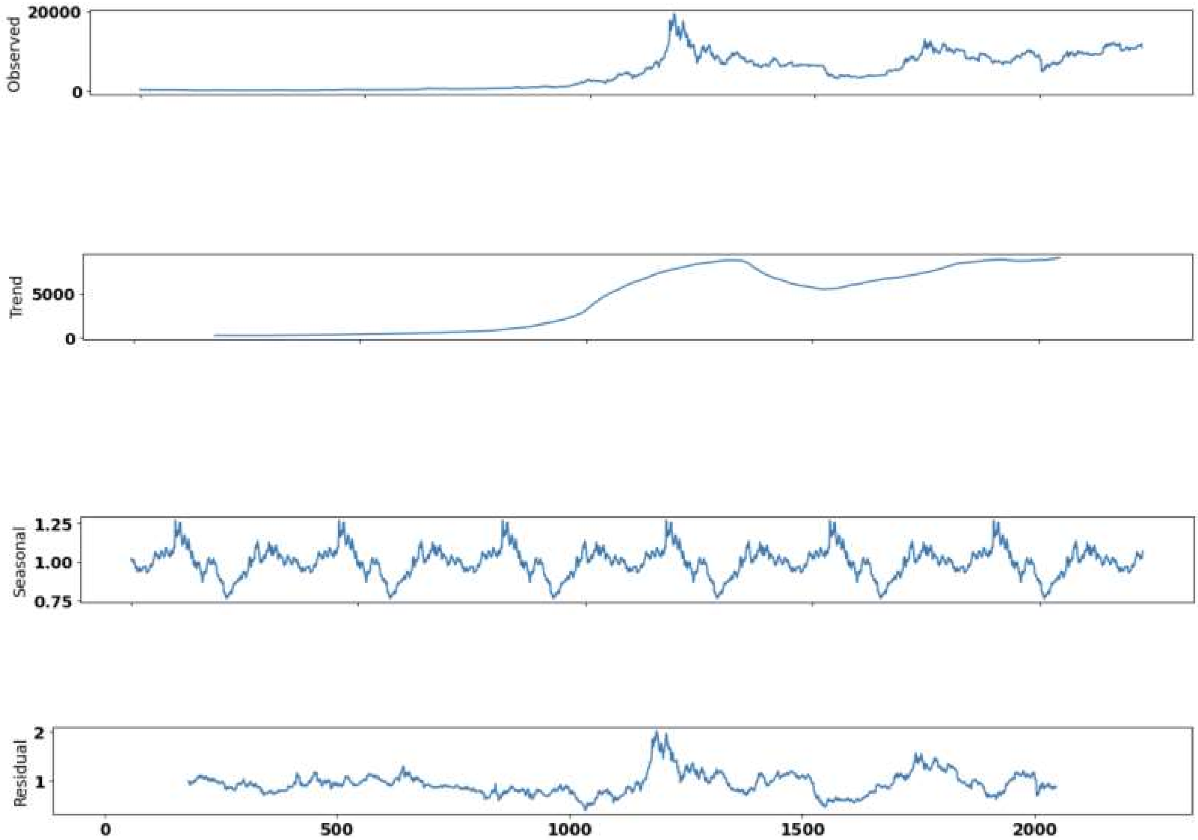


Figure 4-5: Bitcoin Time Series Decomposition

5. IMPLEMENTATION

5.1 SOFTWARE AND TOOLS USED

For this project, various software and tools were employed to handle data preprocessing, model implementation, training, and evaluation. The primary tools used include:

- **Python**: The main programming language used for implementing machine learning models.
- **Tensor Flow**: A deep learning framework used for building and training the N-BEATS and LSTM models.
- **Keras**: An API running on top of Tensor Flow for easier model building.
- **Stats models**: A Python library used for implementing the ARIMA model.
- **Pandas**: A data manipulation library for handling and pre-processing the data.
- **Numpy**: A fundamental package for scientific computing with Python.
- **Matplotlib** and **Seaborn**: Libraries for data visualization.
- **Jupyter Notebook / VS Code**: An interactive computing environment for writing and running code.

5.2 CODE FOR DATA PREPROCESSING: Data preprocessing is a crucial step in preparing the dataset for model training. Below is the code snippet used for data pre-processing:

<CODE>

```
import pandas as pd
import numpy as np

# Load the dataset
data = pd.read_csv('bitcoin_price.csv')

# Convert the 'Date' column to datetime format
data['Date'] = pd.to_datetime(data['Date'])

# Set the 'Date' column as the index
data.set_index('Date', inplace=True)

# Handle missing values
```

```
data.fillna(method='ffill', inplace=True)

# Normalize the data
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
data_scaled = scaler.fit_transform(data)

# Split the data into training and testing sets
train_size = int(len(data) * 0.8)
train, test = data_scaled[:train_size], data_scaled[train_size:]
```

5.3 CODE FOR N-BEATS MODEL: The N-BEATS model is implemented using Tensor Flow and Keras. Below is the code for building and training the N-BEATS model:

<CODE>

```
import tensorflow as tf
from tensorflow.keras import layers

# Define the N-BEATS block
class NBeatsBlock(layers.Layer):
    def __init__(self, input_dim, output_dim, num_layers, num_units):
        super(NBeatsBlock, self).__init__()
        self.num_layers = num_layers
        self.input_dim = input_dim
        self.output_dim = output_dim
        self.dense_layers = [layers.Dense(num_units, activation='relu') for _ in range(num_layers)]
        self.theta = layers.Dense(output_dim)

    def call(self, inputs):
        x = inputs
        for layer in self.dense_layers:
            x = layer(x)
        theta = self.theta(x)
        return theta
```

```
# Define the N-BEATS model
class NBeatsModel(tf.keras.Model):
    def __init__(self, input_dim, output_dim, num_blocks, num_layers,
num_units):
        super(NBeatsModel, self).__init__()
        self.blocks = [NBeatsBlock(input_dim, output_dim, num_layers, num_units)
for _ in range(num_blocks)]

    def call(self, inputs):
        block_outputs = [block(inputs) for block in self.blocks]
        return tf.reduce_sum(block_outputs, axis=0)

# Initialize and compile the model
input_dim = train.shape[1]
output_dim = train.shape[1]
model = NBeatsModel(input_dim=input_dim, output_dim=output_dim,
num_blocks=3, num_layers=4, num_units=64)
model.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
history = model.fit(train, train, epochs=50, batch_size=32, validation_data=(test,
test))
```

5.4 CODE FOR LSTM MODEL

The LSTM model is implemented using TensorFlow and Keras. Below is the code for building and training the LSTM model:

<CODE>

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

# Define the LSTM model
model_lstm = Sequential()
model_lstm.add(LSTM(50, return_sequences=True, input_shape=(train.shape[1],
train.shape[2])))
```

```
model_lstm.add(LSTM(50, return_sequences=False))
model_lstm.add(Dense(1))

# Compile the model
model_lstm.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
history_lstm = model_lstm.fit(train, train, epochs=50, batch_size=32,
validation_data=(test, test))
```

5.5 CODE FOR ARIMA MODEL

The ARIMA model is implemented using the Statsmodels library. Below is the code for building and training the ARIMA model:

<CODE>

```
from statsmodels.tsa.arima.model import ARIMA

# Fit the ARIMA model
model_arima = ARIMA(train, order=(5, 1, 0))
model_arima_fit = model_arima.fit()

# Predict the values
predictions_arima = model_arima_fit.forecast(steps=len(test))
```

5.6 CODE FOR MODEL EVALUATION

Evaluation of the models is performed using metrics such as Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE). Below is the code for evaluating the models:

<CODE>

```
from sklearn.metrics import mean_absolute_percentage_error,
mean_squared_error

# Evaluate N-BEATS model
predictions_nbeats = model.predict(test)
mape_nbeats = mean_absolute_percentage_error(test, predictions_nbeats)
rmse_nbeats = np.sqrt(mean_squared_error(test, predictions_nbeats))
```

```
# Evaluate LSTM model
predictions_lstm = model_lstm.predict(test)
mape_lstm = mean_absolute_percentage_error(test, predictions_lstm)
rmse_lstm = np.sqrt(mean_squared_error(test, predictions_lstm))

# Evaluate ARIMA model
mape_arima = mean_absolute_percentage_error(test, predictions_arima)
rmse_arima = np.sqrt(mean_squared_error(test, predictions_arima))

print(f'N-BEATS MAPE: {mape_nbeats}, RMSE: {rmse_nbeats}')
print(f'LSTM MAPE: {mape_lstm}, RMSE: {rmse_lstm}')
print(f'ARIMA MAPE: {mape_arima}, RMSE: {rmse_arima}')
```

5.7 VISUALIZATION OF RESULTS: Visualizing the results helps in understanding the model performance and comparing the predictions with actual values. Below is the code for visualizing the results:

<CODE>

```
import matplotlib.pyplot as plt

# Plot the actual vs predicted values for N-BEATS
plt.figure(figsize=(14, 7))
plt.plot(test, color='blue', label='Actual Bitcoin Price')
plt.plot(predictions_nbeats, color='red', label='Predicted Bitcoin Price (N-
BEATS)')
plt.title('Actual vs Predicted Bitcoin Price (N-BEATS)')
plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
plt.show()

# Plot the actual vs predicted values for LSTM
plt.figure(figsize=(14, 7))
plt.plot(test, color='blue', label='Actual Bitcoin Price')
plt.plot(predictions_lstm, color='green', label='Predicted Bitcoin Price (LSTM)')
```

```
plt.title('Actual vs Predicted Bitcoin Price (LSTM)')
plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
plt.show()

# Plot the actual vs predicted values for ARIMA
plt.figure(figsize=(14, 7))
plt.plot(test, color='blue', label='Actual Bitcoin Price')
plt.plot(predictions_arima, color='orange', label='Predicted Bitcoin Price
(ARIMA)')
plt.title('Actual vs Predicted Bitcoin Price (ARIMA)')
plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
plt.show()
```

6. RESULTS AND DISCUSSION

Figure 4-11: ARIMA Hourly Forecast

Figure 4-7: LSTM Daily Forecast

Figure 4-8: ARIMA Daily Forecast

Figure 4-9: N-BEATS Hourly Forecast

Figure 4-10: LSTM Hourly Forecast

Figure 4-14: ARIMA Minute Forecast

Figure 4-13: LSTM Minute Forecast

Figure 4-12: N-BEATS Minute Forecast

Figure 4-13: LSTM Minute Forecast

7. CONCLUSION

This research is the first paper that introduces the N-BEATS time series forecasting deep learning model trained on the Bitcoin data. The developed model shows promising results, achieving a MAPE of 2.261% on daily data, a MAPE of 0.388% on hourly data, and a MAPE of 0.096% on up-to-the-minute-data. The results slightly surpass the results of an ARIMA model and are significantly better than LSTM model's results. The developed model can be used by financial analysts in financial time series forecasting, risk assessment, and modeling.

7.1 FUTURE WORK

Many improvements and changes could be made in future implementations of the N-BEATS time series forecasting machine learning model.

- First of all, the limitations outlined in the findings chapter can be avoided in future works.
- This study used historical 'Open' price data for model training. It would also be interesting to train the model on the Bitcoin returns data, which depicts the money made or lost on an investment over some time.
- The profitability analysis of the model for investment decision making can be implemented in future works.
- The model should be deployed and tested on the real-time market data.
- This paper forecasted the future price values of the Bitcoin, which is a regression problem. It would also be interesting to see the model's performance on the classification task, i.e., forecasting the stock's future downward/upward direction.

BIBLIOGRAPHY

[1] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. International Conference on Learning Representations, 2020.

[2] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Cryptography Mailing list at https://metzdowd.com, 03 2009.

[3] Cryptocurrency market capitalizations.

[4] V Dimitrova, M Fernandez-Martinez, MA Sanchez-Granero, and JE Trinidad Segovia. Some comments on bitcoin market (in) efficiency. PloS one, 14(7), 2019.

[5] Vladimir Puzyrev. Deep convolutional autoencoder for cryptocurrency market analysis. arXiv preprint arXiv:1910.12281, 2019.

[6] Anne Haubo Dyhrberg. Hedging capabilities of bitcoin. is it the virtual gold? Finance Research Letters, 16:139–144, 2016.

[7] Elie Bouri, Peter Moln´ar, Georges Azzi, David Roubaud, and Lars Ivar Hagfors. On the hedge and safe haven properties of bitcoin: Is it really more than a diversifier? Finance Research Letters, 20:192–198, 2017.

[8] David Yermack. Is bitcoin a real currency? an economic appraisal. Handbook of Digital Currency, May 2015.

[9] Khaled Guesmi, Samir Saadi, Ilyes Abid, and Zied Ftiti. Portfolio diversification with virtual currency: Evidence from bitcoin. International Review of Financial Analysis, 63:431–437, 2019.

[10] Baohua Wang, Hejiao Huang, and Xiaolong Wang. A novel text mining approach to financial time series forecasting. Neurocomputing, 83:136–145, 2012.

[11] Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PloS one, 12(7), 2017.

Enhancing Bitcoin Price Prediction with N Beats Machine Learning Model

Introduction

In this presentation, we will explore how the N Beats *Machine Learning* model can enhance **Bitcoin** price prediction. We will delve into the potential of this model and its impact on the cryptocurrency market.

Understanding N Beats Model

The N Beats model is a state-of-the-art **time series forecasting** model that can capture complex patterns and non-linear trends in data. Its ability to handle multiple seasonalities makes it ideal for cryptocurrency price prediction.

Bitcoin Price Prediction Challenges

The **volatility** and unpredictability of Bitcoin prices pose significant challenges for accurate forecasting. Traditional models often struggle to capture the rapid fluctuations and irregular patterns in cryptocurrency markets.

Benefits of N Beats Model

The N Beats model offers superior performance in capturing the intricate dynamics of Bitcoin prices. Its adaptability to varying time horizons and robustness against outliers make it a compelling choice for accurate **prediction**.

Implementation and Data Requirements

Implementing the N Beats model for Bitcoin price prediction requires historical **price data**, market indicators, and relevant **macroeconomic factors**. Proper data preprocessing and feature engineering are crucial for optimal model performance.

Evaluation Metrics

To assess the performance of the N Beats model, we will utilize **evaluation metrics** such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). These metrics provide insights into the model's accuracy and reliability.

Case Study: Bitcoin Price Prediction

We will present a case study showcasing the application of the N Beats model for Bitcoin price prediction. The study will highlight the model's ability to capture **trends**, seasonality, and short-term fluctuations in Bitcoin prices.

Conclusion

In conclusion, the N Beats machine learning model offers a promising approach to enhancing Bitcoin price prediction. Its adaptability to complex market dynamics and ability to capture irregular patterns make it a valuable tool for cryptocurrency traders and investors.

Thanks!

Sandhya S