Satvik

Project_File_format satvik.pdf

Bennett University

Document Details

Submission ID

trn:oid:::29034:93542103

Submission Date

Apr 29, 2025, 10:18 PM GMT+5:30

Download Date

Apr 29, 2025, 10:22 PM GMT+5:30

File Name

Project_File_format satvik.pdf

File Size

271.8 KB

10 Pages

3,180 Words

17,715 Characters

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 14 words)

Match Groups

- **7** Not Cited or Quoted 5%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

3%	Internet sources
1%	Publications
5%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **7** Not Cited or Quoted 5%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

3% Internet sources
1% Publications
5% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Submitted works	
	Sunway Education Group on 2023-12-11	1%
2	Submitted works	
	University of Leeds on 2024-09-05	1%
3	Submitted works	
	Cranfield University on 2023-06-02	<1%
4	Submitted works	
	Jacobs University, Bremen on 2024-12-04	<1%
5	Internet	
	www.fastercapital.com	<1%
6	Submitted works	
	University of Central Lancashire on 2024-08-08	<1%
7	Internet	
	mdpi-res.com	<1%

Bitcoin Price Prediction using Hybrid Time Series and Deep Learning Models

School of Computer Science Engineering and Technology

Bennett University
Greater Noida, Uttar Pradesh

Submitted by:
Veni Madhav, E22CSEU0487
Satvik Ranjan, E22CSEU0484
Shikhar Srivastava, E22CSEU0492

Submitted To:
Arun Singh Bhadwal

Introduction and Objective

Cryptocurrencies have gained immense popularity over the past few years, with Bitcoin (BTC) leading the charge as the most recognized and traded virtual currency. Unlike traditional financial assets, Bitcoin is found in a decentralized environment, making it highly volatile and unstable.Its price is set by the intricate interaction of market mood, world events, demand and supply, speculations on the part of investors, words of regulatory bodies, and social media biases. This volatile pattern is both the challenge and the opportunity for data-driven forecasting.

Our objective within this project is to solve the problem of future value prediction of Bitcoin with traditional time series methods and deep learning. The main objective is to come up with a hybrid model for prediction that is superior to single models by effectively capturing the linear trends and the non-linear patterns in the history of Bitcoin.

Bitcoin price prediction has various real-world applications such as:

- Investor decision-making
- Helping algorithmic and high-frequency trading platforms
- Improving financial risk management
- Enabling financial forecasting research and AI

Research Question

Does combining classical time series models (like ARIMA and SARIMA) with deep learning models (like LSTM and BiLSTM) in a hybrid manner result in more accurate and reliable predictions of Bitcoin's price than applying these models independently?

This question forms the focus of our inquiry, seeking to bridge the gap between financial forecasting and neural network models and statistical techniques.

Problem Statement

Classic statistical forecasting techniques such as ARIMA are convenient for detecting and modeling linear patterns in stationary time series data. They tend to fail, though, when confronted with real data such as the prices of Bitcoin, which tend to be noisy, non-stationary, and subject to non-linear relations. Conversely, deep learning models such as LSTM and BiLSTM have the ability to learn intricate patterns in data but can be computationally expensive, data-hungry, and at times behave like "black boxes" with low interpretability.

This article proposes the solution of uniting the two forms of the model into a hybrid pipeline such that the hybrid model will take advantage of both worlds' strength: the interpretability and ability to capture seasonality of vintage models, as well as the adaptability and ability to learn patterns of neural networks.

The project applies these models on historical day-wise Bitcoin data, compares their performances based on different error metrics, and visually and statistically compares their predictive capabilities. The final aim is to build a system which predicts accurately as well as is resilient, natural, and adaptive in future developmental and implementation processes.

II. Data Collection and Preprocessing

- **Dataset Description:**

  The data used in this project is the daily historical data of the cryptocurrency - Bitcoin (BTC) and is covering more than five years of data, taking into account various market phases such as high volatility, steep trends, and abrupt reversals. The data which is used has been accessed from public and reliable sources like CoinMarketCap or Yahoo Finance and saved in a CSV format with the filename btc_data.csv.

The dataset includes the following attributes:

2
- Date: The calendar date of each observation (formatted as YYYY-MM-DD).
- Open: The price of Bitcoin at the beginning of the trading day.
- High: The maximum price of Bitcoin that it reached within the day.
- Low: The lowest price of Bitcoin recorded during the trading session.
- Close: The price at market close, which also serves as the target variable for prediction.
- Volume: The total number of Bitcoins traded that day, expressed either in BTC or USD.

6
- Change (%): The daily percentage change in the closing price compared to the previous day.

The dataset consists of over 1800 daily entries, which provides a large enough time interval for training and testing both short term and long term forecasting models.

- **Preprocessing Steps:**

To facilitate model compatibility, performance, and accuracy, the following preprocessing steps were executed:

1. <u>Data Cleaning</u>: Non-numeric characters like commas, percentage marks, and unit suffixes (e.g., K, M, B) were stripped out of numerical columns. This helped to have more consistent data types across all features.
2. <u>Chronological Sorting</u>: The data was arranged in an ascending order targeting the 'Date' column to maintain the continuity of the temporal dependencies that is very particularly essential for time series models such as ARIMA and LSTM.
3. <u>Stationarity Transformation (for ARIMA and SARIMA)</u>: The dataset was tested for stationarity using the Augmented Dickey-Fuller (ADF) test. The original 'Close' price series proved to be non-stationary ($p > 0.05$), and therefore log transformation with subsequent first-order differencing was applied in order to stabilize the variance and the mean.
4. <u>Handling missing values</u>: The dataset is scanned for missing or null values. The missing/null values were dealt with forward fill and linear interpolation methods and the continuity of the time series data was maintained.
5. <u>Feature Scaling (for LSTM and BiLSTM)</u>: Deep learning algorithms need input values to have a comparable scale for effective gradient descent. Therefore, the 'Close' price was scaled to the range [0, 1] via MinMaxScaler from scikit-learn.
6. <u>Sequence Generation</u>: In order to convert the univariate series to a supervised learning task, sequences of 60 time steps (days) were constructed as inputs and the 61st day's

closing price as the output. This window size was determined deductively so that it can trade off between the learning context and the model complexity.

7. Train-Test Split: The data was partitioned into an 80/20 training and testing set so that the test set contains only future unseen data not encountered during training, thus simulating real-world deployment conditions.

All these preprocessing steps helped us to ensure that the model received clean. structured and consistent data. This preprocessing is very important in building a very reliable forecasting model which is capable of learning patterns in the future.

III. Time Series Modeling and Diagnostics

Model Selection and Fitting: In order to tackle the huge problem of volatile and non linear nature of the Bitcoin prices, a diverse modeling approach was adopted . The traditional time series models as well as advanced machine learning and deep learning models were utilized for the best results. The models were used to capture the different trends and patterns of the data.

- **ARIMA (Auto-Regressive Integrated Moving Average):**

The ARIMA model was used as an initial statistical base for the Bitcoin price predictions. Because of the non stationary nature of the data, the Augmented Dickey-Fuller (ADF) test was used, which confirmed the non-stationarity with a p-value of 0.271. First order differencing was applied to convert the non stationarity into stationarity. The best specification, ARIMA(4,1,5), was determined using AIC minimization. Although the model did a good job of modeling short-term linear relationships, it could not model seasonality or respond to non-linear trends and thus was better suited for short-horizon forecasting.

- **SARIMA (Seasonal ARIMA)**

SARIMA model is used to address the shortcomings of the ARIMA model. Seasonal decomposition of the series identified regular weekly cycles, and seasonal terms were incorporated into the model. Using the automatic parameter selection of the pmdarima library, we used the SARIMA .. The model handled trends and seasonality very effectively. The results show that the SARIMA model is better if the data has seasonality in it.

- **LSTM (Long Short-Term Memory)**

For efficiently incorporating the non-linear and temporal relationships of the data, an LSTM neural network was engineered. Training was carried out using sequences of 60 previous timesteps, with scaled data through MinMaxScaler for better convergence. The architecture included an LSTM layer with 50 units and a fully connected output layer. The network was trained for 50 epochs with the Adam optimizer, with early stopping to avert overfitting. On the test set, LSTM achieved an RMSE of around 2133 and an R^2 value of 0.9933. It showed excellent forecasting performance,therefore outperforming traditional time series models by a considerable margin.

- **BiLSTM (Bidirectional LSTM)**

To make the prediction better, a Bidirectional LSTM (BiLSTM) model was developed to build further upon temporal learning. In contrast to regular LSTMs that learn patterns in just one forward direction, BiLSTMs handle input sequences both forward and backward, enabling the model to learn context from both future and past data. The BiLSTM architecture included two stacked Bidirectional LSTM layers (each with 50 units) followed by a Dense output layer for regression or binary classification. The network was also trained for 50 epochs with Adam optimizer, The BiLSTM performed better than the baseline LSTM, with an RMSE of 3909 and an R^2 value of 0.9775. Its capacity to learn more complex temporal features made it especially good at predicting sudden price reversals and trend changes.

- **Ensemble Tree-Based Model (Linear Regression + Decision Tree + XGBoost)**

An ensemble stack model was constructed by combining Linear Regression, Decision Tree and XGBoost. All these models were trained on statistics such as lagged prices and rolling statistics. A weighted ensemble was created based on individual parameters. Each component had its each unique ability like - XGBoost's ability to handle non-linearity, Decision Tree's interpretability, and Linear Regression's simplicity, This model showed moderate performance. It achieved an RMSE of about $2.2e^{-10}$ with an R^2 score of 1.0, providing a competitive but less reliable solution than the deep learning models.

- **Hybrid Model (ARIMA + LSTM)**

To combine the benefits of both ARIMA as well as the LSTM model, a hybrid model was created. ARIMA captured the linear parts of the price series, while LSTM was trained on the residuals to detect non-linear patterns. Two-stage modeling provided decomposition of signal components such that the resultant forecast was derived from the sum of ARIMA and LSTM. The hybrid model exhibited moderate performances, with the lowest RMSE of 15607 and the highest R^2 score of 0.64 among all models considered. It succeeded in balancing predictive accuracy and interpretability.

- **Hybrid Model (BiLSTM + LSTM)**

To overcome all the previous drawbacks in temporal patterns in Bitcoin Price fluctuations, a model consisting of BiLSTM and LSTM was created. The LSTM layer was employed for extracting forward-dependent sequential patterns and the BiLSTM layer, which extracted bidirectional dependencies through processing the sequence in both directions (past and future).The model was able to capture import trends and signals within the time series model. The hybrid model was trained on the closing prices of Bitcoin for over a 60 days window. The architecture consists of involving dropout layers for regularizing and using the Adam optimizer. The Mean Squared Error (MSE) was used as the loss function to be optimized for improving forecasting accuracy. The stacked model of BiLSTM - LSTM showed great performance and achieved a RMSE of 2320 and a R2 score of 0.9921, surpassing individual deep learning-based models. Its capability to understand both directional and contextual patterns makes it a leading contender for high-frequency trading and short-term investment prediction scenarios.

Forecasting and Evaluation

Forecasting

Forecasting is a vital component of any time series analysis, particularly when it is applied to financial data such as cryptocurrency prices. In this research, our aim was to predict future closing prices of Bitcoin by training a number of time series models and measuring their forecasting performance. As a result of Bitcoin's nature of volatility, non-linearity, and sensitivity to outside market conditions, the task presented stern challenges that called for classical and deep learning methods.

Model Implementation Overview

Evaluation

To quantify the performance and effectiveness of our models, we used a number of commonly known evaluation measures. These measures enabled absolute error comparison alongside knowledge of the reliability of the statistical models.

5

1. Mean Absolute Error (MAE):

MAE estimates the average absolute difference between the actual value and the predicted value. It provides a direct measure of how far off, on average, the model is from the actual value. Because it assigns equal weight to all errors, MAE is resistant to outliers and ideal for general magnitude of error estimation.

2. Mean Squared Error (MSE):

MSE, whereby the error is squared before finding the average, punishes more the larger of the deviations. It therefore gets its use as identifying those models which at some times have extremely high forecasting errors, and finds especial use to financial data in which sudden downturns or peaks are a costly result.

3. Root Mean Squared Error (RMSE):

By reducing the MSE to its square root, RMSE restores the unit of measurement to that of the target variable (i.e., Bitcoin price) such that it is easy to understand in monetary terms. RMSE is a suitable measure for understanding the size and uniformity of the errors of prediction.

4. R-squared Score (R^2 Score):

This is used to determine how much variance in actual values a model can explain. A result close to 1 means the model can account for most of the variation in the data. In time series, where volatility and noise are high, R^2 assists in evaluating relative predictive power of a model.

Model Comparison and Insights

The ARIMA model was not able to capture intricate patterns of prices and created smoothed trends that were incapable of anticipating abrupt market reversals. Its residuals were autocorrelated and heteroskedastic, and therefore, the model did not possess low explanatory power for this data. Nevertheless, ARIMA.

SARIMA, although more effective at modeling seasonality, was still not flexible enough to accommodate high-volatility setups such as cryptocurrency trading. It performed somewhat better than ARIMA on seasonal sub-segments but could not follow sudden price reversals.

LSTM performed extremely well at modeling both trend and volatility. It followed quick price swings well and exhibited little lag in predicting turning points. The predictive curve of the model closely followed actual market action when validated.

BiLSTM outperformed all the other models based on forecast precision. Through use of context in both directions when training, it acquired richer representations of the behavior of prices so that its own predictions were significantly closer to true values for the majority of time intervals.

Conclusion of Evaluation

Based on the estimated metrics of evaluation, it was clear that deep learning models (BiLSTM) performed better than conventional statistical models in forecasting Bitcoin prices. Not only did the models improve qualitatively (as indicated by lower MAE, RMSE) but were also more responsive to market movement.

In the future, inclusion of external variables like volume of trade, social sentiment scores, or macroeconomic variables could also prove useful to more accurate forecasting. In addition, ensembling for averaging out multiple model predictions can help in the removal of short-term price noise with the identification of long-term trends.

V. Discussion and Conclusion

- **Results Summary:**
    - Interpret the key findings of the project.
    - Discuss the implications of your forecasting and any limitations of your model.

MODEL	MAE	MSE	RMSE	R^2 Score
LSTM	1626.0744	4,551,551.5897	2133.4366	0.9933
Avg_LSTM_BiLSTM	1516.2771	5,385,882.5467	2320.7504	0.9921
Stacked	2061.8138	9,402,099.0755	3066.2842	0.9860
BiLSTM	2645.0914	15,280,974.7910	3909.0887	0.9775
Avg_ARIMA_BiLSTM	12634.1213	243,600,813.4048	15607.7165	0.6412
ARIMA	26156.7956	1,097,078,156.0310	33122.1702	0.6140
SARIMA	26156.7956	1,097,078,156.0310	33122.1702	0.6140

Forecasting

Forecasting is a vital component of any time series analysis, especially in the domain of financial markets such as cryptocurrencies. In this study, we focused on predicting the future closing prices of Bitcoin, a highly volatile digital asset. Due to its unpredictable nature, modeling Bitcoin requires techniques that can handle non-linearity, abrupt fluctuations, and temporal dependencies. To address these challenges, we trained a variety of models—ranging from traditional statistical methods like ARIMA and SARIMA to advanced deep learning architectures like LSTM, BiLSTM, and hybrid approaches.

Model Implementation Overview

Each model was trained using historical Bitcoin closing prices. Deep learning models such as LSTM, BiLSTM, and their combinations were optimized using appropriate sequence lengths and activation functions. Classical models like ARIMA and SARIMA were tuned based on their AIC and residual analysis. Hybrid models such as Avg_LSTM_BiLSTM and Avg_ARIMA_BiLSTM were designed to leverage the strengths of both deep learning and statistical methods.

Evaluation

To assess the performance and reliability of each model, we used the following evaluation metrics:

- **Mean Absolute Error (MAE):** Measures the average magnitude of errors in a set of predictions, without considering their direction. Lower values indicate better accuracy.
- **Mean Squared Error (MSE):** Captures the average of the squared differences between actual and predicted values. It penalizes larger errors more heavily.
- **Root Mean Squared Error (RMSE):** The square root of MSE, offering a more interpretable metric in the same units as the target variable.
- **R-squared Score (R^2):** Represents the proportion of the variance in the dependent variable that is predictable from the independent variables. Closer to 1 indicates better performance.

Model Comparison and Insights

A detailed comparison of the models yielded the following insights:

- **LSTM** achieved excellent performance with a **MAE of 1626.07**, **RMSE of 2133.44**, and a **high R^2 of 0.9933**, reflecting its strong ability to capture both trend and volatility.
- **Avg_LSTM_BiLSTM**, an ensemble of two deep learning models, slightly improved upon LSTM with a **lower MAE of 1516.28** but a marginally higher RMSE (**2320.75**) and R^2 (**0.9921**). This indicates a good average representation of price behavior while slightly sacrificing consistency in error size.
- The **Stacked model** also performed well with **MAE = 2061.81** and **R^2 = 0.9860**, suggesting it learned complex feature representations but was not as accurate as the standalone LSTM in minimizing errors.
- Surprisingly, **BiLSTM**, while having the theoretical advantage of learning in both time directions, recorded **higher errors (MAE = 2645.09, RMSE = 3909.09)** and a lower **R^2 = 0.9775**, potentially due to overfitting or suboptimal training settings.

- **Avg_ARIMA_BiLSTM** showed weaker performance with **MAE = 12634.12**, **RMSE = 15607.72**, and **R^2 = 0.6412**, highlighting that mixing deep learning with traditional methods does not always lead to improved results, especially when the base statistical model is poorly suited for high volatility data.
- Both **ARIMA** and **SARIMA** failed to capture Bitcoin's price dynamics, with **very high MAE (~26156.80)**, **RMSE (~33122.17)**, and even a **negative R^2 score (-0.6140)**, indicating that these models performed worse than simply predicting the mean of the dataset.

Conclusion of Evaluation

The evaluation metrics clearly demonstrate that **deep learning models significantly outperformed traditional statistical approaches** in forecasting Bitcoin prices. Models like **LSTM and Avg_LSTM_BiLSTM** not only reduced prediction error but also maintained higher explanatory power (as shown by R^2 values). In contrast, **ARIMA and SARIMA**, which rely on assumptions of linearity and stationarity, proved inadequate for capturing the complex, nonlinear, and volatile patterns of cryptocurrency markets.

Going forward, integrating exogenous factors such as trading volume, global economic indicators, or even sentiment analysis from social media could enhance the predictive power of deep learning models. Additionally, advanced ensembling techniques and attention-based architectures could further improve both accuracy and interpretability of the forecasts.