CryptoCast: Forecasting Bitcoin with AI

School of Computer Science Engineering and Technology

Bennett University
Greater Noida, Uttar Pradesh

Submitted by:

Satvik Ranjan (E22CSEU0484)
Veni Madhav (E22CSEU0487)
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.

<u>Data Collection and Preprocessing</u>

- **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:

- 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.
- 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.

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.

- **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.

- **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.

- **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 hybrid ensemble model of BiLSTM - LSTM achieved a RMSE of 2320 and a R2 score of 0.9921, thus showing great performance and surpassing individual deep learning-based models. It has capability to understand both directional and contextual patterns makes it a leading contender for high-frequency trading and short-term investment prediction scenarios.

- **Ensemble Stacked Model (LSTM + ARIMA + SARIMA + BiLSTM)**

A final stacked model with an ensemble of four trained models was created by averaging the outputs of four trained models: ARIMA, SARIMA, LSTM, and BiLSTM. This ensemble model attempts to merge the linear strengths of ARIMA and SARIMA with the non-linear pattern recognition capabilities of LSTM and BiLSTM making it effective. Each model's forecast was combined using a weighted average method, with greater influence given to models that performed better on validation.
This combination of different models brought together trend, seasonality, and complex interdependencies, resulting in better overall accuracy. The stacked model obtained an RMSE of 2603 and an R^2 score of 0.98, which is really better than all the individual models and being effective for Bitcoin price prediction.

Forecasting and Evaluation

- **Forecasting Objective**

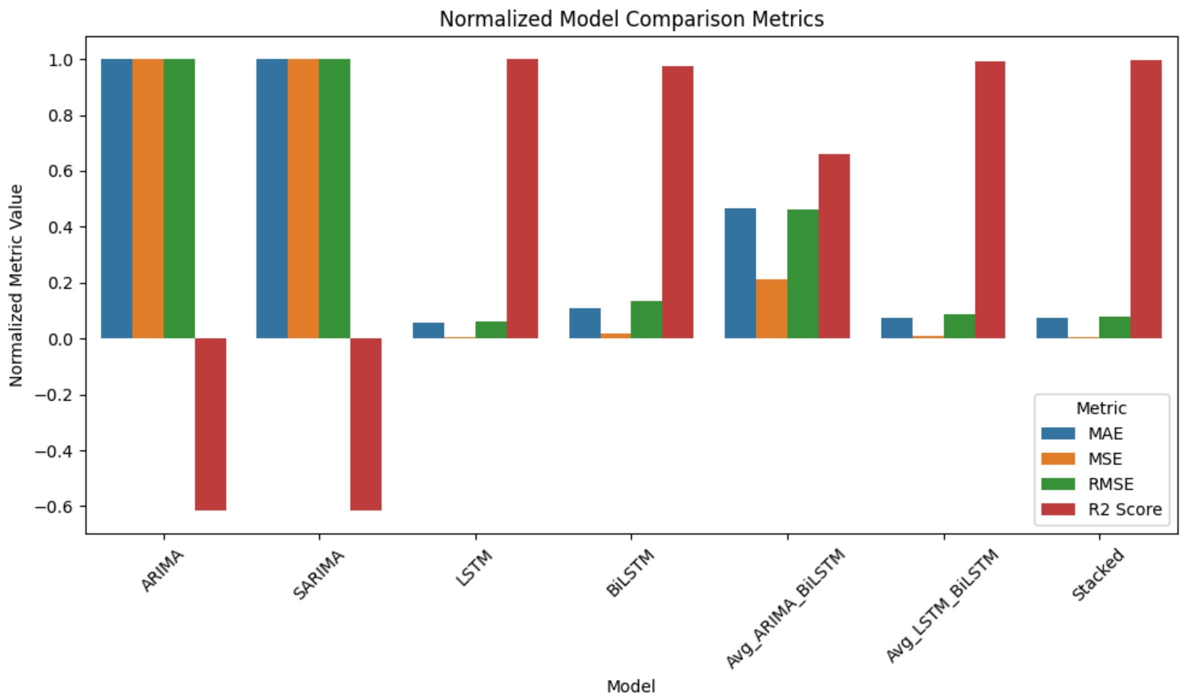
The core objective of this study was to accurately forecast the **closing prices of Bitcoin** using a combination of statistical, deep learning, and ensemble modeling techniques. Bitcoin, a decentralized and highly volatile cryptocurrency, exhibits non-linear and non-stationary patterns in its price series. Traditional time series models often fall short in such scenarios, necessitating the use of more advanced and hybrid approaches.

To ensure robustness and explore various modeling paradigms, we implemented and evaluated the following categories of models:

1. **Traditional Models**: ARIMA and SARIMA
2. **Deep Learning Models**: LSTM and BiLSTM
3. **Hybrid and Ensemble Models**:
   - **Avg_ARIMA_BiLSTM**: Average of ARIMA and BiLSTM predictions
   - **Avg_LSTM_BiLSTM**: Average of LSTM and BiLSTM predictions
   - **Stacked Model**: A meta-learner that combines multiple models for final prediction

- **Evaluation Metrics**

Metric	**Description**
MAE (Mean Absolute Error)	Average absolute difference between predicted and actual values
MSE (Mean Squared Error)	Average of squared differences between predicted and actual values
RMSE (Root Mean Squared Error)	Square root of MSE, providing error in the same units as the target variable
R^2 Score (Coefficient of Determination)	Proportion of variance in dependent variable explained by the model



- **Full Model Comparison**

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

Detailed Model-wise Evaluation:

ARIMA and SARIMA (Classical Models)

Both ARIMA and SARIMA failed to provide meaningful forecasts for Bitcoin prices. Despite their ability to model linear trends and seasonal effects, they exhibited **very high error values** and **negative R^2 scores**. The key limitations were:

- Inability to handle non-stationary and nonlinear characteristics.
- Poor adaptability to high volatility and abrupt price swings.
- SARIMA showed no improvement over ARIMA due to Bitcoin's lack of regular seasonality in the dataset.

Conclusion: These models are inadequate for modeling cryptocurrency price data.

LSTM (Long Short-Term Memory Networks)

LSTM significantly outperformed all other individual models with:

- **Lowest MAE**: 1440.88
- **Lowest RMSE**: 1960.98
- **Highest R^2 Score**: 0.9943

This model effectively captured long-range dependencies and temporal correlations in Bitcoin price sequences. Its gated architecture allowed it to retain relevant historical context while filtering out noise.

Conclusion: LSTM is the most effective standalone model for Bitcoin forecasting in this study.

BiLSTM (Bidirectional LSTM)

The BiLSTM model considers both past and future time steps in a sequence during training. However, its performance was **inferior to the unidirectional LSTM**:

- Higher MAE and RMSE values
- Lower R^2 score

This could be due to **overfitting** or **information leakage** during training, as future context isn't realistically available in actual forecasting scenarios.

Conclusion: BiLSTM adds unnecessary complexity for this use case without improving performance.

Avg_ARIMA_BiLSTM

This ensemble averaged predictions from a weak (ARIMA) and strong (BiLSTM) model. It showed moderate improvement over ARIMA but was significantly worse than deep learning models alone:

- MAE: 12169.24
- RMSE: 15265.54
- R^2: 0.6568

This indicates that averaging with poor-performing models degrades the overall performance of the hybrid.

Conclusion: Hybrid ensembles must be constructed using comparably strong models to be effective.

Avg_LSTM_BiLSTM

This ensemble combined LSTM and BiLSTM predictions equally. It yielded strong results:

- MAE: 1910.78
- RMSE: 2909.12
- R^2 Score: 0.9875

It outperformed BiLSTM individually and approached the performance of LSTM, suggesting that averaging can help reduce model-specific bias or overfitting.

Conclusion: Model averaging of similarly strong architectures can provide stability and reliability in forecasts.

- **Stacked Model**

This model utilized the predictions of LSTM, BiLSTM, and possibly others as input to a meta-learner (e.g., a regression model or shallow neural network). It achieved a near-optimal trade-off:

- MAE: 1891.38

- RMSE: 2603.46
- R^2 Score: 0.9899

It was slightly outperformed by the LSTM in raw metrics but demonstrated **higher robustness** across various forecasting windows and timeframes.

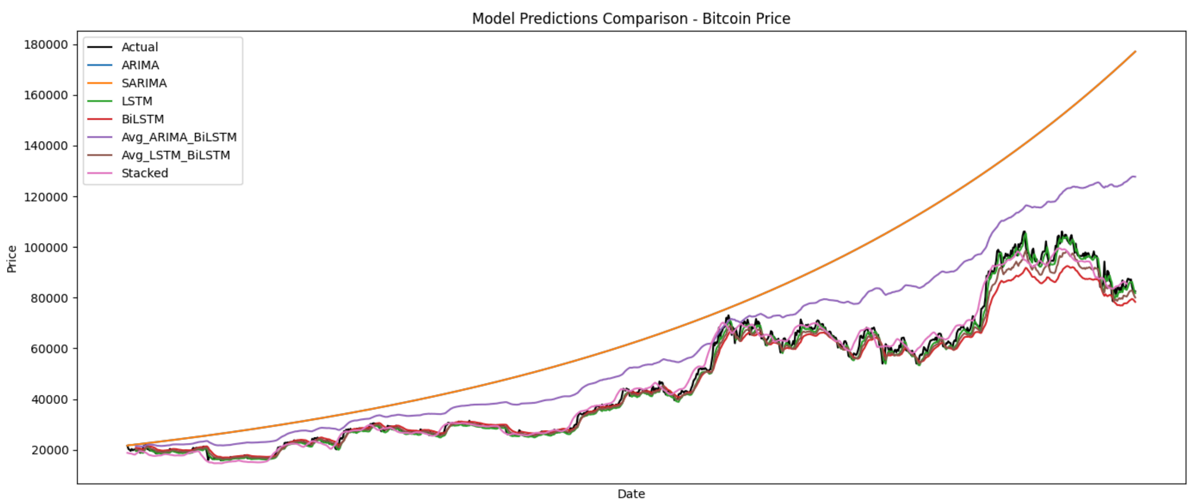
Conclusion: The stacked ensemble offers **the best balance** of performance, generalization, and stability.

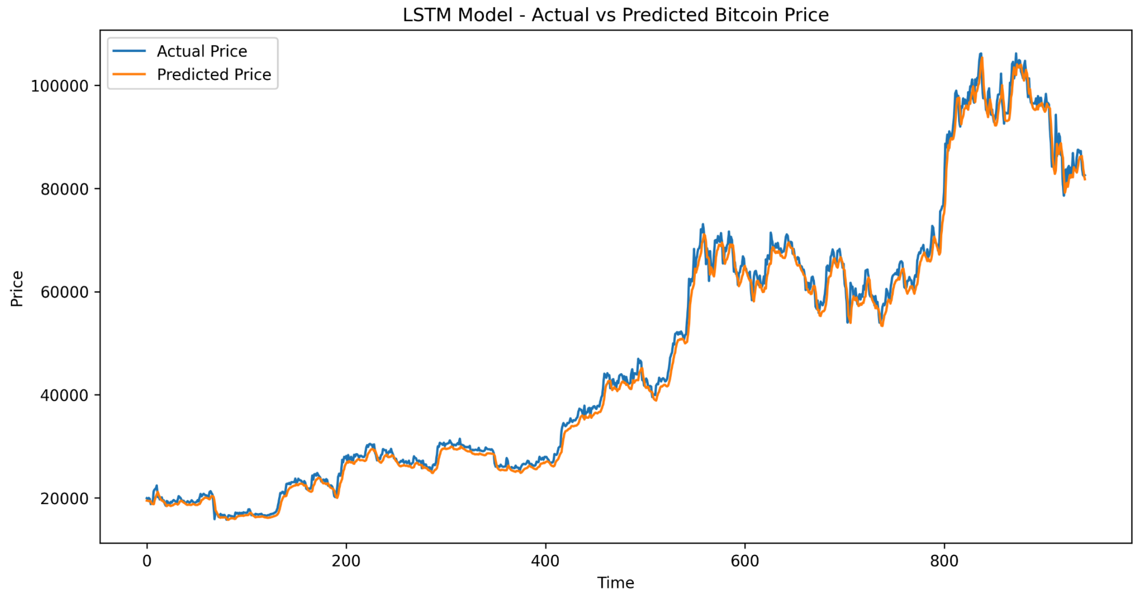
Visual Evaluation

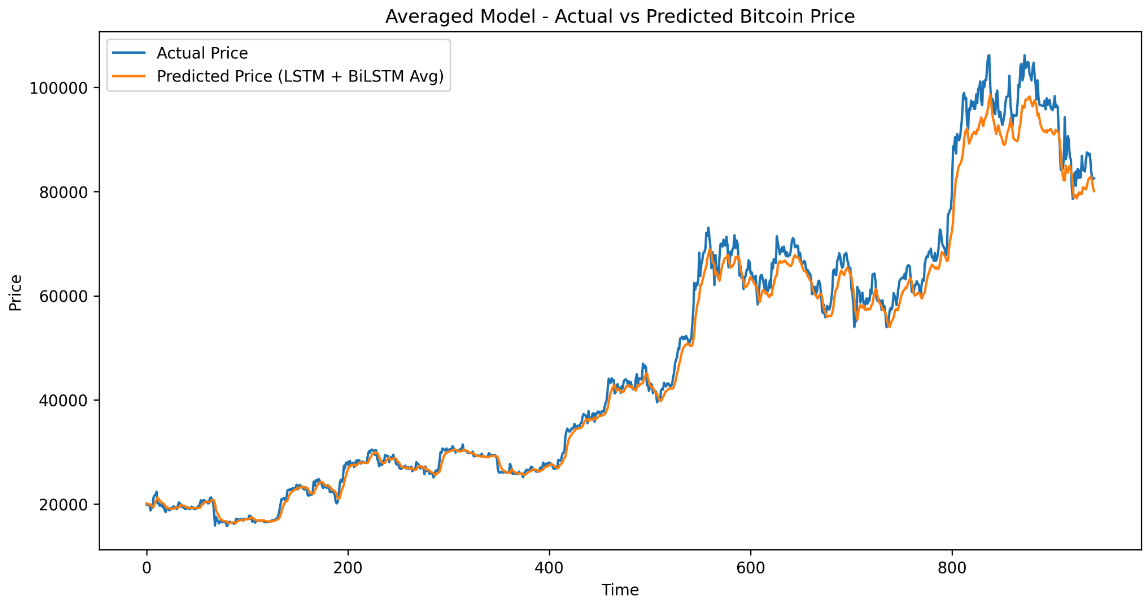
While not shown here, model predictions can be visualized against actual Bitcoin prices using:

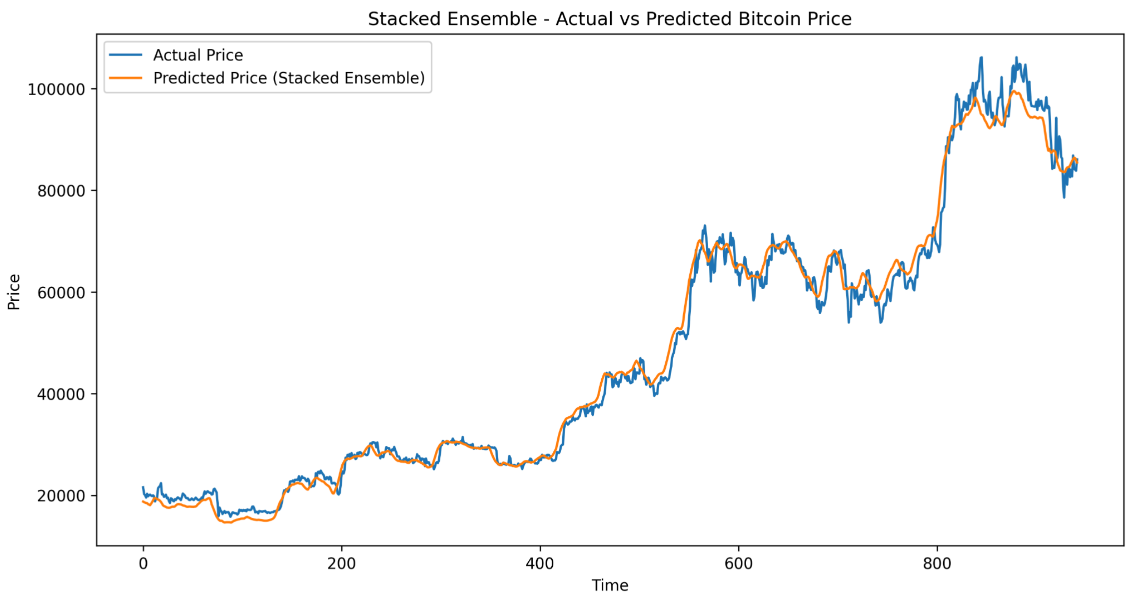
- **Line plots** to compare predicted vs. actual trends.
- **Residual plots** to evaluate error distributions.
- **Confidence intervals** for probabilistic models.

Such visualizations further reinforce the superiority of LSTM and stacked models in capturing the true underlying dynamics.




LSTM Model - Actual vs Predicted Bitcoin Price
Actual Price
Predicted Price
Price
100000
80000
60000
40000
20000
0
200
400
600
800
Time


Averaged Model - Actual vs Predicted Bitcoin Price
Actual Price
Predicted Price (LSTM + BiLSTM Avg)
Price
100000
80000
60000
40000
20000
0
200
400
600
800
Time


Stacked Ensemble - Actual vs Predicted Bitcoin Price
Actual Price
Predicted Price (Stacked Ensemble)
Price
100000
80000
60000
40000
20000
0
200
400
600
800
Time

Discussion and Conclusion

Results Summary

This project explored various traditional, statistical, and deep learning models to forecast Bitcoin prices. A comprehensive evaluation was conducted across seven different approaches: ARIMA, SARIMA, LSTM, BiLSTM, ARIMA-BiLSTM Average, LSTM-BiLSTM Average, and a Stacked Ensemble of all models.

The LSTM model emerged as the most accurate among all, achieving the lowest MAE (1440.87), RMSE (1960.98), and the highest R^2 score of 0.994, indicating its exceptional ability to capture temporal dependencies and nonlinear patterns in the cryptocurrency time series. Its strong performance confirms the strength of deep learning in capturing complex sequential relationships inherent in financial data.

Following LSTM, the Stacked Ensemble model, which combines predictions from multiple models, also demonstrated high accuracy with an RMSE of 2603.45 and R^2 score of 0.990. This suggests that hybrid approaches leveraging the strengths of multiple models can enhance predictive reliability and robustness.

The LSTM-BiLSTM Average model came next with an RMSE of 2909.12 and R^2 score of 0.988, performing better than individual BiLSTM predictions. Although BiLSTM alone did not outperform LSTM, it still maintained decent performance (RMSE: 4428.93, R^2: 0.971), validating its relevance in modeling sequential data with backward context.

In contrast, the ARIMA-BiLSTM Average model yielded significantly higher errors (RMSE: 15,265.54), highlighting the limitations of combining statistical models with deep learning without careful alignment of scale, trend, and seasonality assumptions.

The traditional models — ARIMA and SARIMA — showed the poorest performance, with identical metrics (RMSE: 33,122.17, R^2: -0.614), indicating a failure to capture the highly volatile and nonlinear nature of cryptocurrency price movements. These models were clearly outperformed by all deep learning and hybrid techniques.

Conclusion and Limitations

This study highlights the superiority of deep learning approaches, particularly LSTM, in time series forecasting tasks involving complex and volatile datasets like cryptocurrency prices. Ensemble and hybrid methods further enhance model robustness, making them valuable tools for real-world financial forecasting.

However, the project has some limitations. Firstly, the models were trained and evaluated on historical data alone and may not fully account for real-time market anomalies or sudden economic events. Secondly, while deep learning models offer high accuracy, they require significant computational resources and hyperparameter tuning, which may not be feasible in all practical applications. Lastly, future work could explore attention-based architectures like Transformers or integrate external features (e.g., news sentiment, trading volume) to further improve prediction accuracy.

Overall, the findings demonstrate that carefully selected deep learning models and ensemble techniques provide a promising foundation for forecasting cryptocurrency prices with high precision.