Enhancing Bitcoin Price Prediction: Integrating LSTM with Key Technical Indicators for Advanced Financial Forecasting

MSc Research Project
Data Analytics

Gokhan Fidan
Student ID: 22148795

School of Computing
National College of Ireland

Supervisor: Hicham Rifai

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Gokhan Fidan		
Student ID:	22148795		
Programme:	MSc in Data Analytics	**Year:**	2023
Module:	MSc Research Project		
Supervisor:	Hicham Rifai		
Submission Due Date:	14/12/2023		
Project Title:	Enhancing Bitcoin Price Prediction: Integrating LSTM with Key Technical Indicators for Advanced Financial Forecasting		
Word Count:	6111	**Page Count:**	18

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Gokhan Fidan

Date: 14/12/2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Bitcoin Price Prediction: Integrating LSTM with Key Technical Indicators for Advanced Financial Forecasting

Gokhan Fidan
22148795

Abstract

This study examines how effectively Long Short-Term Memory (LSTM) models predict Bitcoin prices when combined with particular technical indicators like Super Trend, Kaufman's Adaptive Moving Average, Fibonacci's Weighted Moving Average, and Average True Range Trailing Stop-Loss. The increasing popularity of Bitcoin as a leading cryptocurrency and the demand for precise financial forecasting instruments in this unstable market are the motivations behind this research. The study uses LSTM networks to capture the intricate, nonlinear patterns present in Bitcoin price movements using a dataset from Yahoo Finance. How these specific technical indicators are included influences the precision of Bitcoin price forecasts is the main research question. Several metrics, including Mean Absolute Error, Mean Squared Error, and the Coefficient of Determination, are used to thoroughly assess the performance of the LSTM model through cross-validation. The results show that the LSTM model's predictive accuracy is improved when the chosen technical indicators are added, underscoring the importance of these indicators in describing market dynamics. By addressing a gap in the literature about the integration of these particular indicators with LSTM models for Bitcoin price prediction, this study advances the field. It establishes the framework for upcoming studies in sophisticated predictive modelling and automated trading systems and provides traders and investors in the cryptocurrency markets with insightful information. The study also emphasizes how crucial model interpretability and transparency are in the rapidly changing field of AI-driven financial analysis.

1 Introduction

Since its launch by (Nakamoto, S., 2008), Bitcoin has become the most well-known cryptocurrency, generating a lot of attention in the technology and financial industries. Its distinctive qualities anonymity, decentralization, privacy, peer-to-peer networking, security, and resilience, among others have helped it become incredibly popular around the world and seen an exponential increase in value. Due to its complex technology and the conjecture surrounding its value, Bitcoin has drawn the attention and discussion of both supporters and detractors (Rathore, 2020). The cryptocurrency market, which includes Bitcoin, is renowned for its ability to trade continuously, as opposed to the conventional idea of daily closes. This unique characteristic of the market has an impact on investment plans and conversations about market efficiency. It has also sparked the development of new techniques for identifying market inefficiencies, such as the day-of-the-week effect. (Miralles-Quirós & Miralles-Quirós, 2022). Bitcoin is a digital currency that blurs the line between fiat and commodity currencies by running on a decentralized peer-to-peer payment network. Its independence from the government and currency management, along with its unique statistical features, set it apart from more traditional asset classes like stocks, bonds, and commodities. Due to its independence and distinctiveness, there is a continuous discussion concerning its function as an asset class or a medium of exchange (Rowland, Suler and Cajkovicova, 2021).

Deep learning algorithms have changed a number of fields, including financial forecasting. The application of these algorithms has garnered significant interest, particularly in the context of predicting Bitcoin prices. The ability of deep learning models, such as Long Short-Term Memory (LSTM) networks, to process and learn from large datasets, including complex and nonlinear financial data, makes them particularly well-suited for cryptocurrency price prediction. Using information from social media sentiment indexes and technical indicators, a novel ensemble deep learning model that integrates LSTM and Gate Recurrent Unit (GRU) networks has been proposed for predicting Bitcoin prices. When compared to traditional models, this model has demonstrated improved accuracy in near-real-time prediction (Ye et al., 2022). To predict Bitcoin prices, another study used a deep learning integration method (SDAE-B) that combined Bootstrap Aggregation (Bagging) and Stacking Denoising Autoencoders (SDAE). Compared to conventional machine learning techniques, this approach showed improved accuracy and decreased error in capturing the nonlinear and random aspects of Bitcoin prices (Zhang, Li & Yan, 2022). According to (Ji, Kim, and Im's, 2019) comparative study of deep learning techniques, including LSTM, for predicting Bitcoin prices, LSTM-based models performed marginally better in regression tasks than other models, but Deep Neural Network (DNN)-based models performed better in classification tasks associated with price movements.

Precise estimation of Bitcoin values is essential for multiple rationales. It helps traders and investors manage risk, make well-informed decisions, and gain a deeper comprehension of market dynamics. For both individual and institutional investors, precise price prediction tools are crucial given the volatility of Bitcoin and the increasing interest in cryptocurrencies.

Based on given information above, here the research question is set to as "How usage of technical indicators such as Super Trend, Kaufman's Adaptive Moving Average, Fibonacci's Weighted Moving Average, and Average True Range Trailing Stop Loss, affect BTC price prediction accuracy?". There is still a gap in the literature regarding the application of particular indicators like Super Trend, Kaufman's Adaptive Moving Average, Fibonacci's Weighted Moving Average, and Average True Range Trailing Stop-Loss indicator, despite the fact that numerous studies have used LSTM models with technical indicators for Bitcoin price prediction. In the field, integrating these indicators with LSTM models offers a novel approach that could result in predictions that are more reliable and accurate.

The primary research question of this study is: "*How effectively can LSTM models, integrated with technical indicators like Super Trend, Kaufman's Adaptive Moving Average, Fibonacci's Weighted Moving Average, and Average True Range Trailing Stop-Loss indicator, predict Bitcoin prices?*". Also, this study will focus on evaluation the effectiveness of LSTM models in predicting Bitcoin prices and examine how adding distinct technical indicators affects how accurate these forecasts are. The data to be used was downloaded from the finance yahoo website[1]. The mentioned indicators were calculated and added to the dataset. After the necessary data analysis steps are applied, the model will be subjected to LSTM. The performance metrics of the model will be calculated and interpreted.

This report's remaining sections will conduct a thorough literature review and examine previous studies on Bitcoin price prediction, with an emphasis on those that use LSTM models in conjunction with different technical indicators. In order to provide a strong basis for our own research, this review will critically analyze the approaches, conclusions, and

[1] https://finance.yahoo.com/quote/BTC-USD/history/

limitations of previous studies. The literature review will be followed by a detailed explanation of the study's methodology in the report. In addition to providing an explanation of how to calculate certain technical indicators like Super Trend, Kaufman's Adaptive Moving Average, Fibonacci's Weighted Moving Average, and Average True Range Trailing Stop-Loss indicator, it will also detail the procedure for gathering data from Yahoo Finance and explain how to apply the LSTM model. Along with covering the data analysis procedures, this section will also justify the methodology that was selected. The findings of this study, including an evaluation of how well the LSTM model predicted Bitcoin prices, will be presented in the section that follows. With a discussion of these findings in light of the body of current research, it will emphasize how the incorporation of the chosen technical indicators affects the prediction accuracy. This section's goal is to draw attention to the importance and contributions of our study. The report will conclude with a summary of the key findings and a discussion of the broader implications of our research for researchers, traders, and investors in cryptocurrencies. The limitations of our study will be discussed in this conclusion, along with possible future research directions.

2 Related Work

Many studies have been conducted on the use of deep learning algorithms, especially Long Short-Term Memory (LSTM) networks, in the financial markets. The development of technical trading systems and their combination with long short-term memory (LSTM) for price prediction in stock markets and cryptocurrencies specifically, Bitcoin are critically examined in this literature review.

2.1 Technical Analysis and Indicators

In the financial markets, technical trading systems have been fundamental. Many of the technical indicators in use today have their roots in Wilder's (1978) introduction of novel ideas to technical trading systems. These ideas were further developed by Achelis (2001), who gave a thorough synopsis of technical analysis. Chan et al. (1996) studied on Momentum strategies. This study investigates momentum strategies in the stock market. The authors explore the tendency of stocks that have performed well in the past to continue performing well in the near future, and vice versa for poorly performing stocks. The study analyses data from the NYSE and AMEX over a period of several years and finds significant returns to buying past winners and selling past losers. The integration of technical indicators with contemporary machine learning techniques has been made possible by these seminal works.

2.2 Bitcoin Price Prediction with LSTM

The use of LSTM for stock market forecasting has produced encouraging outcomes. It is illustrated by Alam et al. (2021) how well LSTM performed when employing multi-feature input variables to predict stock prices. Their research demonstrates how well LSTM handles time series data, which makes it a good option for financial forecasting. In their application of LSTM to the Indian stock market, Ghosh et al. (2019) emphasized the stochastic nature of stock prices and the potential of LSTM to produce precise forecasts. These studies highlight the usefulness of LSTM in stock market analysis by utilizing its long-term dependency memory.

Predicting prices in the cryptocurrency market is a unique challenge due to its volatility, especially with regard to Bitcoin. An early benchmarking study on cryptocurrencies was conducted by Hileman and Rauchs (2017), which laid the groundwork for later predictive analyses. The potential of long short-term memory (LSTM) in the cryptocurrency price prediction space was demonstrated by Jay et al.'s (2020) exploration of stochastic neural networks. Saad et al. (2019) and Oncharoen and Vateekul (2018) both conducted additional research on the application of deep learning for Bitcoin price prediction, highlighting the significance of LSTM in identifying intricate patterns in price fluctuations. Comparative summary can be seen in **Table 1**.

Table 1: Comparative Summary of Related Studies in BTC Price Prediction with LSTM

Reference	Technique	Outcome	Evaluation	Dataset
Alam et al.(2021)	LSTM	Stock Price Prediction	Not Applicable	National Stock Exchange
Ghosh et al. (2019)	LSTM	Stock Price Prediction	Not Applicable	Bombay Stock Exchange
Jay et al.'s (2020)	Multi-layer perceptron and LSTM	Bitcoin, Ethereum, and Litecoin Price Prediction	Not Applicable	https://bitinfocharts.com/
Saad et al. (2019)	Linear Regression, Random Forest, Gradient Boosting, LSTM	Bitcoin and Ethereum price prediction	RMSE Values LR:0.0207 RF:0.0141 GB:0.0146 LSTM:0.15	“Blockchain” Exchange API
Oncharoen and Vateekul (2018)	CNN and LSTM by using Technical Indicators and textual data	Stock market prediction	Accuracy: 69.86%	Reuters, Reddit, Intrinio

2.3 Bitcoin Price Prediction with LSTM and Technical Indicators

Recently, there has been an emphasis on integrating LSTM with technical indicators for price prediction. Shynkevich et al. (2017) looked into how changing the input window length affected the ability to use technical indicators to forecast price movements, which shed light on how to optimize LSTM models. In order to predict high-frequency trends in cryptocurrency markets, Alonso-Monsalve et al. investigated the convolution of neural networks and showed how useful it is to combine LSTM with technical indicators.

The use of LSTM and technical indicators together for Bitcoin price prediction has advanced recently thanks to research studies. To improve the model's capacity to concentrate on pertinent features for more precise forecasts, a 2021 study integrated a multiplicative LSTM with technical indicators and an attention mechanism. Ye et al. (2022) used data from social media sentiment indexes and technical indicators to propose a stacking ensemble deep learning model that combines LSTM and GRU networks. This model showed how well LSTM combined with a variety of data sources, such as technical indicators, could predict the price of Bitcoin almost instantly. For the purpose of improving the accuracy of Bitcoin price prediction, an optimized LSTM model with a modified cuckoo search optimization was suggested. Li et al. (2020) used an attentive LSTM network to study Bitcoin price fluctuation prediction. Their method demonstrated the efficacy of LSTM in capturing intricate patterns in cryptocurrency price movements by combining fundamental features, conventional technical trading indicators, and features produced by a Denoising autoencoder. Cao et al. (2022) used deep learning and technical indicators to forecast short-term trends in cryptocurrency prices. Time-series data from Bitcoin and Ethereum were used in this study to demonstrate how LSTM can be used in conjunction with technical indicators for short-term investment strategies. Ortu et al.'s (2022) deep learning study examined the effects of social media indicators and technical trading on the categorization of cryptocurrency prices. Their results highlight the importance of diverse data sources in LSTM models by showing that the addition of trading indicators to traditional technical indicators enhances the accuracy of Bitcoin and Ethereum price predictions.

Despite these advancements, the literature still lacks studies specifically exploring the combination of LSTM with technical indicators such as Super Trend, Kaufman's Adaptive Moving Average, Fibonacci's Weighted Moving Average, and Average True Range Trailing Stop-Loss indicator for Bitcoin price prediction. This gap presents a unique opportunity for this study to contribute novel insights to the field. Comparative summary can be seen in **Table 2**.

Table 2: Comparative Summary of Related Studies in BTC Price Prediction with LSTM and Technical Indicators

Reference	Technique	Outcome	Evaluation	Dataset
Shynkevich et al. (2017)	ANN, SVM, kNN	S&P 500 stock market index prediction	Accuracy: SVM: 60.89% ANN: 56.68% kNN: 41.86%	Yahoo Finance website
Ye et al. (2022)	LSTM	BTC price prediction	MSE: 184,271.5385	http://data.binance.vision/ and Twitter Search API
Li et al. (2020)	LSTM	Bitcoin Price Fluctuation Prediction	Accuracy: 57.75%	Collected from five bitcoin exchanges (i.e., Huobi, Coinbase, Binance, Bitstamp, Bitfinex)
Cao et al. (2022)	LSTM	Bitcoin and Ethereum Price Prediction	Profitability factor: BTC: 1.2364 ETH: 1.2682	Binance
Ortu et al.'s (2022)	Technical indicators and social media data to LSTM	Bitcoin and Ethereum Price Prediction	Accuracy: 67%–84%	Crypto Data Download web services and Bitfinex

3 Research Methodology

The methodology used in this study is called CRISP-DM (Cross-Industry Standard Process for Data Mining), and it is a well-known technique in predictive analytics and data mining. The CRISP-DM framework offers a systematic approach to directing research at different phases, such as comprehending the business goal, comprehending data, preparing data, modeling, assessing, and implementing the findings. Using LSTM models combined with technical indicators, the CRISP-DM methodology is specifically designed to tackle the unique problem of predicting Bitcoin prices in the context of this study. This study's main goal is to investigate how well LSTM models, when supplemented with specific technical indicators, can forecast Bitcoin prices. This entails a thorough procedure that begins with gathering and preparing historical Bitcoin price data, continues with the use of cutting-edge deep learning techniques, and ends with assessing the model's predictive ability. The study's adherence to the CRISP-DM framework guarantees a methodical and comprehensive approach, facilitating the replication and validation of findings by other investigators. This methodology adds to the validity and trustworthiness of the results in addition to making the research process easier to understand.

3.1 Data Acquisition

This study's dataset, which focuses on the historical daily prices of Bitcoin (BTC-USD), is taken from Yahoo Finance. The maximum data is available for the period of September 17, 2014, through December 5, 2023. Predictive modeling has a solid foundation thanks to this sizable dataset, which provides a comprehensive understanding of the price fluctuations of Bitcoin over a sizable period of time. The dataset includes a wide range of historical market conditions, such as times of high volatility, consistent growth, and market corrections. For a model to be trained that can effectively generalize across various market scenarios, there must be diversity in the market conditions. The daily price data is balanced in terms of computational manageability and granularity. Although it records daily changes in the market, it is not detailed enough to add too much noise to the model. Because Bitcoin data from Yahoo Finance is easily accessible and widely regarded as reliable, it is a popular choice for both practical and academic research in cryptocurrency markets. The dataset contains important price indicators that shed light on market dynamics, including volume, close, high, low, and open. These indicators are essential for applying technical analysis and for creating a more sophisticated understanding of market trends.

3.2 Feature Engineering

A crucial phase in the research process is feature engineering, which entails transforming and producing data features to improve the predictive ability of the model. In order to find trends and patterns in market data, feature engineering in this study concentrated on generating technical indicators from the Bitcoin price data. These indicators are frequently used in trading and financial analysis.

The following technical indicators were calculated and added to the dataset:

1- **Super Trend**: This indicator combines volatility and price movement. This trend-following indicator can be used to determine whether a trend is bullish or bearish. The asset's price and its average true range, a measure of volatility, were combined to create the Super Trend indicator.

2- **Kaufman's Adaptive Moving Average (KAMA)**: In response to changes in market volatility, this adaptive moving average modifies its sensitivity. When there are large price fluctuations, it responds more, and when the market is stagnant, it responds less. To reflect the ever-changing nature of the Bitcoin market, KAMA was added.

3- **Fibonacci's Weighted Moving Average (FWMA)**: This indicator, which is based on the Fibonacci sequence, gives recent prices more weight. Given that recent price changes are thought to be more significant, it is used to pinpoint possible market reversal points.

4- **Average True Range Trailing Stop-Loss Indicator**: Based on average true range (ATR) measurements of market volatility, this indicator offers a dynamic stop-loss level. By adapting to the present volatility of the market, it aids in the identification of the best trading exit points.

These indicators were chosen because they are useful in financial trading and can accurately depict a range of market behavior factors, including momentum, volatility, and trend direction. They are well-known for their usefulness in technical analysis and are frequently utilized by traders.

The feature engineering process included several crucial data preparation steps in addition to calculating technical indicators to make sure the dataset was optimally structured for the LSTM model. These steps are Handling Missing Values, Dropping Unnecessary Columns and Setting the Date as Index. The performance of the model can be greatly impacted by missing values in the dataset. Backfilling, or bfilling, was used to fill in the missing values in order to address this. The continuity and integrity of the time series data are preserved by this method, which makes sure that any gaps in the data are filled in with the next valid value that becomes available. Some columns were removed in order to simplify the dataset and concentrate on the most important features. These included ATR, 'Trailing_Stop_Long_Prev', 'Open', 'High', 'Low', 'Adj Close', 'Volume', and 'Trailing_Stop_Short_Prev'. This was an essential point to decreasing the dimensionality of the data and avoiding unnecessary data from impacting the model. ATR, Trailing_Stop_Long_Prev and Trailing_Stop_Short_Prev' columns directly coming from Average True Range Trailing Stop-Loss Indicator calculation. The column of Open, High, Low, Adj Close and Volume comes from original dataset. The datetime format of the 'Date' column was converted, and it was assigned as the Data Frame’s index. Because it makes data manipulation and analysis based on time intervals easier, this transformation is crucial for time series analysis. Additionally, it makes using pandas' other time series functionalities, such as time-based slicing, easier. These extra feature engineering steps were crucial in helping to improve the dataset for the LSTM model. The dataset was made more condensed, pertinent, and appropriate for capturing the temporal dynamics of Bitcoin prices by adding missing value fills, removing superfluous columns, and designating the date as the index. Preparing the data with great care is essential to guaranteeing the predictive model's efficacy and accuracy.

3.3 LSTM

The LSTM (Long Short-Term Memory) model, a kind of recurrent neural network (RNN) particularly skilled at processing and learning from sequential data, forms the basis of this study's predictive analysis. Because the LSTM model can effectively capture temporal dependencies and patterns in time-series data a critical skill for precise Bitcoin price prediction it has been selected in particular.

3.4 Evaluation Metrics

Metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and Coefficient of Determination (R2 Score) are used to assess the performance of the model. These measures offer a thorough understanding of the model's precision and forecasting power.
The model's performance was assessed using a variety of metrics following training. During each cross-validation fold, these metrics were computed for the training and validation sets. Matplotlib.pyplot was used to visualize the results, showing the performance metrics (MSE, RMSE, MAE and R^2 Score) across various folds as well as the learning curves (loss and validation loss). An intuitive grasp of the model's learning procedure and predictive accuracy was made possible by these visuals.

3.5 Predictions and Future Price Forecasting

Future Bitcoin prices will be predicted and predictions will be made on the test set using the trained model. This entailed adding the final data sequence to the model and projecting the predictions 30 days ahead of time for this study. Using the MinMaxScaler, the projected future prices were rescaled to the original price scale and displayed alongside the actual historical prices. As a result, comparing the model's predictions with the actual market trends was simple.

Python is the primary programming language used with libraries such as Pandas, NumPy, Matplotlib, and TensorFlow for data handling, numerical computations, plotting, and building LSTM models.

4 Design Specification

The architecture and configuration of predictive models are critical factors that determine their accuracy and efficacy in the field of financial time series prediction. This section explores the design parameters of the LSTM model used in this research, which was created especially for the purpose of predicting the price of bitcoin. Because the LSTM model can capture long-term dependencies in time series data, it is especially suitable for this task; it was first introduced by Hochreiter and Schmidhuber (1997). Optimizing the LSTM model's capacity to process and learn from Bitcoin's historical price data is the goal of the design decisions made, including how different layers are arranged and functional. Every part of the model, from the LSTM layers to the Dropout and Dense layers, is chosen and set up to improve the predictive performance of the model while resolving typical issues like overfitting and feature interpretation. A thorough description of each of these elements is given in the ensuing subsections, along with an explanation of their functions and contributions to the model's overall ability to predict Bitcoin prices.

4.1 LSTM Layers

The model has more than one LSTM layer. The input sequence data is processed by each layer of the LSTM in order to gather and store significant information over time. Understanding the long-term trends and patterns in the fluctuations of the price of Bitcoin requires this. The LSTM layers aid in the model's memory of previous data points, which is essential for making price predictions for the future based on historical trends.

4.2 Dropout Layers

In order to reduce the possibility of overfitting, the model incorporates dropout layers. A model may overfit when it learns the training set too thoroughly—including the noise and fluctuations—which can have a detrimental effect on the model's performance on newly discovered data. During training, the Dropout layers randomly deactivate a predetermined percentage of the neurons (nodes) in the network, forcing the model to acquire more resilient features that are more resilient to new data. A 20% dropout rate is employed in this study, which indicates that 20% of the neurons are arbitrarily disregarded during each training phase. This aids in developing a more versatile model that exhibits strong performance on both training and untested data.

4.3 Dense Layers

The model comprises fully connected neural network layers called Dense layers after the LSTM layers. These layers serve as the final prediction-making layers and interpret the features that the LSTM layers have learned. The data that has been processed by the LSTM layers is combined and mapped to the required output format by the Dense layers. A Dense layer with 35 units is included in the model in this study, and the final output Dense layer has one unit, which represents the anticipated price of Bitcoin.

4.4 Model Compilation

One popular option for training deep learning models is the Adam optimizer, which is used to compile the LSTM model. Financial time series and other complex datasets are well-suited for the Adam optimizer due to its effectiveness in managing sparse gradients and adjusting the learning rate during training. Mean Squared Error (MSE), a common loss function for regression tasks, was employed in this investigation. The Mean Squared Error (MSE) computes the average of the squares of the errors or deviations, or the difference between the expected and actual values, to give a clear indication of the model's prediction accuracy.

The LSTM model used in this research has been thoughtfully built to satisfy the particular requirements of bitcoin price prediction. The model is well-equipped to provide dependable and accurate predictions of Bitcoin prices by utilizing the LSTM's capacity to comprehend temporal patterns in conjunction with mechanisms to prevent overfitting and accurately interpret features.

5 Implementation

5.1 Creating Sequence and Normalization

Following feature engineering, the dataset was ready for the LSTM model. This required using the MinMaxScaler from the sklearn.preprocessing module to normalize the features. Deep learning models require normalization because it uniformizes the input feature range and promotes faster training convergence. The normalization process involved to the Trailing_Stop_Long, Trailing_Stop_Short, KAMA, SuperTrend, Close and FWMA features. Sequences are the type of input data that the LSTM model needs. As a result, the prepared dataset was split up into sequences, each of which stood for a distinct time period 60 days in this case. For the data to show the temporal dependencies, this transformation was essential. From the normalized data, these sequences were produced using a custom function called create_sequences.

The Sequential model from the tensorflow.keras.models module was used to construct the LSTM model. As previously mentioned, the model architecture consisted of Dense, Dropout, and LSTM layers.

5.2 LSTM Model Architecture

The LSTM layers in the model are made to process sequential data and retain patterns over extended periods of time. The model's 100 units per LSTM layer strikes a balance between computational efficiency and model complexity. Dropout layers, which randomly remove a percentage of the neurons (set at 20% in this study) during the training phase, are used to prevent overfitting. By doing this, it is made sure that the model is not unduly dependent on any one feature or pattern found in the training set. Dense layers are employed after the LSTM layers to interpret the features that the LSTM has learned. The model consists of a 35-unit Dense layer and a final Dense layer with an output of one unit, which represents the anticipated price. The Adam optimizer, which is well-liked for deep learning applications because of its effectiveness in managing sparse gradients and adaptable learning rates, is used to compile the model. Mean squared error (MSE) is the loss function that is employed, and it is suitable for regression tasks such as price prediction.

5.3 LSTM Model Training and Validation

As recommended by Browne (2000), a K-Fold cross-validation method is used to confirm the model's applicability and functionality also avoiding overfitting. As it is pointed out Weigend, Mangeas and Srivastava (1995), overfitting can be real problem for model efficiency. Using this method, the dataset is divided into 'k' subsets, or folds. The model is then trained on 'k-1' folds, and its validity is checked on the remaining fold. Every fold serves as the validation set once during the 'k' iterations of this process. Five-fold cross-validation is employed in this work. With a batch size of 32, the model is trained on each fold for 15 epochs. The LSTM model is trained by feeding it data sequences, including technical indicator data, and modifying the model weights to minimize the loss function.

6 Evaluation

The model's performance was evaluated using metrics like MAE, MSE, RMSE and R2 Score. These metrics provided a comprehensive understanding of the model's accuracy and predictive power. The evaluation included visualizing the performance metrics and learning curves to gain an intuitive understanding of the model's learning process and predictive accuracy.

6.1 Experiment with LSTM by feeding Technical Indicators

The main purpose of this experiment is to assess the effectiveness of an LSTM model enhanced with technical indicators in predicting Bitcoin prices. The technical indicators used were Super Trend, Kaufman's Adaptive Moving Average (KAMA), Fibonacci's Weighted Moving Average (FWMA), and Average True Range Trailing Stop Loss Indicator.

As specified in section 3.1 this study's dataset, which focuses on the historical daily prices of Bitcoin (BTC-USD), is taken from Yahoo Finance. The data is available for the period of September 17, 2014, through December 5, 2023. Also as specified in section 3.2, previously mentioned technical indicators were calculated within the scope feature engineering process. Preprocessing actions like addressing missing values, removing unnecessary columns, and designating the date as the index were also performed on the dataset. Plotting performed for Btc Closing Prices, SuperTrend, FWMA and KAMA indicators can be seen in **Figure 1**.

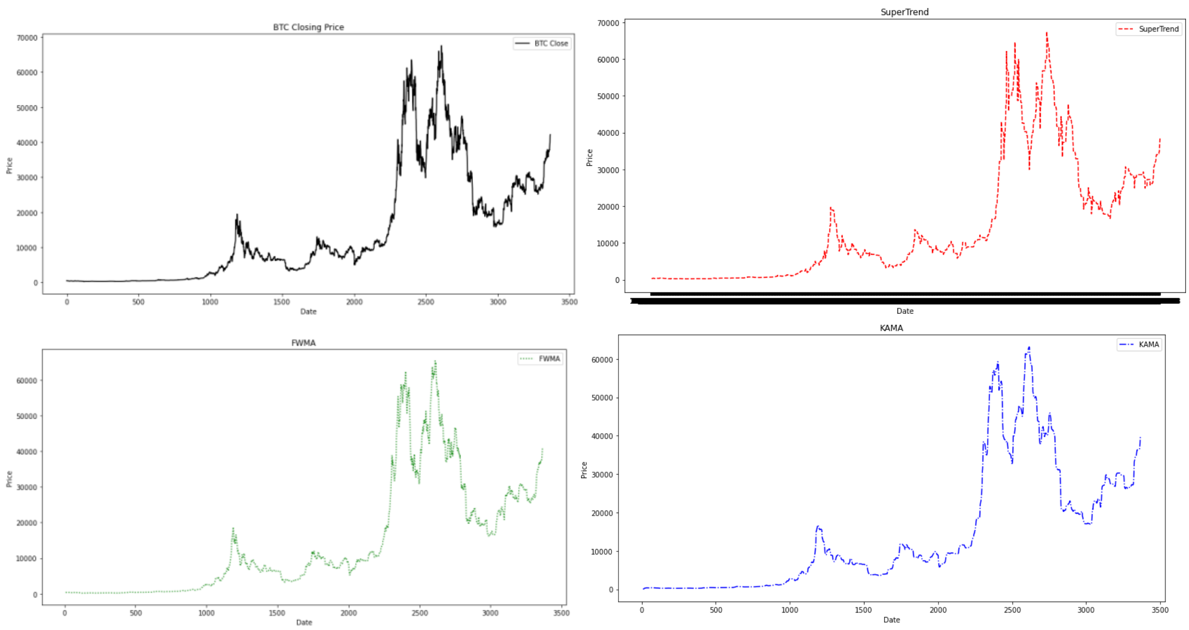

Figure 1: BTC Closing Prices, SuperTrend, KAMA and FWMA indicators plotted

The model begins with an LSTM layer of 100 units. A second LSTM layer with 75 units follows. Two Dropout layers, each set at a rate of 20%, are included to reduce the possibility of overfitting. During the training stage, these layers randomly deactivate a portion of the neurons, which helps the model learn more resilient and broadly applicable features. After the LSTM layers, the model includes a Dense layer with 35 units, employing the "relu" activation function for non-linear transformations. The final output layer is a Dense layer with a single unit, using a "linear" activation function, suitable for predicting continuous values like Bitcoin prices. The model is compiled with the Adam optimizer. The loss function used is Mean Squared Error (MSE).

A K-Fold cross-validation technique with five folds is used to train the model. By ensuring a thorough validation of the model across various data subsets, this approach improves the robustness and dependability of the outcomes. The model is validated on a different subset of data after being trained on a subset for every fold. Technical indicator data and other data sequences are fed into the model during the training process, and the model weights are adjusted to minimize the loss function. The explanation can be found belove How Cross Validation stage set?

Setting up K-Fold Cross-Validation:

‘KFold’ is a model cross-validator that divides the dataset into k consecutive folds (in this case, 5 folds). Each fold is then used once as a validation while the k - 1 remaining folds form the training set. ‘shuffle’=True ensures that the data is shuffled before splitting into batches. ‘random_state’=42 sets a seed for the random number generator that shuffles the data, ensuring reproducible splits.

Initializing Lists and DataFrame for Metrics:

‘histories’ variable will store the history object returned by the fit method of the Keras model for each fold. ‘metrics_df’ is a DataFrame to keep track of the metrics computed for each fold and set (training or validation). ’fold_predictions’ variable will keep tuples containing the validation dates, actual prices, and predicted prices.

Defining a Function to Calculate Metrics:
‘calculate_metrics’ is a utility function that calculates various performance metrics such as MSE (Mean Squared Error), RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and R^2 (Coefficient of Determination).

Training and Evaluating the Model in a K-Fold Cross-Validation Loop:

The loop iterates over each fold generated by kf.split(X), where X is the input data. ‘train_index’ and ‘test_index’ are arrays of indices for the training and validation data, respectively. The model is built using ‘build_model’ and trained using the fit method on the training set. Predictions are made on the training set to calculate training metrics and on the validation set to calculate validation metrics. Metrics for each fold are appended to the ‘metrics_df’. The predictions and actual values are inverse transformed to their original scale and stored in ‘fold_predictions’ with their corresponding dates.

Plotting Evaluation Metrics:

A number of metrics, such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the Coefficient of Determination (R2 Score), are computed during the training process to assess the performance of the model. Each fold undergoes 15 epochs of training with a batch size of 32 in order to balance computational efficiency and the likelihood that the model will learn enough from the data. These measures shed light on the model's precision and capacity for prediction. Evaluation metrics results can be seen in **Table 3**.

Table 3: Evaluation Metrics for each fold in Train and Validation Set

Fold	Set	MSE	RMSE	MAE	R2
1	Train	0.000352	0.018766	0.010465	0.993800
1	Validation	0.000320	0.017877	0.009788	0.994454
2	Train	0.001415	0.037617	0.024075	0.974639
2	Validation	0.001623	0.040281	0.025692	0.973666
3	Train	0.000299	0.017278	0.009514	0.994827
3	Validation	0.000346	0.018604	0.010065	0.993604
4	Train	0.000420	0.020505	0.011713	0.992668
4	Validation	0.000476	0.021827	0.011928	0.991427
5	Train	0.000320	0.017902	0.010685	0.994407
5	Validation	0.000283	0.016835	0.010248	0.994916

After evaluation metrics calculated and visualized using bar plots. Also, overall performance is calculated the average of each metric for both training and validation sets. The data by grouped by the 'Set' column (which contains 'Train' and 'Validation' labels) and computed the mean for each group. Bar chart visualization for train, validation, overall train and overall validation in MSE, RMSE, MAE and R2 scores can be seen in **Figure 2**.

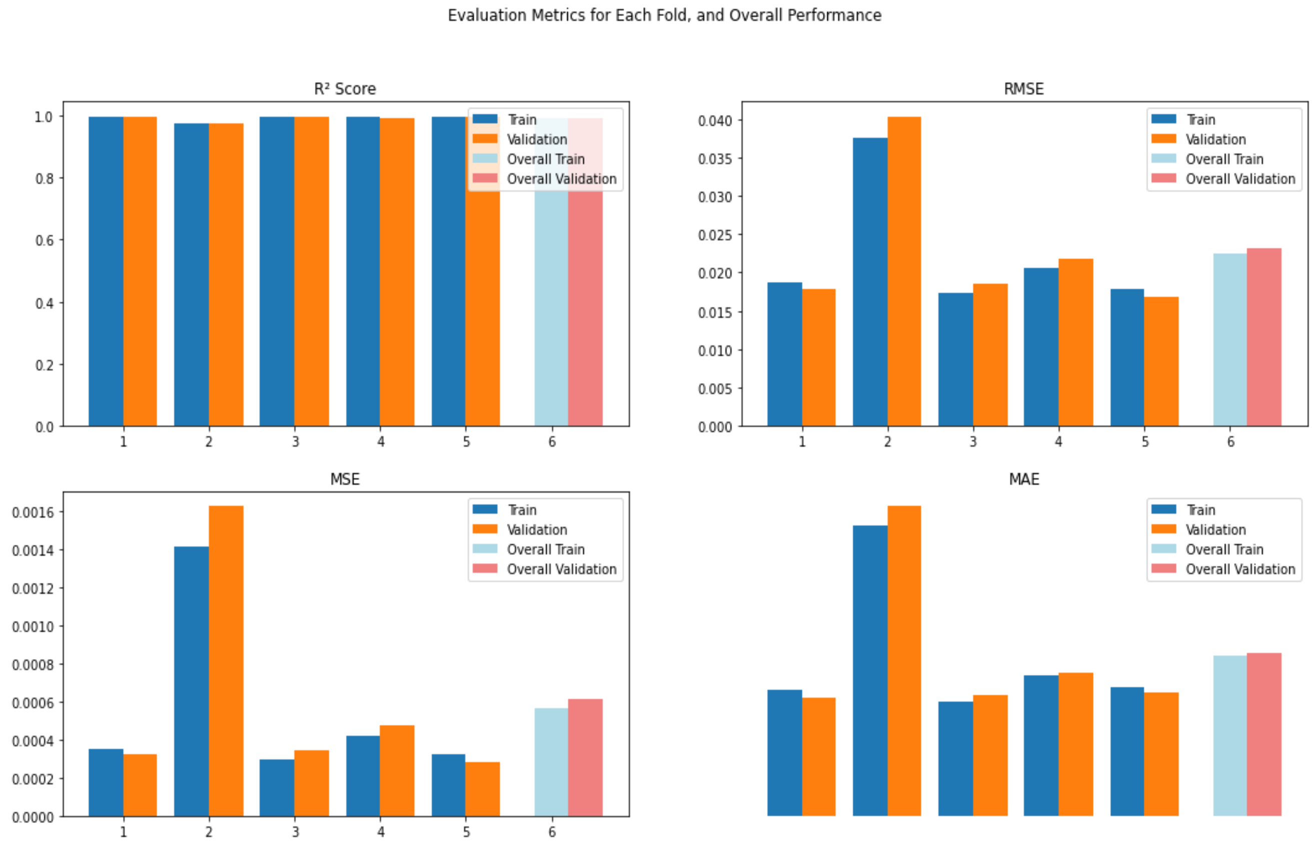

Figure 2: Evaluation Metrics for each fold in Train and Validation Set

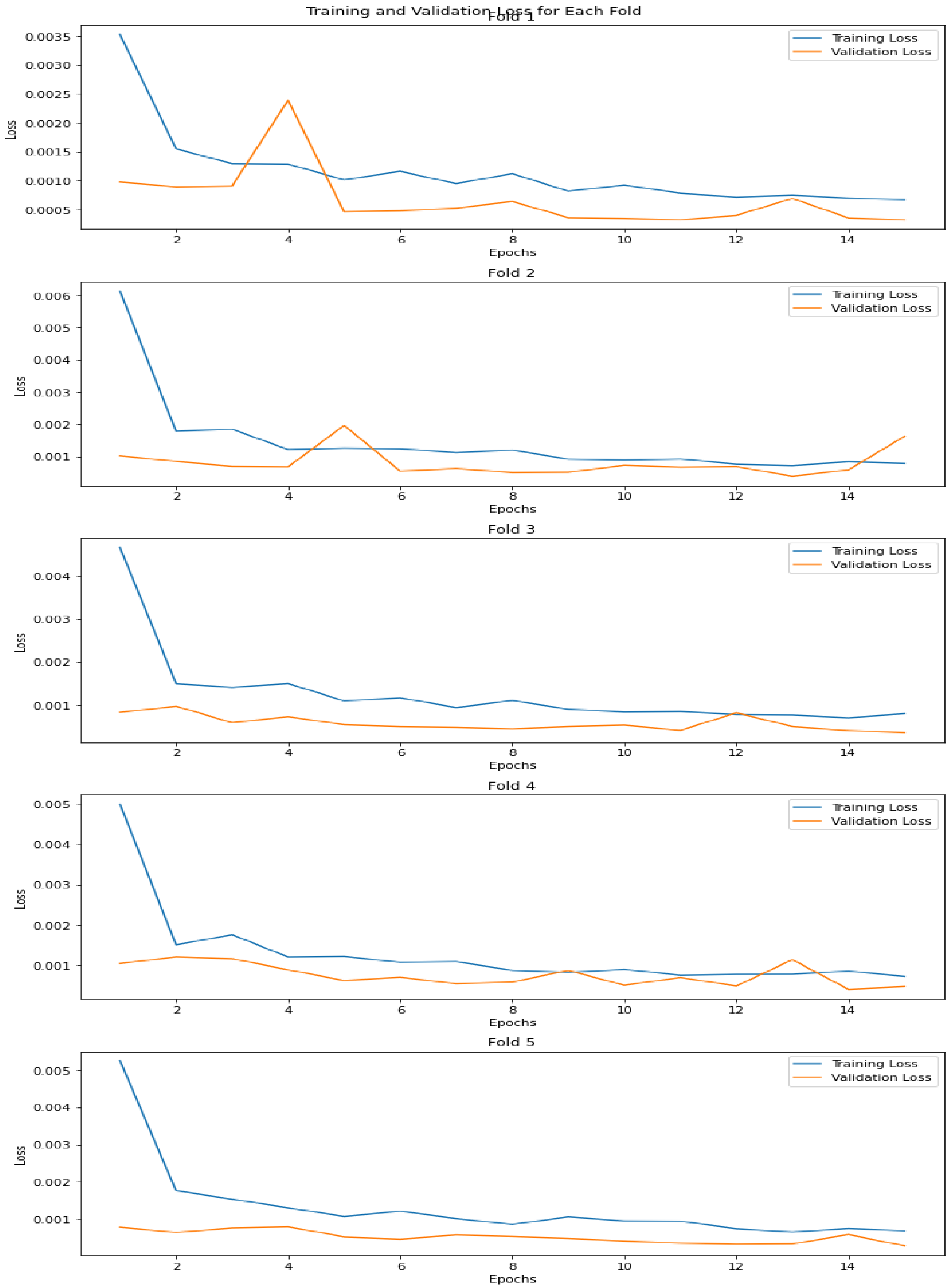

Figure 3: Training and validation loss for each fold.

Lastly, training and validation loss are calculated and plotted for each fold. Results can be seen in **Figure 3** above.

6.2 Discussion

As can be seen in **Table 3, f**or both the training and validation sets, the MSE and RMSE are extremely low across all folds, indicating that the model fits the data quite well. Additionally, the MAE is relatively low, suggesting a low average prediction error. For both training and validation, the R2 values are very near to 1, indicating that the majority of the variability in the target variable around its mean is explained by the model. The model appears to be generalizing well and is not overfitting, as indicated by the metrics' similarity between the training and validation sets across all folds. It can be seen that the validation metrics closely resemble the training metrics and remain consistent across all folds, the model is stable and exhibits comparable performance on various data subsets.

As can be seen in **Figure 3**, both train and validation losses decrease together. There is no fluctuation on validation loss, it decreases steadily. When we look at the losses for each fold, pattern is pretty much same, and the model is not sensitive to the specific data it's trained on. All this interpretation indicates that the model is not overfitted.

The experiment concludes by showing that the LSTM model has excellent predictive performance in predicting BTC prices when it uses the chosen technical indicators as features. The model architecture and features selected appear to be well-suited for the given task, as evidenced by their robustness across all metrics. The model's effectiveness in capturing the underlying trends and patterns of the Bitcoin market is confirmed by the close alignment of its predictions with actual prices, as indicated by the high R2 values.

7 Conclusion and Future Work

This study's main objective was to determine the effect of adding particular technical indicators Super Trend, Fibonacci's Weighted Moving Average (FWMA), Kaufman's Adaptive Moving Average (KAMA), and Average True Range Trailing Stop-Loss—on the precision of BTC price forecasts made with an LSTM model. This study has effectively shown that the predictive accuracy of the LSTM model is greatly increased by incorporating these technical indicators.

The stability of the model was shown throughout the experiment by the performance metrics' consistency across all cross-validation folds. The model's capacity to fit the data well and account for a sizable amount of the variance in BTC prices has been shown by the low values of the MSE and RMSE combined with the high R2 scores. The model's accuracy in generating forecasts is further supported by the consistent MAE, which validates its effectiveness in a highly volatile market environment.

The outcomes of this study have numerous ramifications, as they show not only the potential of LSTM models as useful instruments for predicting cryptocurrency prices, but also the enhanced significance of technical indicators in improving these forecasts The research has elucidated the ways in which these models can assist traders and investors in making informed decisions, ultimately contributing to a more profound understanding of market dynamics.

Despite these encouraging outcomes, it's critical to recognize the study's inherent limitations. The volatility of the cryptocurrency market poses a challenge to the model's ability to be generalized over longer time periods. Also, even with its broad scope, the data still represents only a small amount of market behavior.

Further studies can expand on the basis built through this work in multiple ways. Further improvements in prediction accuracy may be revealed by experimenting with different neural network architectures or by adding more LSTM layers as computing power advances. Also try more epoch on model is another option if computational restrictions can be ignored. The inclusion of a wider range of technical indicators, which would correspond with the vast toolkit that traders employ, could potentially improve the model's resilience and ability to adjust to changes in the market.

Complex understanding of predictive behaviors may be obtained by exploring hybrid deep learning models and adding layers or neurons, depending on the complexity of the model. This could have a special effect on developing various model configurations, each customized to capture particular market phenomena and thus address the wide range of trading strategies used in the cryptocurrency space.

The deployment of these models in automated trading systems holds great potential from a business view. Real-time analysis and execution can be facilitated by using computational scalability. This could result in the creation of complex trading algorithms that accommodate the different risk profiles and trading styles found in the cryptocurrency market, along with the investigation of a larger range of indicators.

Also, as models get more complex, future work should take their interpretability and transparency into account. As the field develops, upholding user confidence and meeting regulatory requirements will depend critically on how understandable AI-driven decisions are. Considering this, optimizing the model's explain ability will be just as important as enhancing its predictive capabilities.

References

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Decentralized business review.

Rathore, P. (2020). Bitcoin: Mystery, History and Background Innovative Technology Involved. [Online]. Available: DOI:10.22214/ijraset.2020.3186.

Miralles-Quirós, J. L., & Miralles-Quirós, M. del M. (2022). A new perspective of the day-of-the-week effect on Bitcoin returns: evidence from an event study hourly approach. [Online]. Available: DOI:10.24136/oc.2022.022.

Rowland, Z., Suler, P. and Cajkovicova, B. (2021). Bitcoin: an alternative currency to pay for goods and services or a useful investment tool? SHS Web of Conferences, 129, p.03026. doi: https://doi.org/10.1051/shsconf/202112903026.

Ye, Z., Wu, Y., Chen, H., Pan, Y., & Jiang, Q. (2022). A Stacking Ensemble Deep Learning Model for Bitcoin Price Prediction Using Twitter Comments on Bitcoin. [Online]. Available: DOI:10.3390/math10081307.

Zhang, S., Li, M., & Yan, C. (2022). The Empirical Analysis of Bitcoin Price Prediction Based on Deep Learning Integration Method. [Online]. Available: DOI:10.1155/2022/1265837.

Ji, S., Kim, J., & Im, H. (2019). A Comparative Study of Bitcoin Price Prediction Using Deep Learning. [Online]. Available: DOI:10.3390/math7100898.

Wilder, J. W. (1978). New concepts in technical trading systems.

Achelis, S. B. (2001). Technical Analysis from A to Z.

Chan, L. K., Jegadeesh, N., & Lakonishok, J. (1996). Momentum strategies. The journal of Finance, 51(5), 1681-1713.

Sidhu, P., Aggarwal, H., & Lal, M. (2021). Stock market prediction using lstm. ICIDSSD 2020, 75.

Ghosh, A., Bose, S., Maji, G., Debnath, N., & Sen, S. (2019, September). Stock price prediction using LSTM on Indian Share Market. In Proceedings of 32nd international conference on (Vol. 63, pp. 101-110).

Hileman, G., & Rauchs, M. (2017). 2017 global cryptocurrency benchmarking study. Available at SSRN 2965436.

Jay, P., Kalariya, V., Parmar, P., Tanwar, S., Kumar, N., & Alazab, M. (2020). Stochastic neural networks for cryptocurrency price prediction. Ieee access, 8, 82804-82818.

Saad, M., Choi, J., Nyang, D., Kim, J., & Mohaisen, A. (2019). Toward characterizing blockchain-based cryptocurrencies for highly accurate predictions. IEEE Systems Journal, 14(1), 321-332.

Oncharoen, P., & Vateekul, P. (2018, August). Deep learning for stock market prediction using event embedding and technical indicators. In 2018 5th international conference on advanced informatics: concept theory and applications (ICAICTA) (pp. 19-24). IEEE.

Shynkevich, Y., McGinnity, T. M., Coleman, S. A., Belatreche, A., & Li, Y. (2017). Forecasting price movements using technical indicators: Investigating the impact of varying input window length. Neurocomputing, 264, 71-88.

Li, Y., Zheng, Z., & Dai, H. N. (2020). Enhancing bitcoin price fluctuation prediction using attentive LSTM and embedding network. Applied Sciences, 10(14), 4872.

Nhan Thi Cao, Dai Quoc Nguyen and An Hoa Ton-That (2022). A Combination of Technical Indicators and Deep Learning to Predict Price Trends for Short-Term Cryptocurrency Investment. doi:https://doi.org/10.1109/csde56538.2022.10089300.

Ortu, M., Uras, N., Conversano, C., Bartolucci, S., & Destefanis, G. (2022). On technical trading and social media indicators for cryptocurrency price classification through deep learning. Expert Systems with Applications, 198, 116804.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Browne, M. W. (2000). Cross-validation methods. Journal of mathematical psychology, 44(1), 108-132.

Weigend, A. S., Mangeas, M., & Srivastava, A. N. (1995). Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting. International journal of neural systems, 6(04), 373-399.