Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Interpretable multi-horizon time series forecasting of cryptocurrencies by leverage temporal fusion transformer

Arslan Farooq [a], M. Irfan Uddin [a,*], Muhammad Adnan [a], Ala Abdulsalam Alarood [b], Eesa Alsolami [b], Safa Habibullah [c]

[a] *Institute of Computing, Kohat University of Science and Technology, Kohat, 26000, KP, Pakistan*
[b] *College of Computer Science and Engineering, University of Jeddah, Jeddah, 21959, Saudi Arabia*
[c] *Department of Information Systems and Technology, College of Computer Science and Engineering, University of Jeddah,Jeddah, 21959, Saudi Arabia*

ARTICLE INFO

Keywords:
Block chain
Deep learning
Cryptocurrency
Sentiment analysis
Artificial intelligence
Transformers
Temporal data

ABSTRACT

This research delves into the obstacles and difficulties associated with predicting cryptocurrency movements in the volatile global financial market. This study develops and evaluates an advanced Deep Learning-Enhanced Temporal Fusion Transformer (ADE-TFT) model to estimate Bitcoin values more accurately. This research employs cutting-edge artificial intelligence (AI) and machine learning (ML) techniques to comprehensively examine various aspects of cryptocurrency forecasting, including geopolitical implications, market sentiment analysis, and pattern detection in transactional datasets. The study demonstrates that the ADE-TFT model outperforms its lower-layer counterparts in terms of forecasting accuracy, with reduced Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), and Root Mean Square Error (RMSE) values, particularly when using a higher hidden layer configuration (h = 8). The study emphasizes the importance of experimenting with different normalization strategies and utilizing various market-related data to enhance the model's performance. The results suggest that improving forecasting accuracy may require addressing these limitations and incorporating additional factors, such as market sentiment. By providing investors with more precise market predictions, the techniques and information presented in this research have the potential to significantly increase investor power in an unpredictable digital currency market, enabling wise investment choices.

1. Introduction

Investors worldwide experience significant financial uncertainty because of the drop in traditional currency values and vulnerabilities in the stock market. This unstable environment has sparked the transition to digital currency as a novel alternative. The world saw the future of decentralized financial transactions with the launch of Bitcoin (BTC) in 2009 [43]. Blockchain technology, which is the foundation of cryptocurrencies, introduces two key changes from conventional financial systems [47]. First, they are entirely digital, allowing for instantaneous global transactions that get around the restrictions and delays associated with traditional banking.

* Corresponding author.
E-mail addresses: syed277526@gmail.com (A. Farooq), irfanuddin@kust.edu.pk (M. Irfan Uddin), adnan@kust.edu.pk (M. Adnan), aasoleman@uj.edu.sa (A.A. Alarood), eaalsulami@uj.edu.sa (E. Alsolami), shhabiballah@uj.edu.sa (S. Habibullah).

https://doi.org/10.1016/j.heliyon.2024.e40142
Received 19 February 2024; Received in revised form 16 September 2024; Accepted 4 November 2024

Available online 5 November 2024

Every transaction is recorded on a public ledger maintained by a decentralized computer network. Second, transactional security offered by digital currencies is unmatched because of their inherently cryptographic nature.

Any deception or fraud is extremely difficult because it requires validation from the entire network. Digital currencies also avoid the difficulties associated with conventional finance, such as changing exchange values and interest rates. Many even provide the extra benefit of low or no transaction costs. After Bitcoin's launch, the world of cryptocurrencies exploded, with a wide variety of other virtual currencies. This progression sparked an increase in computer science research. Forecasting has several facets in the world of cryptocurrencies. It involves more than just forecasting price changes based on past data; it also considers studying market sentiment, comprehending geopolitical developments that affect the adoption of digital currencies, and spotting patterns in sizable transactional datasets. Forecasting is a dynamic and ever-evolving field, and advanced ML and AI techniques are being used to increase prediction accuracy. However, Bitcoin's notorious volatility has deterred substantial institutional investors, resulting in a market landscape dominated by less-experienced traders [2]. Such a shift has ramifications for the trading volume and overall market stability. Against this backdrop, this study developed a robust forecasting model for the cryptocurrency market. We aim to empower investors to anticipate pivotal market events and make judicious investment decisions based on informed forecasts.

With the continuous expansion of deep learning (DL), various relevant learning methods, such as Long Short-Term Memory (LSTM), Temporal Convolutional Networks (TCN), and transformers, have been enhanced and deployed in a manner that is more appropriate for extracting time-series information. LSTM is an improved RNN that can be trained to extract relevant information from the temporal data automatically. It can be used to analyse time series and works by accepting inputs related to past and future states, thus considering attributes related to time [21]. TCN are networks that focus on interpreting sequence data and provide high parallel computing capabilities, a large receptive field, and a constant gradient that LSTM lacks [35], [24], [20]. In addition, some researchers have begun to use attention mechanisms for sequence problems because they perform well in computer vision and natural language processing [46,27]. The Transformer model improves the interpretability of the model by classifying a significant portion of the input for each instance, depending on the size of the attention weight. In terms of capturing long-distance dependencies, the transformer model performed noticeably better than the RNN approach.

Due to several features such as lack of trustworthy indications, high unpredictability, and multiple affecting elements including technology improvements, market pressures, use patterns, expenses, security concerns, and cultural circumstances, price prediction for cryptocurrencies is a difficult endeavour. Cryptocurrencies have more unpredictability than conventional financial projections, such as stock market forecasts, making investments riskier and less rewarding [3]. Owing to their inherent volatility and scarcity of historical data, cryptocurrency values cannot be predicted with any degree of accuracy.

The proposed work's main objectives are:

1. To determine the precision of the ADE-TFT model, future results will be forecast.
2. To demonstrate the importance of variables in forecasting using the interpretation of the model.
3. To evaluate the model performance using supervised metrics techniques.

2. Background and related work

2.1. Multivariate time series

Time-series data can be univariate or multivariate. Single data points can be seen in the first scenario, but many data points are acquired for each iteration of the observation process in the second [5]. Considering the important factors for prediction, it is likely that the results will improve. One may argue, for instance, that how much salmon is purchased at a supermarket during a given day depends not only on how many replacements are purchased at that time but also on how many customers are there. The financial time series' final stock price may also be influenced by the stock's open, high, low, and volume and its historical value. In multivariate time-series systems, spatial and temporal features can be utilized for more information to improve forecasting. Temporal connections illustrate how well the variables vary over time, whereas spatial relations demonstrate how one variable affects others. Each time step in multidimensional data is expressed as a vector, $\mathrm{Z_t} = (z_{1t}, z_{2t}, \ldots, z_{kt})$, where k denotes the total number of features used in the prediction [8]. Consequently, the characteristics and implementation of each series were considered. This suggests that the lag among the time series should be used to determine the covariance between them rather than the time interval between them [40].

2.2. Financial time series

In time-series literature, price movements are a typical illustration of non-stationarity and nonlinearity. Additionally, they are notable for having a lot of noise [8,13,29]. Financial time series contain impacts and causes that can be identified by data analysis. For instance, [9] discovered daily trading trends between volume trading and returns for stock market indices of foreign currency, and [18] discovered a lead-lag correlation between both Bitcoin and commodity markets. Many investors use chart patterns to determine market-shifting moments and their objectives. It is possible to analyze the differences between a stock's short- and long-term moves using multiple moving average windows. Additionally, the use of Bollinger Bands, an indicator that captures the price and volatility history of a financial asset, is widespread [15,23].

2.3. Traditional forecasting methods

Traditional forecasting techniques can derive linear patterns from data using historical processes. The single-line formula, which is adapted to the data's simplicity of understanding, is a benefit of conventional methods. They are also user-friendly and memory-efficient.

2.4. Autoregression and moving average

A statistical technique called autoregression is used to forecast a variable's future value based on its past values while analyzing the time series. This is a linear regression model; in other words, it employs lag variables as predictors. The word "autoregression" derives from the fact that the variable's current value is based on its prior values, hence the "auto" in the phrase. An order p autoregressive model, designated by AR(p), can be expressed mathematically as

$$\mathrm{y_t} = \mathrm{b} + \phi_1 \mathrm{y}_{(\mathrm{t}-1)} + \phi_2 \mathrm{y}_{(\mathrm{t}-2)} + \ldots + \phi_\mathrm{p} \mathrm{y}_{(\mathrm{t}-\mathrm{p})} + \epsilon_\mathrm{t} \tag{1}$$

where y_t is the relevant variable at time t, b is a positive constant, $\phi_1, \phi_2, \ldots, \phi_p$ are the vectors of autoregressive coefficients, ϵ_t is the error term, and p is the order of the autoregression. The error term ϵ_t is assumed to be white noise, which means it has a mean of 0 and constant variance and is uncorrelated with the lagged values of y. The autoregressive coefficients $\phi_1, \phi_2, \ldots, \phi_p$ represent the impact of the past values of y on the current value. A positive autoregressive coefficient for a particular lag indicates that the current value of y is positively related to the value at that lag. Conversely, a negative coefficient indicates a negative relationship. The time series data autocorrelation function (ACF) and partial autocorrelation function (PACF) were used to determine the order of autoregression. While the PACF calculates the correlation between a variable at a specific lag and its previous lags, the ACF calculates the correlation between variables at various lags. We can determine the correct order of autoregression by analyzing the ACF and PACF plots.

In contrast to the autoregressive (AR) model, the Moving Average (MA) takes the average of a specified number of prices over a given time. The result is a line that tracks the average price of an asset over a specified time, which can help identify trends and provide signals for trading. Specifically, an MA model of order q is expressed as a linear combination of the q previous forecasting errors.

$$\mathrm{y_t} = \mu + \theta_\mathrm{t} \epsilon_{\mathrm{t}-1} + \theta_\mathrm{t} \epsilon_{\mathrm{t}-2} + \ldots + \theta_\mathrm{t} \epsilon_{\mathrm{t}-\mathrm{q}} + \epsilon_\mathrm{t} \tag{2}$$

In the given equation, the parameter $\theta_t \in \mathbb{R}$ represents the model parameters, while $\epsilon_t \sim N(0, \sigma^2)$ [1,22] is the predictive error that has a mean of 0 and a variance of σ^2 under a standard deviation. White noise is used to model these errors. Additionally, μ stands for the time series' average value.

Autoregressive (AR) and Moving Average (MA) models are frequently combined to produce complex models. These models include the ARMA model [22], ARIMA model [42], and ARIMAX model [44]. The ARMA model, which combines the MA(q) and AR(p) models, is the most fundamental among the three models.

$$\begin{aligned} &\mathrm{ARMA(p,q)} = \mathrm{AR(p)} + \mathrm{MA(q)} = \\ &\quad c + \epsilon_t \sum_{i=0}^{p} \phi_i y_{t-i} + \sum_{j=1}^{q} \theta_t \epsilon_{t-j} \end{aligned} \tag{3}$$

The Exponential Smoothing (ES) approach is an alternative to the MA (Moving Average) and autoregressive (AR) models. In ES, the weights assigned to past observations are not uniformly weighted, as in the AR and MA models. Instead, the weights in the ES models were exponentially reduced over time. The Simple Exponential Smoothing (SES) model [22,26] is one of the simplest ES models, and is suitable for time series that do not exhibit a trend. However, more complex models such as the Double Exponential Smoothing (DES) model are more appropriate for time series with a trend.

$$\hat{\mathrm{y}}_\mathrm{t} + 1 = \alpha \mathrm{y_t} + (1-\alpha)\mathrm{y}_{\mathrm{t}-1} + (1-\alpha)^2 \mathrm{y}_{\mathrm{t}-2} \tag{4}$$

Reddy et al. compared the LASSO machine-learning algorithm with other models, such as QUANDL, RNN, SVM, and CNN. LASSO offers excellent time management, allowing it to get superior results from a large dataset. They have the highest level of accuracy in predicting daily trending changes in the Bitcoin market [39]. Derbentsev forecasted the bitcoin values by using two machine learning algorithms, stochastic gradient boosting machine (SGBM) and random forest (RF). The results demonstrate that machine-learning approaches can estimate Bitcoin values. However, the decisions were made at the appropriate time to reduce the risks associated with the investment decision [14].

Kumar et al. demonstrated that everyday data comprised 1000 data samples, hourly data of 1500 data samples, and minute data of 400000 data lines. They discovered how DL algorithms enabled pricing trends in the Ethereum cryptocurrencies. For forecasting Ethereum value and mapping Mean Absolute Percentage (MAPE) errors, they used Multi-Layer Perceptron (MLP) or Long Short-Term Memory (LSTM) models [25].

A study conducted by Ahmed B. et al. performed a comparative analysis of deep learning and ensemble learning models for predicting various cryptocurrencies [6]. The results showed that ensemble learning and deep learning models perform better than shallow neural networks and traditional statistical methods. If there is less complexity in the time series cryptocurrency data then conventional statistical methods can achieve better results for regression metrics. The study concluded that LightGBM outperformed Bitcoin, Litecoin and Ethereum cryptocurrencies while GRU showed the best result for Ripple.

Another study by Lucas D. A. Takara et al. argued that advanced computation models based on deep reinforcement learning (DRL) are better in decision-making, particularly in quantitative trading [12]. If properly trained on past data, DRL can automatically develop profitable strategies for trading. The authors presented a customized and innovative model called Extended Trading DQN (ETDQN), based on the Deep Q-Network (DQN) algorithm. The particular feature of ETDQN was that it was dynamic and adapted to changing market conditions. Moreover, ETDQN receives feedback only when trades are completed, enabling the model to perform better trade decisions. The distributed learning feature of ETDQN enables it to improve decision-making and only focus on maximizing profits.

Other studies such as [16] and [41,17] used various ML models such as GARCH, HAR, SVR, LASSO, MLP, RF, and LSTM to forecast cryptocurrency volatility that will help investors in making the right and informed investment decisions. Because there are several cryptocurrencies, no single ML model is best for all cryptocurrencies. The studies revealed that different models perform differently depending on forecast context and error metrics. Interestingly, simple models such as linear regression and ridge regression show similar results to that of complex ML models such as LSTM and RF.

Lopez and other scholars collected data from six different facilities in Germany and Australia by applying the TFT model to predict hourly day-ahead PV power generation with statistical error indicators to compare the outcomes with other models. TFT has shown more precise results than the other algorithms to forecast PV power generation [31].

Furthermore, Feng addressed the supply air temperature in high-speed train carriages by enhancing two TFT architecture components: the Double-Convolutional Residual Encoder and the spatio-temporal double-gate. Additionally, a loss function is also developed which is appropriate for generic long-sequence time-series forecast tasks for predicting temperature, that increases MAPE by 11.73% and 21.70% compared to the original model [19].

[45] suggested a novel wind speed prediction system that uses VMD, adaptive DE (ADE), and TFT algorithms. The system distinguishes the significance of each meteorological variable for wind-speed forecasting by considering past and present wind-speed data and a variety of climate parameters simultaneously. While ADE optimizes the TFT model variables, VMD divides the raw wind speed information into various band-limited mode functions. The proposed approach performs more effectively than other similar models, demonstrating its consistency and dependability. Additionally, it identifies essential aspects in predicting wind speed, supporting decision-makers in their choices.

[49] proposed transformer-based forecasting models for cryptocurrencies. The model was fed with historical tweet data as well as cryptocurrency price information. The proposed model makes predictions using attention mechanisms to identify significant patterns and features in the input data. The model was tested using two different datasets, Ethereum and Bitcoin, with encouraging outcomes. The accuracy of the model, which outperformed other approaches, was 75.87% for Bitcoin and 78.55% for Ethereum. According to the study, adding sentiment analysis to the prediction model can significantly increase the accuracy of predicting cryptocurrency prices.

[20] proposed two deep models: the LSTM network, which was used with deep multi-input, multi-output (MIMO), and multi-input single-output (MISO) architectures, and the TCN network, which was combined with the MIMO and MISO architectures for long-term forecasting. These models, which can be run on a standalone computer, accept up to 10 parameters as input and output a single parameter as a prediction. The data source was a set of daily weather records from South Korea from 2011 to 2018. The proposed models produce 9-hour weather forecasts that are trustworthy and accurate predictions, which have not previously been investigated in weather forecasting.

[30] studies propose deep learning models like LSTM, BiLSTM, and convolutional layers in an ensemble learning approach to forecast cryptocurrency prices. Ensemble models are assessed on both classification and regression tasks and are based on three widely used ensemble learning strategies: ensemble averaging, bagging, and stacking. The authors agree that careful feature engineering and hyperparameter tuning are necessary to enhance the prediction accuracy of the ensemble models. Choosing the quantity and type of base learners in an ensemble strategy is important for precise and trustworthy predictions, but it can also impact the cost and speed of computation. Owing to its accuracy and dependability, this approach holds promise for low-frequency applications, even though it might not be feasible for high-frequency real-time applications.

[34] uses semantic analysis and machine learning approaches to investigate the consistency of Bitcoin's USD price direction. This study concentrates on the Bitcoin closing price and sentiments of the present market to create a forecasting model. Twitter and Reddit were used to collect data on popular opinions on Bitcoin. The findings demonstrate that in comparison to the ARIMA model's RMSE of 209.263, the LSTM with multiple features produces a more accurate prediction, with an RMSE of 197.515. The report admits that the model's accuracy and consistency would be enhanced by the capacity to anticipate data streaming. The author also acknowledges that it may be biased to only look at tweets and posts on Reddit and Twitter and that adding data from LinkedIn and Facebook posts might yield a more complete picture of public opinion. Finally, the study found that LSTM is a more effective and accurate model than ARIMA for predicting Bitcoin's USD price direction.

[33] attempted to anticipate the daily movements in Bitcoin, Ethereum, and Ripple prices using LSTM networks using various factors, including sentiment, power prices, financial instability, and historical price data. In addition to sentiment data that went beyond the individual cryptocurrencies being forecasted, the article was distinguished for its usage of an original variable set that had never been examined previously. The findings show that the top-performing models for Ethereum and Ripple had accuracy levels of over 50% overall and up to 72% in forecasting rising Ethereum trends, which might help cryptocurrency investors make better decisions. However, these models were not sufficiently precise for automatic trading. The generated models showed overfitting symptoms, proving that the particular collection of variables combined with LSTM was unreliable for forecasting Bitcoin trends. Ultimately, by

demonstrating the potential of LSTM networks and our weeksits in foretelling Bitcoin trends, this study made a significant addition to the area of cryptocurrency prediction.

[10] authors' use of a multivariate method for estimating patient volume in public hospitals by leveraging the deep neural network architecture of the Temporal Fusion Transformer, on the dataset of Emergency Departments (EDs) of Portuguese public medical facilities by Health Regional Areas (HRA). For four weeks, the model projected forecast intervals and precise predictions using covariates from the calendar and time series. With a Mean Absolute Percentage Error (MAPE) of 5.90% and a Root Mean Squared Error (RMSE) of 84.4102 individuals/day, the results demonstrated that the model outperformed other frequently encountered models discussed in the literature. By including more pertinent static variables and enlarging the forecast window, this study also demonstrates the possibility of future advancements.

[32] examined the predictability of financial returns for six significant digital currencies Binance Coin, Bitcoin, Cardano, Dogecoin, Ethereum, and Ripple during the pre-COVID-19 and ongoing COVID-19 periods. In contrast to conventional forecasting, this study suggests a novel method to identify cryptocurrency returns that fall within the first, second, third, or other quantiles of gold prices the following day. The support vector machine (SVM) technique was used to analyze the data, and the proposed algorithm enabled updated data analysis by utilizing sensors in the database. The study results confirm that the SVM algorithm is a reliable method for building profitable trading approaches, and can be applied with the help of algorithms to achieve a reliable estimate before or during a global pandemic [7]. During challenging times such as the COVID-19 crisis, stakeholders who are keen on following Bitcoin trends may benefit from this research by enhancing their understanding of Bitcoin dynamics and subsequently making wise investment decisions.

Another related work conducted by [28] examined the prediction of Bitcoin price changes due to significant fluctuations using fundamental market indicators as well as technical features obtained through a denoising autoencoder. The Attentive LSTM network and Embedding Network were used by the authors to evaluate the features. (ALEN). ALEN surpasses all baselines by gathering hidden representations from related currencies and capturing the representation of Bitcoin. This study explores the effects of several characteristics on the forecast of Bitcoin price fluctuations, which may have practical benefits for investors in a real market environment. The findings imply that technical market indicators outperform the characteristics produced by DAEs, and adding associated cryptocurrencies can help predict swings in the price of bitcoin. Additionally, ALEN performs 3.3% better in terms of accuracy and 3.2% better in terms of the F1 score than ALUE, which employs a uniform embedding approach.

[38] used three algorithms based on deep learning to forecast daily arrivals at an emergency room, which frequently experiences overload and seasonal increases. While the last two models are based on a newly created TFT architecture for multi-horizon time-frame forecasting, the first model is a straightforward RNN with LSTM cells. The dataset used to train the models contained a variety of pertinent variables, with the TFT models beating the LSTM model for prediction success. The hourly frequency TFT model outperforms the competition model by using layers of temporal self-attention and a smart network design to develop dependency patterns over time. The TFT model also provides greater interpretability, enabling the detection of important variables and eliminating those that provide little value. Information on what the model is attentive to regarding time-series segments can be deduced from the self-attention layer weights.

In their research, [37] addressed three issues to better understand this complex problem, which included employing advanced deep learning architectures for predicting cryptocurrency prices. The findings indicated that the chaotic and complicated nature of the market structure hinders deep learning algorithms such as LSTM and CNNs from being good predictors of the Bitcoin price. It uses a clever mechanism to detect a few hidden patterns in the random walk process that determines the path of cryptocurrency prices to achieve accurate predictions. The study concludes that additional validation techniques, complex computational methods, and ensemble methods are required to increase the precision of Bitcoin price prediction. Accurate price prediction is a difficult challenge, given the growing importance of cryptocurrencies in the financial sector; therefore, researching new methods is essential for better results.

[36] explores the difficulties and opportunities associated with academic research on financial time-series data, including different approaches to issue framing and feature extraction. Researchers aim to close the gap between academia and business by proposing a machine-learning-based forecasting approach that starts with feature extraction and selection. Hourly statistics from Solana, Bitcoin, and Ethereum were gathered from the exchange FTX and used in the study. This study investigated the problem statement of how effectively market movements can be predicted and the contribution of each feature to the forecasts for a six-hours-ahead regression job using a collection of candlestick patterns and a feature selection method. The findings demonstrate that several forecasting models provide evidence of a market's short-term predictability. When volatility was low, LSTM and ARIMA-GARCH performed the best; LSTM outperformed the other models. The experiments also highlighted the significance of feature selection for some time pertinent to the prediction window and non-stationary indicators. Finally, the data exhibit a significant mean-reverting pattern and may be roughly predicted by a Naïve walk.

3. Methodology

Fig. 1 shows the complete workflow of Bitcoin price prediction by leveraging the ADE-TFT. The workflow is divided into four phases: data retrieval, preprocessing, model construction and performance evaluation. In the data retrieval phase, the daily transaction dataset for Bitcoin is gathered from many sources (Kaggle, DataHub, and DataWorld) and integrated such that it has at least eight years of daily transaction records from the past and only extracts the key elements required for this study.

Before preparing the data, an essential stage is filtering the necessary data columns, which entails choosing only the appropriate data columns needed for analysis. By doing so, the amount of data that must be processed can be decreased, enhancing the effectiveness

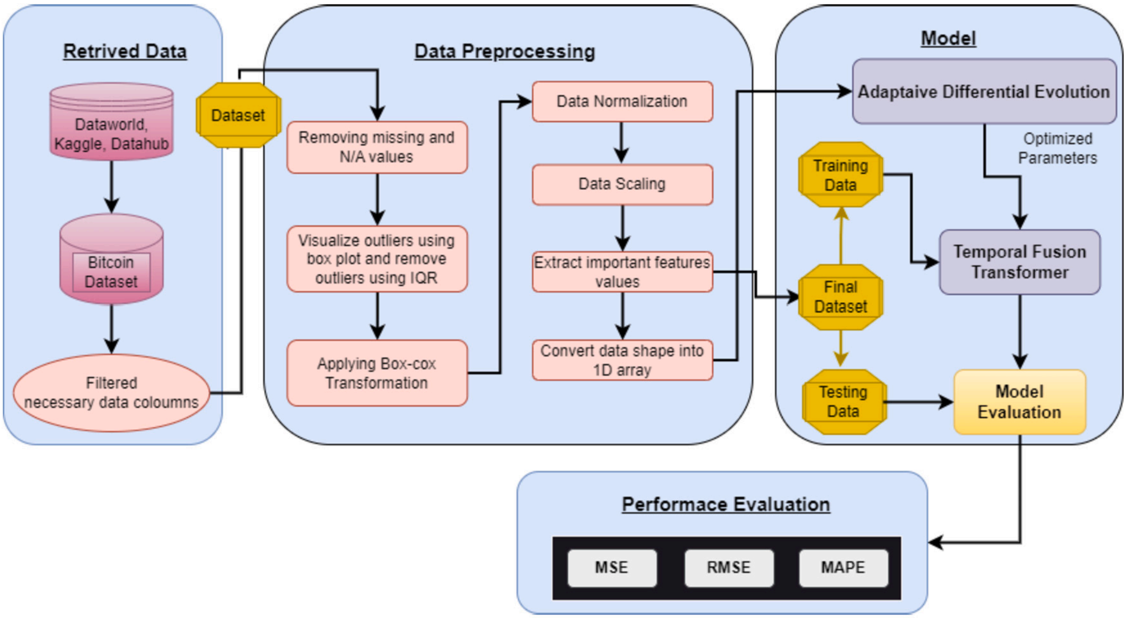

Retrived Data
Dataworld, Kaggle, Datahub
Bitcoin Dataset
Filtered necessary data coloumns
Dataset
Data Preprocessing
Removing missing and N/A values
Visualize outliers using box plot and remove outliers using IQR
Applying Box-cox Transformation
Data Normalization
Data Scaling
Extract important features values
Convert data shape into 1D array
Model
Adaptaive Differential Evolution
Optimized Parameters
Training Data
Final Dataset
Testing Data
Temporal Fusion Transformer
Model Evaluation
Performace Evaluation
MSE
RMSE
MAPE

Fig. 1. The image illustrates a workflow for Bitcoin price prediction encompassing data retrieval, preparation with cleaning and outlier detection, preprocessing with normalization and correlation analysis, model training using ADE-TFT, and performance evaluation through regression metrics.

of the analysis. Raw data must be changed and cleaned by removing N/A and missing values rather than values that can be filled with mean, median, or mode values for the respective features. The next step involved the use of a box plot library to visualize the outliers. Any data points that are noticeably distinct from the rest of the dataset can be identified using this graphical approach. Once they have been located, outliers are eliminated using the Interquartile Range (IQR) approach to ensure that they do not harm the findings of the analysis. The IQR method, which calculates the difference between a dataset's upper and lower quartiles, is frequently used to identify the range of values that fall within the middle 50% of the dataset. Any data point that deviated from the IQR range was eliminated as an outlier. After removing outliers, it is crucial to ensure that the remaining data are normally distributed. This skewed or unusual data prediction may result in inaccurate or unreliable analytical output. The Box-Cox transformation is frequently used as a well-known data preparation approach to normalize data and lessen its skewness to solve this issue. To transform the data such that it conforms to a normal distribution, this process involves raising the values of the data to power and then estimating the best lambda. The Box-Cox transformation improves the distribution of data, producing more reliable and accurate analytical results.

Further, after preparing and cleaning the data, they need to be normalized and scaled to ensure uniformity and standardization. In this study, the z-score method was used for data normalization, whereby the data were converted into a standard normal distribution with a mean of 0 and a standard deviation of 1. This helps neutralize variations owing to the different scales of various variables within the dataset. Min-max scaler was used and each data point got the same weightage. It was used to scale the data between zero and one. This scaling is particularly useful when relative differences between the values are important, but not absolute ones. In addition to normalizing or scaling, other important attributes that are highly related to the target variable must be extracted or identified from such features. Pearson's correlation was used in this study to determine the linear relationship between the two variables. It is feasible to minimize the dimensionality of the data and choose only the most important characteristics for analysis by locating the strongly associated features. Reducing the amount of noise in the data and increasing the signal-to-noise ratio can produce analytical findings that are more accurate and effective.

Fig. 2 shows the complete operation of the Temporal Fusion Transformer (TFT) model. TFT is a deep-learning algorithm that leverages the benefits of both transformers and temporal convolutional networks (TCNs) to predict time-series data. The TFT model is built on an attention-based deep neural network architecture to integrate multi-horizon forecasting with a clear understanding of temporal dynamics. The attention mechanism adds new levels of interpretability to the model. TFT involves complex inputs such as static covariates, known future inputs, and exogenous time series, which are only observed historically. By using several techniques, including sequence-to-sequence layers for local processing of observed inputs, sample-dependent variable selection to weed out unnecessary inputs, a static covariate encoder for encoding context vectors, and temporal self-attention that makes use of a decoder to capture long-term dependencies in the dataset, the model combines accurate forecasting with interpretability.

Other key features of the Temporal Fusion Transformer are:

a) A Gating Residual Network (GRN) supports multiple datasets and dilemmas by dropping unnecessary elements of architecture to avoid nonlinear processing. b) Variable Selection Network (VSN) utilizes a GRN under the hood for its filtering capabilities, producing a normalized vector of weights. Finally, the output is calculated using a linear combination. c) Static covariate encoders transform input variables into a fixed-size vector representation. This vector representation, sometimes referred to as a contextual feature vector, includes details of the static variables that might provide the model with extra context and boost its prediction precision. TFT

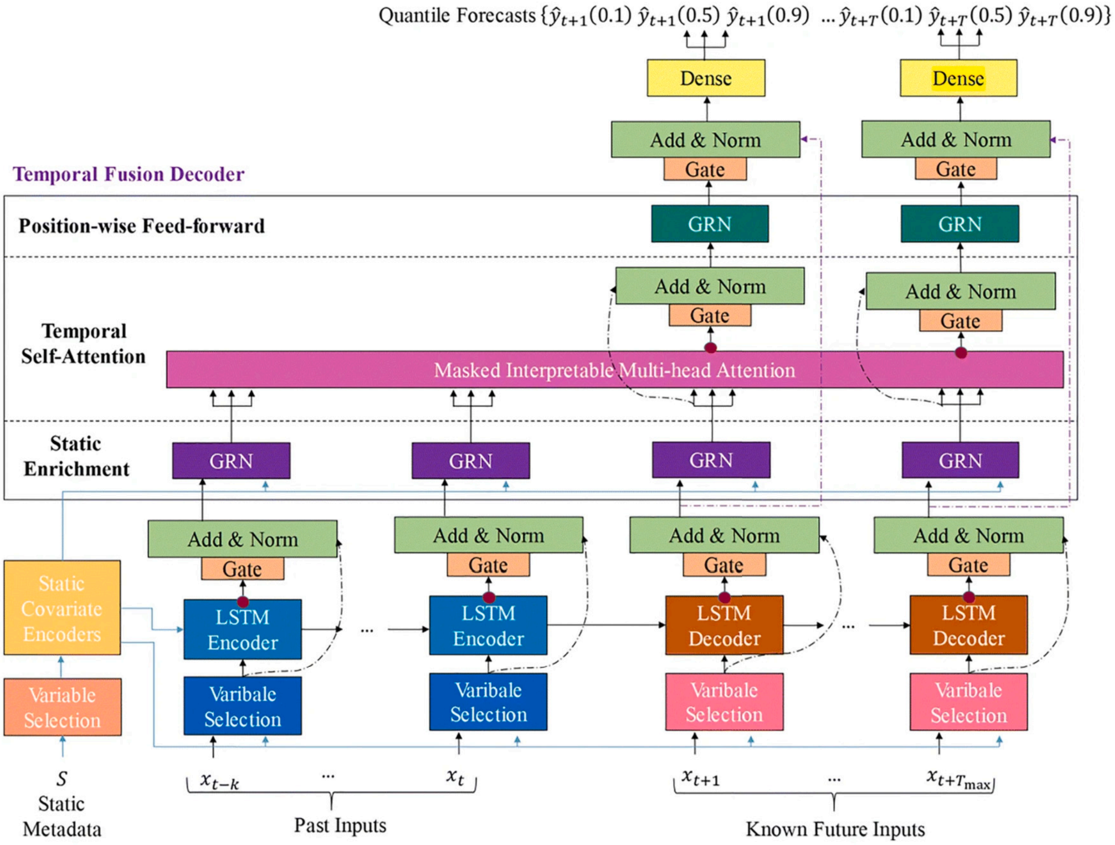


Fig. 2. Illustrates the TFT's model framework. TFT's ability to efficiently generate feature representations for each type of input using fundamental components improves the performance of many prediction tasks. Components of TFT include an input embedding layer, Static Covariate Encoders, a Gated Residual Network, a Temporal Fusion Decoder and an output layer.

may exploit extra information and increase prediction accuracy by integrating static covariates via the implementation of a static covariate encoder, especially when analyzing time-series data with complex inputs. d) Temporal processing entails the creation of temporal connections from known time-varying inputs and either short- or long-term data. For short-term local processing, features are created by utilizing the sequence-to-sequence layers of LSTM encoders and decoders for past inputs and observations. TFT multi-head attention component captures the long-term relationships between inputs and is used to capture long-term dependencies. This attention method ranks the input variables according to the magnitude of their attention weights, enabling the detection and removal of underperforming variables. TFT can successfully capture both short- and long-term temporal connections by integrating sequence-to-sequence layers with a multi-head attention block, allowing it to provide more precise predictions. e) Multi-level prediction intervals use a Gaussian mixture model to provide prediction intervals to account for the variability of the target variable. Generate prediction intervals with various degrees of confidence based on estimations of the conditional distribution at each prediction horizon, which can help make decisions in the face of uncertainty. This provides accurate and reliable forecasts.

The properties of the TFT model significantly impact its accuracy and performance. However, selecting the best set of parameters can be challenging. An efficient and trustworthy optimization technique is required to solve this problem. One such algorithm that has proven to be both simple and effective is adaptive differential evolution (ADE), which belongs to the class of evolutionary optimization methods. Evolutionary optimization algorithms offer the benefit of searching across a wider range of parameters, including the input step size, which is not accessible in many other tuning approaches, as compared to other neural network parameter tuning techniques. In this study, ADE was used to determine the optimal combination of hyperparameters for the TFT model, including the number of time steps, learning rates, batch sizes, hidden layer counts, consecutive hidden layer counts, and attention head count. Fig. 3 shows the ADE process that incorporates the TFT model for performing prediction tasks.

The initial setup of the ADE algorithm involved setting up its parameters and population. This involves determining the maximum number of iterations allowed (T), crossover factor range (CR), range of the mutation factor (F), size of the population (NP), and gene range to be included in the population. The range of values that each hyperparameter could have was determined using the gene range. A random population was formed based on the gene range to provide a diversified population. Step 3: Primary stage of optimization. Mutation, crossover, and selection activities are used to form the population for the upcoming generation. Equation (5) is used to calculate the mutation factor, where 'F_{max}' and 'F_{min}' stand for the variation factor's maximum and minimum values, respectively.

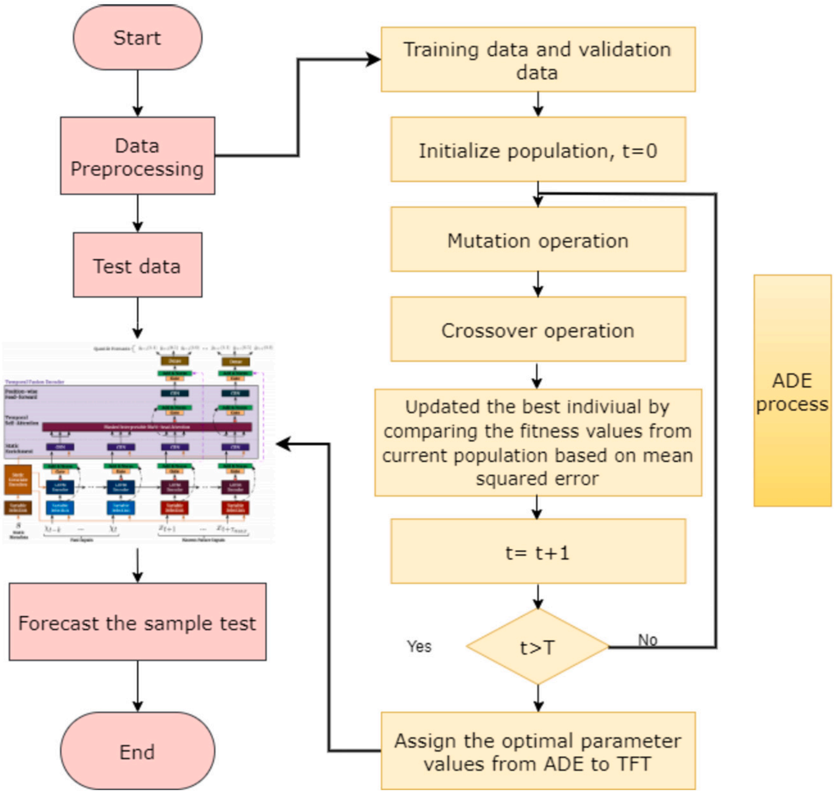

Start
Data Preprocessing
Test data
Forecast the sample test
End
Training data and validation data
Initialize population, t=0
Mutation operation
Crossover operation
Updated the best indiviual by comparing the fitness values from current population based on mean squared error
t= t+1
Yes
t>T
No
Assign the optimal parameter values from ADE to TFT
ADE process

Fig. 3. ADE process includes training and validation data, population initialization, mutation operation, crossover operation and finding optimal parameters for TFT.

$$F = F_{\min} + (F_{\max} - F_{\min}) \cdot \frac{1}{1 + e^{10\cdot(0.5-\frac{t}{T})}} \tag{5}$$

The dataset was divided into three parts: the training set, validation set, and testing set. 70% of the data was reserved for the training set, while 15% was allocated to both the validation and testing sets. The validation set Mean Absolute Percentage Error (MAPE) was used to compute the fitness value. The values 'T' and 't' stand for the maximum and current iteration counts, respectively. The third step is repeated a predetermined number of times to enable the ADE algorithm to converge to the ideal hyperparameters. The variable 't' is for a specific point in time within the sequence i.e., it could represent the price on a particular day. On the other hand, 'T' identifies the complete length of the time series or the maximum time taken for the analysis. The ADE individual with the highest fitness value was selected to obtain the best hyperparameters, which were then assigned to the TFT model. Subsequently, by utilizing both the training and validation datasets, the TFT model was trained. This step ensures that the TFT algorithm can accurately identify and interpret key characteristics of the data and produce accurate forecasts. Predictions on the test dataset were performed using the TFT model with the best training performance. This ensures that the TFT algorithm can produce accurate and reliable forecasts using fresh untested data. Algorithm 1 shows the detailed steps performed by the ADE algorithm for optimal feature and parameter selection.

Algorithm 2 shows the step carried out by TFT to predict bitcoin prices.

3.1. *Mathematical formulation of ADE-TFT model*

The ADE-TFT model joins numerous advanced methods to manage the temporary dependencies and complicated patterns in time-series data. Below, we provide the mathematical definition and explanation of the basic components of the ADE-TFT model. The input data to the ADE-TFT model consists of two parts: past observation and future data.

3.1.1. *Input embeddings*

Let $\mathbf{X}_{\text{past}} = \{\mathbf{x}_{t-n}, \mathbf{x}_{t-n+1}, \ldots, \mathbf{x}_t\}$ and $\mathbf{X}_{\text{future}} = \{\mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \ldots, \mathbf{x}_{t+m}\}$ be the past and future input sequences, respectively.

The embedding for the past and future inputs is represented as:

$$\mathbf{E}_{\text{past}} = \text{Embedding}(\mathbf{X}_{\text{past}})$$

$$\mathbf{E}_{\text{future}} = \text{Embedding}(\mathbf{X}_{\text{future}})$$

Algorithm 1 Adaptive Differential Evolution.

```
Input:
$sizepop$: Size of population
$vardim$: Dimension of variables
$bound$: Boundaries of variables
$MAXGEN$: Maximum number of generations (termination condition)
$params$: List containing crossover rate $CR$ and scaling factor $F$
  Output:
Optimal solution and corresponding fitness value
  Initialize Algorithm Parameters:
    Set population size, variable dimensions, bounds, and maximum generations.
    Initialize an empty population list and fitness array.
  Initialize Population:
    Create and generate a new individual with given dimensions and bounds for each individual in the population.
  Calculate Fitness:
for each individual in the population do
    Calculate and assign fitness based on given criteria.
end for
  Main Evolution Loop:
    Set initial generation counter $t$ to 0.
    Initialize population and calculate initial fitness.
    Find the best initial individual based on fitness.
while $t < MAXGEN$ do
    for each individual in the population do
        Perform mutation, crossover, and selection operations.
        Replace individuals in the population based on selection results.
    end for
      Evaluate the fitness of the new population.
      Update the best individual if a better one is found.
      Dynamically adjust the scaling factor $F$.
      Increment generation counter $t$.
end while
  Mutation:
    Select three distinct individuals from the population.
    Perform mutation based on these individuals and scaling factor $F$.
  Crossover:
for each variable in an individual do
    Perform crossover based on the crossover rate $CR$.
    Create a trial individual.
end for
  Selection:
    Compare the trial individual with the current individual.
    Select the one with better fitness.
  Print Results:
    Print the final optimal values and the corresponding solution.
```

3.1.2. Temporal attention mechanism

Next, the Temporal Fusion Transformer employs a multi-head attention procedure to catch temporal dependencies.

$$\alpha_i = \text{softmax}\left(\frac{(\mathbf{Q}\mathbf{W}_i^Q)(\mathbf{K}\mathbf{W}_i^K)^\top}{\sqrt{d_k}}\right)$$

where $\mathbf{Q} = \mathbf{E}_{\text{past}}$, $\mathbf{K} = \mathbf{E}_{\text{past}}$, and $\mathbf{V} = \mathbf{E}_{\text{past}}$ are the query, key, and value matrices, respectively, and $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, and $\mathbf{W}_i^V$ are the learned weight matrices for the i-th head.

The outputs of all heads are concatenated and linearly transformed:

$$\mathbf{Z} = \text{Concat}(\alpha_1 \mathbf{V}\mathbf{W}_1^V, \alpha_2 \mathbf{V}\mathbf{W}_2^V, \dots, \alpha_h \mathbf{V}\mathbf{W}_h^V)\mathbf{W}^O$$

3.1.3. Static covariate encoders

Static covariate features are those features that do not change over time such as certain demographic features and geographical location. For the ADE-TFT model, static covariate features are encoded to represent the temporal behavior of the model. The static encoder manages these static covariates and combines them into the model to have an impact on the temporal attention and prediction methods.

$$\mathbf{E}_{\text{static}} = \text{StaticEncoder}(\mathbf{S})$$

The encoded static covariates, $\mathbf{E}_{\text{static}}$, are then combined into the model to influence the attention mechanism and prediction layers.

Algorithm 2 Forecasting bitcoin prices using TFT.

```
Input:
$target$: Target variable to forecast ('close')
$retrain$: List of days to retrain the model
$outputs$: List of days for output predictions
$scaling$: Type of data scaling ('minmax')
$tuned$: Indicator whether hyperparameters are tuned (0 or 1)
$window$: Window size for the input data
Output:
$MAPE$: Mean Absolute Percentage Error
$MSE$: Mean Square Error
$RMSE$: Root Mean Square Error
$result$: Array of predictions
Procedure:
Initialize hyperparameters and data structures for results
Set additional split value for train/test splitting
Initialize lists to store metrics and predictions
for each retraining interval index and days in $retrain$ do
    Define the experiment name based on the parameters
    Load and preprocess the dataset for Bitcoin
    Adjust the data splitting based on the retraining schedule
    Load tuned parameters if required
    Initialize the TFT model with the experiment name
    Prepare the dataset for training
    Normalize the dataset
    Summarize the data configuration
    Extract test data to be predicted
    Train the model and predict the test data
    Inverse transform the predictions to the original scale
    Reshape and scale predictions and labels for evaluation
    Calculate and store evaluation metrics
    Print evaluation metrics
    Extend the lists of all predictions and labels
end for
Return Results:
Print and return MAPE, RMSE, MSE, and prediction results
```

3.1.4. *Gated residual network*

To further enhance the learning experience and capacity, the ADE-TFT model employs a critical component called Gated Residual Network (GRN). To effectively propagate the gradients GRN uses residual connections and gating mechanisms.

$$\mathbf{H} = \mathrm{ReLU}(\mathbf{W}_1\mathbf{X} + \mathbf{b}_1) \odot \sigma(\mathbf{W}_2\mathbf{X} + \mathbf{b}_2) + \mathbf{X}$$

where:

- $\mathbf{X}$ is the input to the GRN.
- $\mathbf{W}_1$ and $\mathbf{W}_2$ are learned weight matrices.
- $\mathbf{b}_1$ and $\mathbf{b}_2$ are bias vectors.
- ReLU is the Rectified Linear Unit activation function.
- σ is the sigmoid activation function.
- $\odot$ denotes element-wise multiplication.

The GRN output, $\mathbf{H}$, combines the transformed input through a ReLU activation and a gated mechanism using a sigmoid function. The residual connection $\mathbf{X}$ allows the original input to be added back to the transformed input, enhancing gradient flow and improving learning dynamics.

3.1.5. *Decoder*

The decoder combines the encoded interpretations from the temporal attention mechanism, static covariates, and the Gated Residual Network (GRN) to produce the final output.

$$\hat{\mathbf{y}}_t = \mathrm{Decoder}(\mathbf{Z}, \mathbf{E}_{\mathrm{future}}, \mathbf{E}_{\mathrm{static}})$$

where:

- $\mathbf{Z}$ is the output from the temporal attention mechanism, representing the combined past embeddings.
- $\mathbf{E}_{\mathrm{future}}$ is the embedding of future known inputs.
- $\mathbf{E}_{\mathrm{static}}$ is the embedding of static covariates.
- Decoder is a function that integrates these embeddings to produce the final prediction $\hat{\mathbf{y}}_t$.

The decoder processes these inputs to predict the target feature at each time step t. The combination of temporal attention outputs, future embeddings, and static covariates enables the model to make accurate and context-aware predictions.

3.1.6. ADE-TFT loss function

Finally, the loss function in the ADE-TFT model finds the difference between the predicted values and the actual values, typically based on error metrics such as Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), and Root Mean Square Error (RMSE). The loss function further guides the ADE-TFT training process by feeding a gradient to adjust the model parameters.

The total loss function $\mathcal{L}$ can be a combination of these error metrics to balance their effects:

$$\mathcal{L} = \alpha \cdot \text{MAPE} + \beta \cdot \text{MSE} + \gamma \cdot \text{RMSE}$$

where α, β, and γ are the weights assigned to each error metric to balance their contributions to the total loss.

3.2. Challenges

Several challenges were encountered while predicting cryptocurrency market movements within the unstable global financial markets. First, it was noticed that the dataset contains the intrinsic volatility of cryptocurrencies due to rapid price fluctuations which complicated the development of an accurate prediction model. It was also noticed that the cryptocurrency market is sensitive to regulatory fluctuations, geopolitical events, and economic trends. Therefore, these factors introduce uncertainty and noise into prediction models. Moreover, market mawkishness, often prompted by social media and news outlets, plays an important role in price changes, forcing clever sentiment analysis tools to portray this dynamic. Finally, the distributed and comparatively undeveloped nature of the cryptocurrency market leads to broken data sources and varying reporting standards, making it challenging to obtain clean and consistent datasets for consideration.

3.3. Regression metrics

Statistical measures that indicate how well a model can predict continuous data, such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE), are used to evaluate a model's performance for regression problems [11]. These measurements are used to determine the precision and accuracy of the model's predictions, and the extent of the unpredictability of the outcome variable can be explained by the independent variables. The utilization of these metrics includes training the regression model, making predictions, and determining MSE, RMSE, and MAPE to measure errors. The best regression fitting model can be selected based on lower values of these metrics.

Mean Squared Error (MSE): It counts the average of the squared anomalies gap among the forecasted outcome and ground truth results.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(actual_i - predicted_i)^2 \tag{6}$$

where n is the number of instances in the dataset, $actual_i$ is the ground truth values of the objective variable for observation i, and $predicted_i$ is the predicted value of the target variable for observation i. Moreover, MSE can be used for hyperparameter tuning, regularization, model cross-validation, bias-variance tradeoff, error analysis and outlier detection.

Root Mean Squared Error (RMSE): It quantifies the square root average of the squared anomaly gap between the forecasted outcome and the ground truth results. It was calculated as the square root of the MSE.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(actual_i - predicted_i)^2} \tag{7}$$

where n is the number of instances in the dataset, $actual_i$ is the ground truth values of the objective variable for observation i, and $predicted_i$ is the predicted value of the target variable for observation i. A low RMSE value indicates that the predicted values are close to the actual values indicating the model's good fit.

Mean Absolute Percentage Error (MAPE): It assesses the average percentage deviation between the forecasted outcome and the ground truth results. It quantified and expressed the proportional difference between the expected and actual values.

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\text{Actual Value}_i - \text{Predicted Value}_i}{\text{Actual Value}_i}\right| \times 100 \tag{8}$$

where n is the number of instances in the dataset, $actual_i$ is the ground truth values of the objective variable for observation i, and $predicted_i$ is the predicted value of the target variable for observation i. A low MAPE value suggests that the trained model prediction is close to the actual Bitcoin prediction, in terms of prediction, indicating acceptable model performance. Moreover, cross-validation can also ensure that the trained model is generic and does not suffer from overfitting.

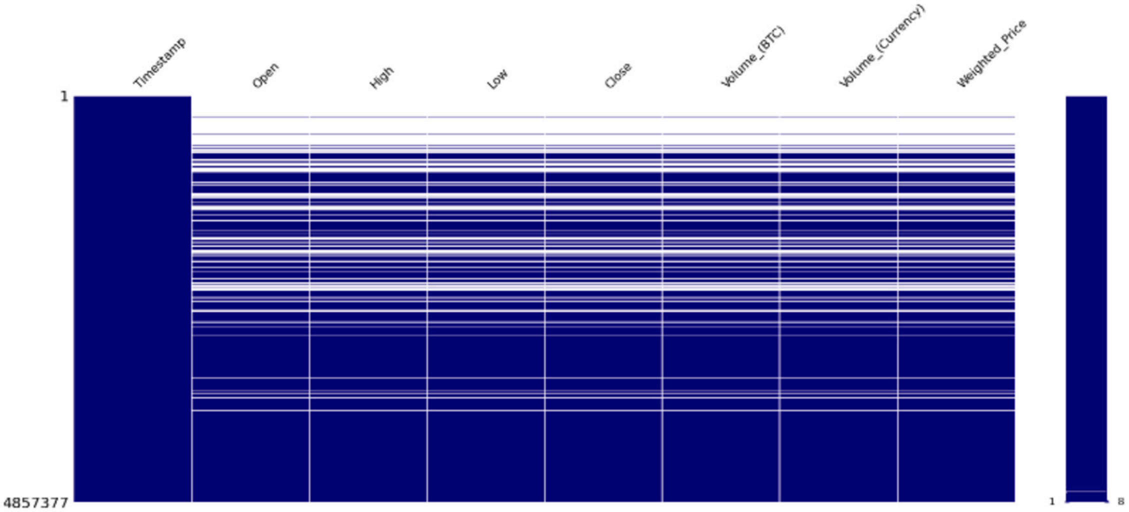


Fig. 4. Missing values.

3.4. Incorporating sentiment analysis into ADE-TFT

Various NLP libraries and tools such as NLTK and SpaCy were used to perform sentiment analysis and incorporate it into the ADE-TFT model. First, the text preprocessing step was performed, separating the text into tokens. Next, the common words that do not contribute to sentiment such as and, is, and the were filtered out. Next, through the process of Lemmatization, the words were reduced to their root form. Using techniques such as polarity scores and aggregating scores, first numerical values were assigned to the sentiment to represent negative, positive and neutral sentiment and then sentiment scores were summarized for a certain period to create time series sentiment data. Next, the aggregated sentiments score was transformed into a form that can be provided to the ADE-TFT model. Next, time series alignment was performed where sentiment attributes were aligned with the corresponding time frame of other input attributes such as volume and price data. Next, sentiment attributes were combined with input attributes to form a complete and comprehensive feature set. Next, TFT was used to assign various weights to sentiment attributes at different time stamps. This allows the ADE-TFT model to focus more on meaningful sentiment alteration with greater market influence. Furthermore, the ADE-TFT model was trained in sentiment attributes and historical market data. Lastly, metrics such as RMSE, MAPE, and MSE were used to evaluate the ADE-TFT model performance.

This study demonstrated that various causes beyond historic price data, market sentiment, and social media trends prompt cryptocurrency markets. Sentiment analysis, which includes parsing and decoding the feelings and thoughts communicated in social media posts and news articles offers a valued understanding of the market's state. By integrating these perceptions, the prediction model can better foresee market changes that are steered by investor sentiment and peripheral events. Furthermore, the study highlights the essential for sophisticated normalization and data preprocessing techniques to process the unpredictability and randomness of cryptocurrency data. Advanced AI techniques are critical for efficiently normalizing and preprocessing this data, guaranteeing that the model is trained on reliable and appropriate information.

4. Experiments and results

The TFT model was trained using historical Bitcoin data from 17-09-2014 to 25-11-2022, which includes 4857377 records with several attributes.

The dataset used for this research shown in the following Fig. 4 has many entries with missing values. Missing values were around 25% which were removed, and the resulting dataset is shown in Fig. 5.

This research study aims to estimate the X_{τ_+1} to X_{τ_max} price of bitcoin using the currently available data. Therefore, the dataset was organized according to date and time feature, and for this $Time_idx$, $Month$, and $Year$ columns were derived from the Timestamp. Fig. 6 shows a box plot to visualize the scattered and skewed data to solve the dataset's outliers issue. The data was changed into a normal distribution where $\mu = 0$ and $\sigma = 1$, by using box-cox transformation and distribution techniques.

To forecast the Weighted_Price feature, we must evaluate its trends and patterns over time. For this, the Weighted_Price graph is plotted in Fig. 7 to observe the behavior and seasonality of the price of Bitcoin in prior years. The Bitcoin price in $\$USD$ was almost stable from 2014-2016 to 2017 it gradually increased from after to 2017-2021. Data were divided into three sets. The training data covered the period from 2014:09 to 2019:12, which is dependent on a 63-month training batch; the validation data covered the period from 2020:01 to 2022:02 with 26 months, and the testing data covered the period from 2022:02 to 2022:11 with 09 months of observations.

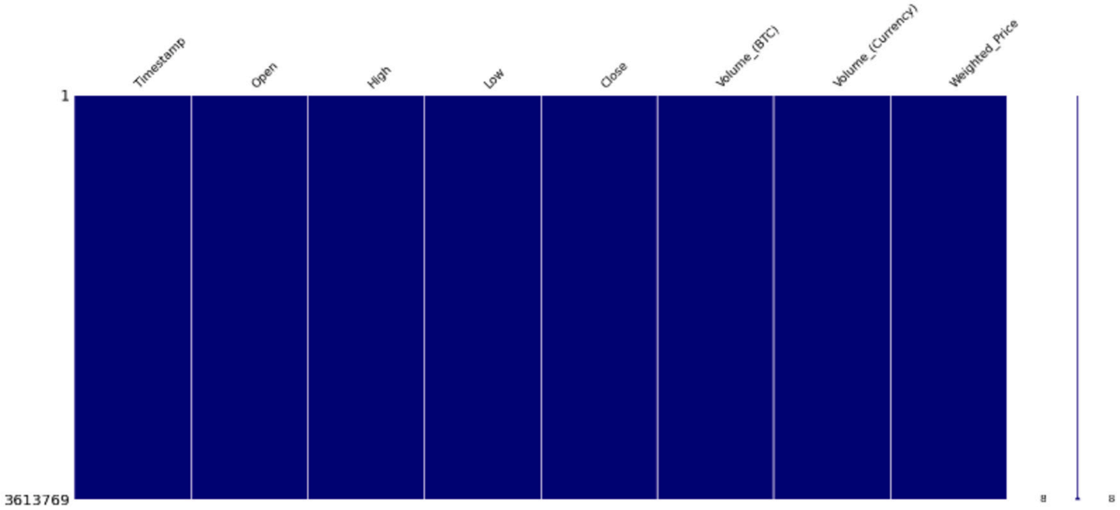


Fig. 5. After removing missing values.

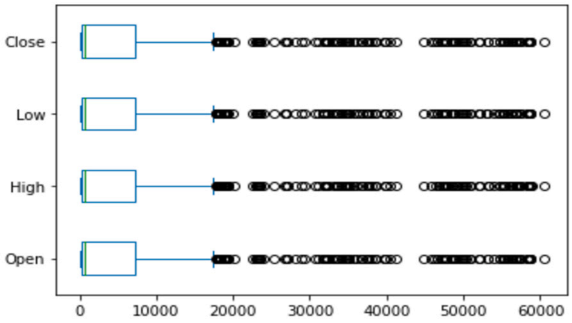


Fig. 6. Outliers in dataset.

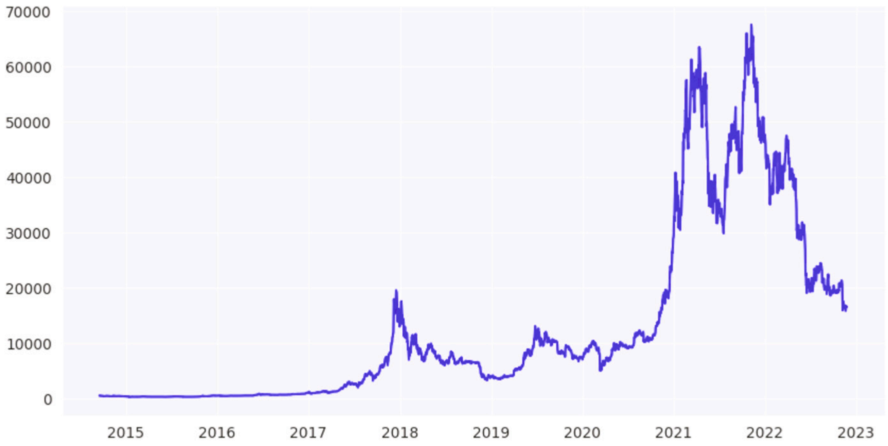
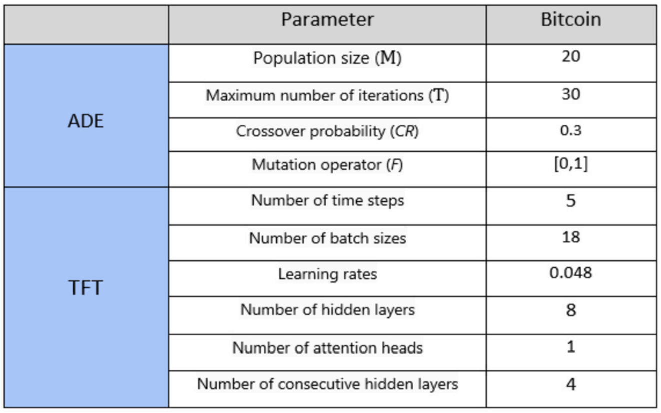


Fig. 7. Trend of weighted price feature over the years.

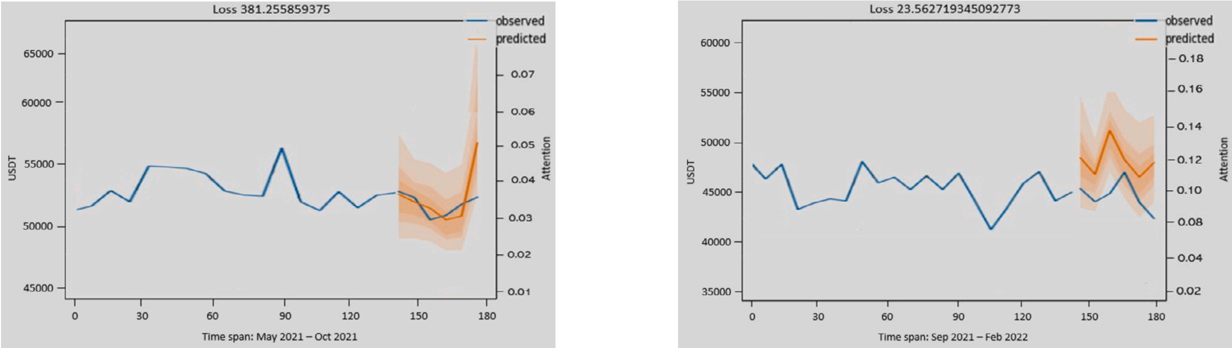
4.1. Hyperparameters selection

The range for TFT model hyperparameters is selected between: The range for batch sizes[5,20]; time steps[2,12]; attention heads[1,4]; hidden layers[2,4,8]; and successive hidden layers[2,4,8]. While keeping all other hyperparameters constant, it is shown in below Fig. 8. Tests on the TFT model were conducted using the hidden layer sizes 2, 4, and 8. In addition, 100 epochs were specified with patience of 15 before the training process was terminated if the model was not converging.



	Parameter	Bitcoin
ADE	Population size (M)	20
	Maximum number of iterations (T)	30
	Crossover probability (*CR*)	0.3
	Mutation operator (*F*)	[0,1]
TFT	Number of time steps	5
	Number of batch sizes	18
	Learning rates	0.048
	Number of hidden layers	8
	Number of attention heads	1
	Number of consecutive hidden layers	4

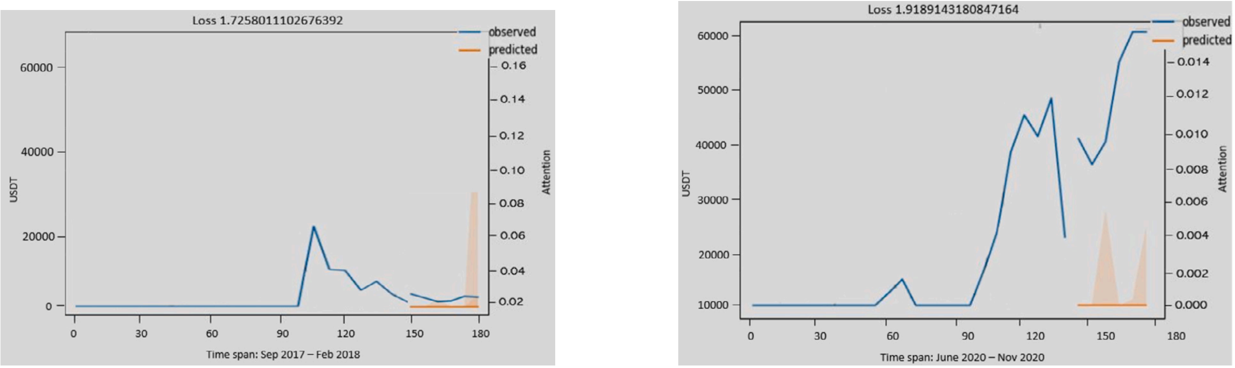
Fig. 8. Hyperparameters of ADE-TFT model.



(a) TFT model performance on validation data from May 2021 to October 2021

(b) TFT model performance on validation data from Sep 2021 to Feb 2022

Fig. 9. TFT model performance on validation data on different dates.



(a) Worst case TFT model performance on Sep 2017 to Feb 2018 data

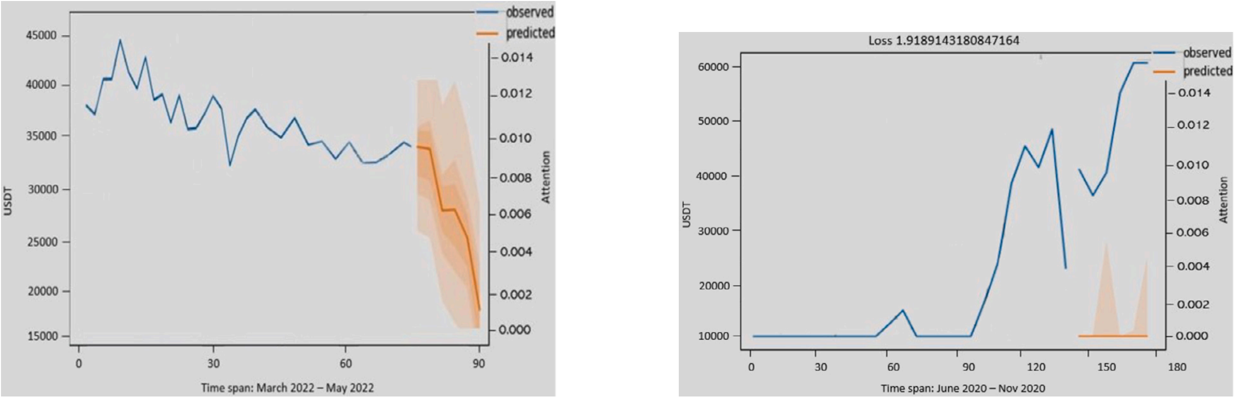
(b) Worst case TFT model performance on Sep 2021 to Feb 2022 data

Fig. 10. TFT model worst-case performance on different dates.

4.2. Performance evaluation

The best model weights were saved after the model was trained on the training dataset. The graphs in Figs. 9a and 9b show some of the prediction quantiles from the validation dataset. Forecasts appeared accurate, and the prediction samples were good enough as the model learned and converged on different points of the time index as fluctuations occurred. The grey lines represent how much weight the model gives to certain periods in time when generating the forecast.

Figs. 10a, 10b show the TFT model learning pattern and its poor convergence, making it an unreliable forecasting model. The reason for poor convergence could be that Bitcoin prices are affected by events such as politics, the supply chain of money, money laundering, the compactness of other currencies, and people's sentiments, which make the TFT model prediction unreliable.



(a) TFT model performance on the testing set for March 2022 to May 2022

(b) TFT model performance on the testing set for Feb 2022 to May 2022

Fig. 11. TFT model performance on testing set.

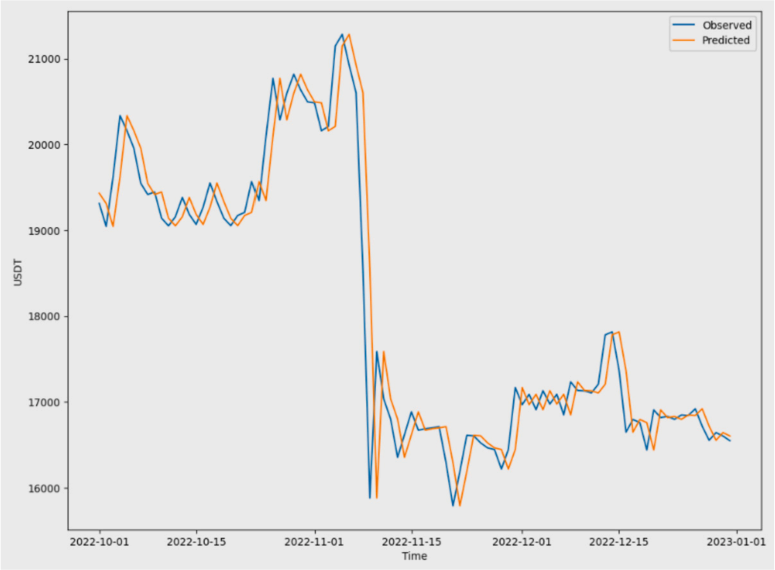


Fig. 12. The performance of a predictive model for USDT prices over time.

4.3. TFT model prediction on testing data

Because the dataset contained covariates and the encoder length was set to the previous 24 months of data when predicting the price, the decoder data were created using forward filing by repeating the covariate features from the most recent known points. The encoder and decoder were integrated to predict the unseen data used as the testing set. The graphs shown in Fig. 11a and Fig. 11b are a few examples of random sample graphs that forecast future prices over different periods.

The performance of the predictive model for USDT prices over time is shown in Fig. 12. The blue line represents the historical prices used to train the model. Next, the correctness of the model was verified using orange unseen data. This alignment suggests a model that is capable of accurately predicting USDT prices, which is crucial for making investment decisions.

Fig. 13 shows how well the trained model predicts the data point and coverage from time to time when compared to the actual values of Bitcoin historical data.

Different hidden layers [2,4,8] of TFT model versions were examined, as indicated in Table 1; however, the 8th hidden layer outperformed the others when combined with additional hyperparameters.

Fig. 14 demonstrate the ADE-TFT model's interpretable results. The explanation is divided into three parts: the relevance of various lag orders, previous inputs' importance, and future variables' importance.

The model identifies patterns and correlations between the input characteristics (encoder variables) and the target variable (decoder variable) using static variables that provide constant information. Fig. 15 shows the importance of variables that are constant throughout the testing phase.

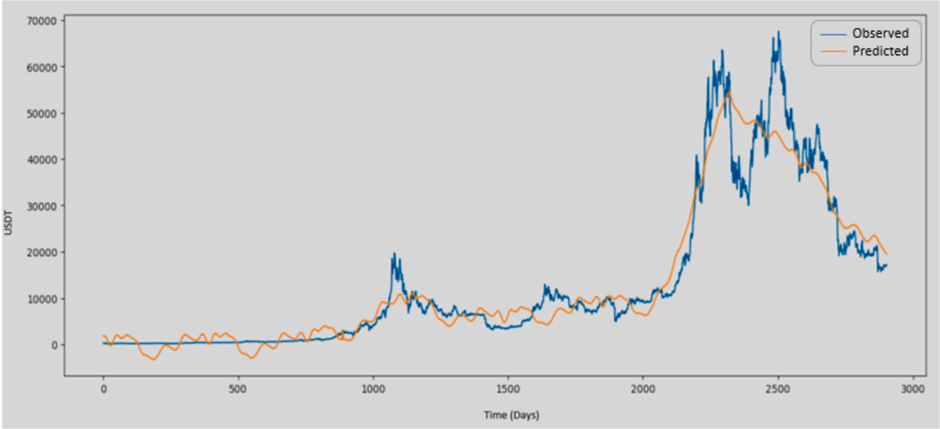


Fig. 13. Comparing model predictions with actual points.

Table 1
Results comparison between the proposed system and the existing models.

Citation	Existing Models	Study Currency	RMSE	MAPE
[4]	Autoregressive integrated moving average (ARIMA), Long Short-Term Memory (LSTM) Gated Recurrent Unit (GRU)	BTC-USDT	302.53 603.68 381.34	42.239 87.413 49.808
[48]	Least Squares Support Vector Machine (LSSVM)	BTC-USD	272.155	37.720
	Neural Networks with Backpropagation		540.087	74.935

Proposed model	ADE-TFT(h = 2)
	ADE-TFT(h = 4)
	ADE-TFT(h = 8)
		BTC-USD	**209.74**
			178.68
			167.12
				31.0104
				29.3295
				23.1734

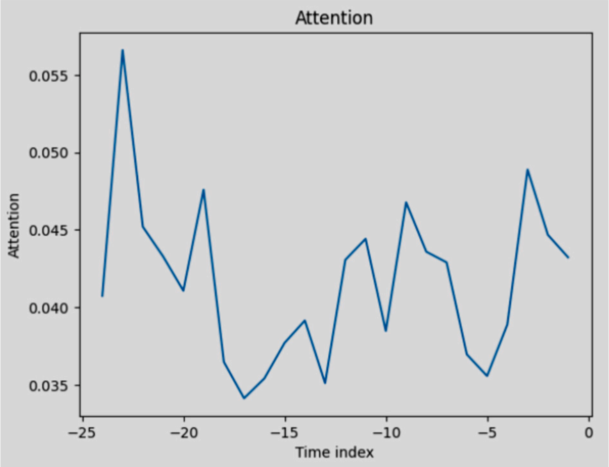


Fig. 14. Attention to various latency orders.

The TFT model results for the price prediction of cryptocurrencies show that more accuracy can be obtained by using the pre-trained model or by training on different hidden layer sizes with different combinations of hyperparameters with more resource consumption. In addition to these deficiencies, the model learns many patterns that converge with more data points and precisely predicts the quantile values.

The ADE-TFT model utilizes deep learning, specifically transformer architectures, to obtain long-term dependencies and dense temporal arrangements in the data, which traditional models rarely achieve. This facilitates the ADE-TFT to successfully understand historical price data and recognize subtle, non-linear relationships that manipulate Bitcoin prices. Secondly, the ADE-TFT model

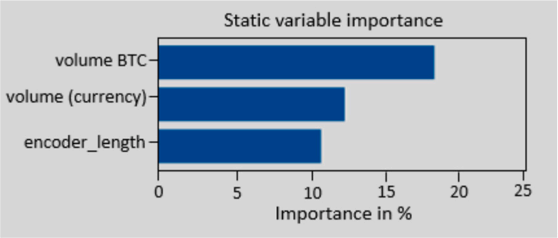


Fig. 15. Importance of static variables.

combines multiple data streams, including geopolitical events, market sentiment, and transactional relationships, allowing for a more widespread analysis of the features shaping Bitcoin values. This multi-source data integration improves the model's power to learn relevant signals from the noise, resulting in more robust predictions. Furthermore, the ADE-TFT model uses an advanced hidden layer configuration (h = 8), which drastically improves its capability to model sophisticated patterns and connections within the data. This depth of representation is put in to decrease prediction errors, as demonstrated by reduced MAPE, MSE, and RMSE values compared to traditional models. Lastly, the model's architecture incorporates advanced normalization techniques that improve its ability to control the irregularities and volatility present in cryptocurrency markets. By utilizing these methods, the ADE-TFT model can adjust to moving market conditions more efficiently than traditional approaches.

5. Conclusion, limitations and future work

Transformer models have been among the most extensively investigated topics. These models are implemented in multiple domains, mostly in time-series prediction or in large-language models. This study presented a complete procedure, from data collection and data cleaning to model training and evaluation. Based on the results in Table 1 the TFT with $h = 8$ performs slightly better than the TFT with a hidden layer size $2, 4$ while setting the other parameters the same. TFT h = 8 excels in producing more precise probabilistic ranges.

This study uses a Temporal Fusion Transformer (TFT) to predict Bitcoin prices. Nonetheless, this study has the following limitations.

1. **Model Complexity and Training Time:** The complexity of the Temporal Fusion Transformer (TFT) model significantly increases the amount of computer resources required for training, particularly when working with a large number of hidden layers. This inhibits the ability of the model to be retrained in response to fresh data and prolongs the development cycle.
2. **Data Quality and Availability:** Although this study has made an effort to include a variety of data points, the model's predictions may be greatly impacted by the quantity and quality of external data, such as news on regulations or market sentiment. The model's ability to completely understand market dynamics may have been restricted by a lack of access to trustworthy and up-to-date data sources.
3. **Generalization Capability:** It was not determined whether the model could be applied to other cryptocurrencies or financial markets. The applicability of the model outside the parameters of this study may be limited by the peculiarities and instability trends associated with Bitcoin, which may not be typical of other assets.

Several improvements could be made to the performance of the ADE-TFT model. Adding additional elements to the dataset, such as market sentiment or regulatory news, can impact cryptocurrency prices. By preprocessing and adding this new data throughout the training procedure, the TFT algorithm can obtain a more comprehensive understanding of market trends. The next enhancement is the possibility that the performance of the TFT model may be affected by the normalization method used. While the standard method for data transformation uses standard z-score scalars, experiments with different normalization techniques may produce superior results. For instance, one may consider employing robust scalars to reduce the impact of outliers in the data on Bitcoin pricing. The model can produce forecasts that are more accurate by reducing the influence of extreme values.

The practical implications of the study include enhanced prediction accuracy, timely insights, informed decision-making, investment portfolio optimization, market volatility and downturn predictions, market analysis, and developing Sentiment-Driven Strategies.

The future work includes the ADE-TFT model expansion to the other Cryptocurrencies, cross-market validation, incorporation of additional features related to cryptocurrencies, advanced sentiment analysis (which includes real-time social media streams and multilingual sentiment analysis), real-time predictions, model optimization and efficient resource management. Moreover, explainable AI can be integrated with ADE-TFT to make it understandable to the common person.

Despite being a fascinating subject, it has been established that predicting the spread between a cryptocurrency and its futures contracts is challenging because of the high volatility and huge amount of noise the time series includes. As a result, this time series requires novel approaches for integrating the initial processing and forecasting algorithms.

CRediT authorship contribution statement

Arslan Farooq: Writing – review & editing, Writing – original draft, Visualization, Methodology, Data curation, Conceptualization. **M. Irfan Uddin:** Methodology, Formal analysis, Data curation, Conceptualization. **Muhammad Adnan:** Methodology, Formal analysis, Conceptualization. **Ala Abdulsalam Alarood:** Funding acquisition, Formal analysis. **Eesa Alsolami:** Writing – review & editing, Project administration, Investigation, Funding acquisition. **Safa Habibullah:** Writing – review & editing, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The support of project number HU24K14859 Hosei University Japan is acknowledged.

Data availability

Data will be made available on request

References

[1] Ratnadip Adhikari, Ramesh K. Agrawal, An introductory study on time series modeling and forecasting, preprint, arXiv:1302.6613, 2013.
[2] Erdinc Akyildirim, Shaen Corbet, Paraskevi Katsiampa, Neil Kellard, Ahmet Sensoy, The development of bitcoin futures: exploring the interactions between cryptocurrency derivatives, Finance Res. Lett. 34 (2020) 101234.
[3] Samina Amin, M. Irfan Uddin, Heyam H. Al-Baity, M. Ali Zeb, Machine Learning Approach for COVID-19 Detection on Twitter, in: Computers, Materials & Continua, Tech Science, 2021, pp. 2231–2247.
[4] Temesgen Awoke, Minakhi Rout, Lipika Mohanty, Suresh Chandra Satapathy, Bitcoin price prediction and analysis using deep learning models, in: Communication Software and Networks: Proceedings of INDIA 2019, Springer, 2020, pp. 631–640.
[5] Lida Barba Maggi, Multiscale forecasting models based on singular values for nonstationary time series, in: III Concurso Latinoamericano de Tesis de Doctorado (CLTD-CLEI)-JAIIO 46 (Córdoba, 2017), 2017.
[6] Ahmed Bouteska, Mohammad Cocco Abedin, Petr Hajek, Kunpeng Yuan, Cryptocurrency price forecasting–a comparative analysis of ensemble learning and deep learning methods, Int. Rev. Financ. Anal. 92 (2024) 103055.
[7] Samina Amin, M. Irfan Uddin, Duaa H. Al-Saeed, Atif Khan, Muhammad Adnan, Early Detection of Seasonal Outbreaks from Twitter Data Using Machine Learning Approaches, in: Complexity, Hindawi, 2021, pp. 1–12.
[8] George E.P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, Greta M. Ljung, Time Series Analysis: Forecasting and Control, Wiley, Hoboken, NJ, 2008.
[9] Francis Breedon, Angelo Ranaldo, Intraday patterns in fx returns and order flow, J. Money Credit Bank. 45 (5) (2013) 953–965.
[10] Francisco M. Caldas, Cláudia Soares, A temporal fusion transformer for long-term explainable prediction of emergency department overcrowding, preprint, arXiv:2207.00610, 2022.
[11] Davide Chicco, Matthijs J. Warrens, Giuseppe Jurman, The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation, PeerJ Comput. Sci. 7 (2021) e623.
[12] Lucas de Azevedo Takara, André Alves Portela Santos, Viviana Cocco Mariani, Leandro dos Santos Coelho, Deep reinforcement learning applied to a sparse-reward trading environment with intraday data, Expert Syst. Appl. 238 (2024) 121897.
[13] Ganesh Chandra Deka, Omprakash Kaiwartya, Pooja Vashisth, Priyanka Rathee, in: Applications of Computing and Communication Technologies: First International Conference, ICACCT 2018, Delhi, India, March 9, 2018, Revised Selected Papers, vol. 899, Springer, 2018.
[14] V. Derbentsev, V. Babenko, Kirill Khrustalev, Hanna Obruch, Sofiia Khrustalova, Comparative performance of machine learning ensemble algorithms for forecasting cryptocurrency prices, Int. J. Eng. 34 (1) (2021) 140–148.
[15] Renato Di Lorenzo, Basic Technical Analysis of Financial Markets, Springer, 2013.
[16] Grzegorz Dudek, Piotr Fiszeder, Paweł Kobus, Witold Orzeszko, Forecasting cryptocurrencies volatility using statistical and machine learning methods: a comparative study, Appl. Soft Comput. 151 (2024) 111132.
[17] Sheng Fang, Guangxi Cao, Paul Egan, Forecasting and backtesting systemic risk in the cryptocurrency market, Finance Res. Lett. 54 (2023) 103788.
[18] Athanasios P. Fassas, Stephanos Papadamou, Alexandros Koulis, Price discovery in bitcoin futures, Res. Int. Bus. Finance 52 (2020) 101116.
[19] Guoce Feng, Lei Zhang, Feifan Ai, Yirui Zhang, Yupeng Hou, An improved temporal fusion transformers model for predicting supply air temperature in high-speed railway carriages, Entropy 24 (8) (2022) 1111.
[20] Pradeep Hewage, Ardhendu Behera, Marcello Trovati, Ella Pereira, Morteza Ghahremani, Francesco Palmieri, Yonghuai Liu, Temporal convolutional neural (tcn) network for an effective weather forecasting using time-series data from the local weather station, Soft Comput. 24 (2020) 16453–16482.
[21] Hassane Hotait, Xavier Chiementin, Lanto Rasolofondraibe, Intelligent online monitoring of rolling bearing: diagnosis and prognosis, Entropy 23 (7) (2021) 791.
[22] R.J. Hyndman, G. Athanasopoulos, Forecasting: Principles and Practice [Internet], Otexts, Melbourne, Australia, 2021.
[23] Param Jeet, Prashant Vats, Learning Quantitative Finance with R, Packt Publishing Ltd, 2017.
[24] Qiang Jiang, Chenglin Tang, Chen Chen, Xin Wang, Qing Huang, Stock price forecast based on lstm neural network, in: Proceedings of the Twelfth International Conference on Management Science and Engineering Management, Springer, 2019, pp. 393–408.
[25] Deepak Kumar, S.K. Rath, Predicting the trends of price for Ethereum using deep learning techniques, in: Artificial Intelligence and Evolutionary Computations in Engineering Systems, Springer, 2020, pp. 103–114.
[26] Kin Keung Lai, Lean Yu, Shouyang Wang, Wei Huang, Hybridizing exponential smoothing and neural network for financial time series predication, in: Computational Science–ICCS 2006: 6th International Conference, Reading, UK, May 28-31, 2006, Proceedings, Part IV 6, Springer, 2006, pp. 493–500.
[27] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, Xifeng Yan, Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting, Adv. Neural Inf. Process. Syst. 32 (2019).
[28] Yang Li, Zibin Zheng, Hong-Ning Dai, Enhancing bitcoin price fluctuation prediction using attentive lstm and embedding network, Appl. Sci. 10 (14) (2020) 4872.

[29] Zhixi Li, Vincent Tam, Combining the real-time wavelet denoising and long-short-term-memory neural network for predicting stock indexes, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2017, pp. 1–8.

[30] Ioannis E. Livieris, Emmanuel Pintelas, Stavros Stavroyiannis, Panagiotis Pintelas, Ensemble deep learning models for forecasting cryptocurrency time-series, Algorithms 13 (5) (2020) 121.

[31] Miguel López Santos, Xela García-Santiago, Fernando Echevarría Camarero, Gonzalo Blázquez Gil, Pablo Carrasco Ortega, Application of temporal fusion transformer for day-ahead pv power forecasting, Energies 15 (14) (2023) 5232.

[32] Esam Mahdi, Víctor Leiva, Saed Mara'Beh, Carlos Martin-Barreiro, A new approach to predicting cryptocurrency returns based on the gold prices with support vector machines during the covid-19 pandemic using sensor-related data, Sensors 21 (18) (2021) 6319.

[33] David Meijer, Predicting cryptocurrency price trends with long short-term memory, PhD thesis, Tilburg University, 2020.

[34] Mehedi Hasan Mishal, Nura Jannat Rakhi, Fahmida Rashid, Kawsar Hamid, Md Kishor Morol, Abdullah Al Jubair, Dip Nandi, Prediction of cryptocurrency price using machine learning techniques and public sentiment analysis, in: 2022 25th International Conference on Computer and Information Technology (ICCIT), IEEE, 2022, pp. 657–662.

[35] Shahzad Muzaffar, Afshin Afshari, Short-term load forecasts using lstm networks, Energy Proc. 158 (2019) 2922–2927.

[36] Erik Persson, Forecasting Efficiency in Cryptocurrency Markets: A Machine Learning Case Study, 2022.

[37] Emmanuel Pintelas, I.E. Livieris, Stavros Stavroyiannis, Theodore Kotsilieris, P. Pintelas, Fundamental research questions and proposals on predicting cryptocurrency prices using dnns, Techincal Report, 2020.

[38] Eetu Pulkkinen, Forecasting emergency department arrivals with neural networks, B.S. thesis, 2020.

[39] Lekkala Sreekanth Reddy, P. Sriramya, A research on bitcoin price prediction using machine learning algorithms, Int. J. Sci. Technol. Res. 9 (4) (2020) 1600–1604.

[40] Gregory C. Reinsel, Elements of Multivariate Time Series Analysis, Springer Science & Business Media, 2003.

[41] Gabriel Trierweiler Ribeiro, André Alves Portela Santos, Viviana Cocco Mariani, Leandro dos Santos Coelho, Novel hybrid model based on echo state neural network applied to the prediction of stock price return volatility, Expert Syst. Appl. 184 (2021) 115490.

[42] Mohammad Valipour, Mohammad Ebrahim Banihabib, Seyyed Mahmood Reza Behbahani, Parameters estimate of autoregressive moving average and autoregressive integrated moving average models and compare their ability for inflow forecasting, J. Math. Stat. 8 (3) (2012) 330–338.

[43] Craig Warmke, What is bitcoin, Inquiry 67 (1) (2024) 25–67.

[44] Billy M. Williams, Multivariate vehicular traffic flow prediction: evaluation of arimax modeling, Transp. Res. Rec. 1776 (1) (2001) 194–200.

[45] Binrong Wu, Lin Wang, Yu-Rong Zeng, Interpretable wind speed prediction with multivariate time series and temporal fusion transformers, Energy 252 (2023) 123990.

[46] Ye Yang, Jiangang Lu, A fusion transformer for multivariable time series forecasting: the mooney viscosity prediction case, Entropy 24 (4) (2022) 528.

[47] Jesse Yli-Huumo, Deokyoon Ko, Sujin Choi, Sooyong Park, Kari Smolander, Where is current research on blockchain technology?—a systematic review, PLoS ONE 11 (10) (2016) e0163477.

[48] Shengao Zhang, Mengze Li, Chunxiao Yan, The empirical analysis of bitcoin price prediction based on deep learning integration method, Comput. Intell. Neurosci. (2022) 2022.

[49] Huali Zhao, Martin Crane, Marija Bezbradica, Attention! Transformer with Sentiment on Cryptocurrencies Price Prerediction, 2022.