Expert Systems With Applications 255 (2024) 124404

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Probabilistic deep learning and transfer learning for robust cryptocurrency price prediction

Amin Golnari [a,*], Mohammad Hossein Komeili [b], Zahra Azizi [c]

[a] Faculty of Electrical & Robotics, Shahrood University of Technology, Shahrood, Iran
[b] Faculty of Mathematics & Computer Science, Shahid Beheshti University, Tehran, Iran
[c] Department of Computer Engineering, University of Afarinesh, Borujerd, Iran

ARTICLE INFO

Keywords:
BTC price prediction
Cryptocurrency prediction
Probabilistic gated recurrent units (P-GRU)
Bayesian neural networks (BNNs)
Transfer learning

ABSTRACT

Forecasting the price of Bitcoin (BTC) with precision is a complex endeavor, given the market's inherent uncertainty and volatility, influenced by a diverse range of parameters. This research is driven by the central goal of introducing a specialized deep learning model tailored to predict digital currency prices, with a specific emphasis on BTC. To address this challenge, a pioneering strategy has been established, leveraging probabilistic gated recurrent units (P-GRU). This approach integrates probabilistic attributes into the model, facilitating the generation of probability distributions for projected values. The effectiveness of this method is assessed using one year of BTC price history, sampled at a five-minute interval. In parallel, a comparative analysis is conducted against alternative models, including GRU, long short-term memory (LSTM), and variants thereof (time-distributed, bidirectional, and simple models). In pursuit of optimizing model efficacy, a bespoke callback mechanism is deployed. This callback, driven by R2-score tracking, captures optimal model weights based on validation data. Moreover, a transfer learning paradigm is adopted to broaden the study's horizons. A pre-trained model on BTC data is harnessed to predict prices for six other prominent cryptocurrencies: Ethereum, Litecoin, Tron, Polkadot, Cardano, and Stellar. Consequently, a distinct model is tailored for each cryptocurrency. The outcomes of this investigation conclusively underscore the superior performance of the proposed methodology. In the midst of a volatile and uncertain market landscape, the proposed approach outshines its counterparts, showcasing an enhanced ability for cryptocurrency price forecasting.

1. Introduction

Nowadays, Bitcoin (BTC), as originally proposed by Nakamoto (2008), has become a popular investment choice, marked by heightened market volatility. Predicting cryptocurrency prices is crucial for effective asset management, given the decentralized nature and limited supply of BTC (Chang, Lee, Yang, & Lu, 2023). However, its rapid value fluctuations make accurate prediction challenging, requiring consideration of market uncertainty (Tsantekidis et al., 2020).

In cryptocurrency price prediction, machine learning (ML) methods are common, with recent studies using XGBoost and artificial neural networks (ANNs) (Drahokoupil, 2022; Gradojevic, Kukolj, Adcock, & Djakovic, 2023; Ranjan, Kayal, & Saraf, 2023). In recent years, deep learning (DL) techniques have displayed promising outcomes in BTC price prediction. The capacity of DL models to encapsulate intricate data patterns and interdependencies makes them eminently suitable for time-series forecasting endeavors (Ahmed, Atiya, Gayar, & El-Shishiny, 2010). Innovative strides have been made in amalgamating neural network models, as evidenced by recent research wherein two separate MLP-based neural networks were combined to devise a more resilient predictive system (Rajabi, Roozkhosh, & Farimani, 2022).

This study introduces a probabilistic DL method employing GRU to project cryptocurrency prices. The approach captures temporal relationships in price data, providing a probabilistic estimation of future price. The inherent uncertainty in predictions is crucial for effective risk management in investment and trading. Cryptocurrency markets are known for their high volatility and sensitivity to various factors such as market sentiment, regulatory developments, technological changes, and macroeconomic trends. These factors introduce a level of uncertainty into the price movements of cryptocurrencies, making it challenging to make accurate and deterministic predictions. Therefore, when discussing uncertainty in prediction, we are highlighting an

* Corresponding author.
E-mail addresses: amingolnarii@gmail.com (A. Golnari), mhkomeili98@gmail.com (M.H. Komeili), iazizizahraa@gmail.com (Z. Azizi).

https://doi.org/10.1016/j.eswa.2024.124404
Received 18 August 2023; Received in revised form 13 March 2024; Accepted 3 June 2024
Available online 12 June 2024

acknowledgment and explicit consideration of the fluctuating nature of the cryptocurrency market.

The integration of probabilistic elements results in a probability distribution for predicted values, enhancing accuracy in the capricious cryptocurrency market. The probabilistic GRU (P-GRU) framework demonstrates improved accuracy, offering a reliable prediction mechanism. Evaluation involves seven cryptocurrency price histories from May 16, 2022, to May 17, 2023, comparing P-GRU with alternative models like GRU, LSTM, and probabilistic LSTM (P-LSTM). Throughout the training process, we actively monitor the R2-score on the validation dataset and choose the model based on its performance on this independent dataset. This strategy is designed to address overfitting concerns by ensuring the model's ability to generalize effectively to new unseen data besides capturing the most appropriate model weights. Overfitting, characterized by the model's sensitivity to training data idiosyncrasies and subsequent failure to generalize, is mitigated by prioritizing performance on the validation data. In this approach, we aim to achieve the best result by monitoring the R2-score on validation data, which may not necessarily coincide with minimizing the loss function.

2. Background

The adoption of blockchain technology has brought about increased security and decentralization in the domain of digital currency, yet it is also accompanied by a host of security and privacy concerns (Wen et al., 2023). These concerns intersect with the broader context of digital currency trading, which operates in a global arena characterized by substantial price fluctuations and volatility. However, the global trading nature of digital currency exposes it to substantial price fluctuations and volatility, thus rendering it a comparatively riskier investment avenue when juxtaposed with stocks. Consequently, a growing number of investors are actively seeking advanced models and techniques to enhance their comprehension and predictive capabilities in the realm of digital currency prices.

In response to these challenges and the demand for improved predictive accuracy, researchers have turned to the power of deep learning techniques for cryptocurrency price prediction. A notable study (Liu, Li, Li, Zhu, & Yao, 2021) offers a comprehensive compilation of factors that exert influence on BTC's price, categorized into three distinct groups: (1) the digital currency market, encompassing metrics like BTC's daily turnover, total market capitalization, hash rate, and market data from other digital currencies; (2) public attention, measured by search volumes of BTC-related keywords across various search engines; and (3) the macroeconomic landscape, encompassing stock market indices, crude oil prices, exchange rates, and other pertinent economic indicators. Similarly, Patel, Tanwar, Gupta, and Kumar (2020) have found that DL models, including LSTM and bidirectional GRU (Bi-GRU), effectively predict BTC price using regression analysis. These findings underscore the potential of ML models in the domains of investment, finance, and portfolio management.

ANNs are statistical models that adapt their parameters to make accurate predictions by leveraging complex architectures and cost functions for various tasks. Drawing upon the power of ANNs' adaptability and predictive capabilities, several studies have explored their application in price prediction and financial scenarios. In McNelis (2005), a diverse range of ANN applications in finance is presented, while an initial exploration of econometric applications is found in Kuan and White (1994). In the study conducted by Mallqui and Fernandes (2019), the authors predict BTC's price and its trend, addressing both classification and regression problems through the use of diverse ML algorithms, including RNNs, tree classifier, and the support vector machines (SVMs) algorithm. In a different study carried out by Huang, Nakamori, and Wang (2005), the precision of predicting financial movement direction using SVM is scrutinized and contrasted with alternative classification techniques. The results reveal that SVM surpasses the performance of linear discriminant analysis, quadratic discriminant analysis, and Elman backpropagation neural networks (EBNN) in anticipating the weekly movement direction of the NIKKEI 225 index. The research introduces a hybrid model that fuses SVM with other classification methods, yielding effective results among various forecasting approaches.

In another investigation (Chen, Li, & Sun, 2020), a diverse range of ML techniques was harnessed to forecast the price of BTC. This endeavor involved leveraging an extensive set of high-dimensional attributes encompassing property and network characteristics, trading and market data, attention metrics, and the gold spot price. For the task of daily price prediction, logistic regression and linear discriminant analysis were deployed, drawing on a plethora of features. Additionally, for 5-minute intervals, the investigation utilized a spectrum of models, including random forest, XGBoost, quadratic discriminant analysis, SVM, and LSTM. The outcomes underscore the efficacy of machine learning models, particularly LSTM, which exhibited heightened accuracy compared to statistical methodologies. In Hamayel and Owda (2021), three types of RNN algorithms were proposed to predict the price of three digital currencies: BTC, Ethereum (ETH), and Litecoin (LTC). These models demonstrated strong predictions based on the percentage of average absolute error. Notably, the results indicated that the GRU outperformed LSTM and bidirectional LSTM (Bi-LSTM) models, delivering the most accurate forecasts closely aligned with the actual prices for these three currencies. Hewamalage, Bergmeir, and Bandara (2021) delve into the utilization of RNN for the prediction of time series data. The paper acknowledges that while RNNs are strong contenders in forecasting, traditional statistical models like exponential smoothing (ETS) and autoregressive integrated moving averages (ARIMA) are still popular due to their accuracy and user-friendliness. The paper recommends using RNNs for forecasting but notes that they have limitations and may not be suitable in all cases.

Time series models that utilize Bayesian inference provide precise predictive distributions for data beyond the observed samples, accounting for the inherent uncertainty in the model parameters (Geweke & Amisano, 2010). In another exploration (Sujatha, Mareeswari, Chatterjee, Abd Allah, & Hassanien, 2021), three training algorithms (Bayesian, regularization, Levenberg–Marquardt, and scaled conjugate gradient algorithms) were harnessed to train a neural network. The results, which encompassed error histograms and regression plots, demonstrate the superiority of the Bayesian regular neural network, offering a more precise prediction. Bayesian optimization, a sequential design strategy for optimizing global functions with unknown internal structures, was employed. In Pour, Jafari, Lashgari, Rabiee, and Ahmadisharaf (2022), an approach for predicting BTC price is introduced, involving a hybrid ANN model that combines LSTM and Bayesian optimization. Alonso-Monsalve, Suárez-Cetrulo, Cervantes, and Quintana (2020) offers empirical evidence of the utility of convolutional neural networks (CNNs) for intraday trend classification, including a benchmark hybrid CNN-LSTM model, CNN, MLP, and radial basis function (RBF) neural network, encompassing major cryptocurrencies such as BTC, Dash, Ether, Litecoin, Monero, and Ripple.

Another study by Oyedele, Ajayi, Oyedele, Bello, and Jimoh (2023) presents a benchmark of artificial intelligence and ML techniques for forecasting cryptocurrency prices, including genetic algorithms, boosted tree-based techniques, and DL models such as CNN and GRU. In fact, several studies have delved into the utilization of ML for predicting cryptocurrency prices. McNally, Roche, and Caton (2018) explore machine learning techniques for predicting Bitcoin's price in USD. They use a Bayesian optimized RNN and LSTM network, comparing them to the ARIMA model. The LSTM achieves the best accuracy, showing that deep learning methods outperform ARIMA for forecasting. Kim et al. (2016) introduced a machine-learning-based method for predicting fluctuations in cryptocurrency transactions, leveraging user comments and replies. Apart from forecasting BTC prices, researchers have also delved into the realm of BTC exchange rates. In a study by Chen, Xu, Jia, and Gao (2021), they adopted an ML methodology for predicting exchange rates. Their investigation unfolded in two stages: in the initial phase, utilizing a dual nonlinear feature selection strategy encompassing an ANN and random forest to streamline the array of potential predictors, where they gauged the relevance of economic and technological factors. In the subsequent phase, the identified predictors were seamlessly integrated into the LSTM framework, with the goal of accurately predicting the BTC exchange rate.

Through an in-depth exploration of various researchers' works, this section establishes the context and significance of the present research, while the insights garnered from the reviewed literature serve as a guiding framework for the methodology and analysis employed in this study. By leveraging the capabilities of advanced models, this study aims to enhance our understanding of digital currency's intricate price dynamics and mitigate the inherent risks associated with its volatile nature. For a visual representation of the datasets used in this study, see Fig. 2.

3. Preliminaries

3.1. Bayesian neural networks

Bayesian neural networks (BNNs) represent a distinct form of ANN that seamlessly incorporates Bayesian inference into the training process. BNNs provide a framework for comprehending, quantifying, and estimating the inherent uncertainty associated with predictions made by deep neural networks, offering valuable insights for decision-making processes (Jospin, Laga, Boussaid, Buntine, & Bennamoun, 2022). In contrast to CNNs that consider network weights and biases as fixed values acquired through backpropagation, BNNs treat these parameters as stochastic variables, each governed by its own probability distribution. This approach facilitates the integration of uncertainty into the model, proving particularly advantageous when dealing with incomplete or noisy data (Meng, Babaee, & Karniadakis, 2021). Throughout the training, BNNs employ Bayesian inference to iteratively update the probability distributions associated with weights and biases, leveraging the observed data to refine their estimates. When applied to cryptocurrency price prediction, BNNs regard the price itself as a random variable characterized by its own probability distribution.

The output of a BNN does not yield a solitary point estimate for the price but rather furnishes a probability distribution encompassing the spectrum of conceivable price values. By interpreting the learned parameters of the MLP, the BNN assumes the role of a reliable tool for forecasting cryptocurrency prices with a certain level of confidence, even amidst scenarios marked by partial or inaccurate data. This can significantly assist traders and investors in making well-informed decisions, grounded in the most probable future price fluctuations of a cryptocurrency. For a dataset denoted as $D = (x_i, y_i)$, where N observations are present, the likelihood function encapsulates the likelihood of observing the given data contingent on the model's parameters. Specifically, in the case of a neural network introducing Gaussian noise, the likelihood function can be expressed as:

$$P(D|w) = \prod_{i=1}^{N} G(y_i; f(x_i, w), \sigma^2) \tag{1}$$

where, $f(x_i, w)$ is the output of the neural network for input x_i with parameters w, and σ^2 is the noise variance. Using Bayes' theorem, we can compute the posterior distribution of the model's parameters given the observed data:

$$P(w|D) = \frac{P(D|w)P(w)}{P(D)} \tag{2}$$

where, $P(w|D)$ is the posterior distribution, $P(D|w)$ is the likelihood function, $P(w)$ is the prior distribution, and $P(D)$ is the marginal likelihood (also known as the evidence). Once the BNN is trained, it can be used to make predictions on new input ẋ. This predictive distribution is obtained by integrating the likelihood of the predicted output over the parameters, given the parameters and the posterior distribution:

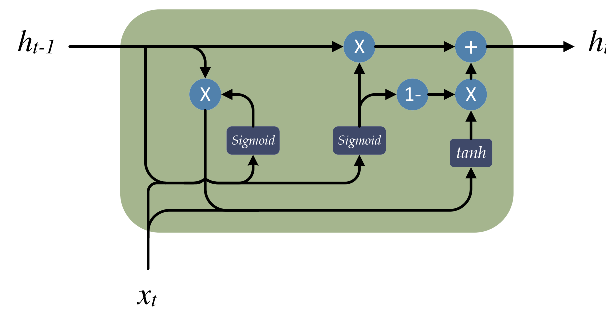


Fig. 1. GRU cell architecture.

$$P(\acute{y}|\acute{x}, x, y) = \int P(\acute{y}|\acute{x}, w)P(w|x, y)dw \tag{3}$$

where, ý is the predicted output, ẋ is the new input, x and y are the training data, and the integral is taken over all possible values of the model's parameters w. In addition to the BNN model, the posterior and predictive distributions, the negative log-likelihood (NLL) is also an important component of BNN training. The NLL is used as the loss function during training and is defined as:

$$-\log P(y|x, w) = -\log \int P(y|x, w)P(w|x, y)dw \tag{4}$$

The NLL serves as a gauge to quantify the dissimilarity between the projected output from the BNN and the actual output. This NLL value plays a pivotal role in adjusting the probability distributions associated with the network's weights and biases during the training phase. The process of minimizing the NLL is executed through methodologies such as stochastic gradient descent (SGD) or other comparable optimization algorithms. The gradients of the NLL in relation to the model's parameters are computed via backpropagation throughout the BNN.

3.2. Probabilistic gated recurrent unit networks

P-GRU networks enhance the conventional GRU (Cho et al., 2014), an RNN similar to LSTM (Hochreiter & Schmidhuber, 1997), by incorporating uncertainty into the model's framework. This entails treating the network's weights as stochastic entities, endowed with their own unique probability distributions. By infusing uncertainty, P-GRUs exhibit enhanced resilience in handling noisy or partial data, thereby yielding more robust predictions when confronted with indeterminacy. In the process of training, P-GRUs harness the principles of Bayesian inference to iteratively refine the probability distributions associated with the weights, iteratively updating them based on the observed data. This nuanced approach empowers the model not only to learn the most probable weight values but also to account for the inherent uncertainty that envelops these estimates. Fig. 1 illustrates a conventional GRU cell architecture. The equations governing the behavior of a GRU cell are provided below:

$$z_t = \sigma(w_z[h_{t-1}, x_t] + b_z) \tag{5}$$

$$r_t = \sigma(w_r[h_{t-1}, x_t] + b_r) \tag{6}$$

$$\tilde{h}_t = \tanh(w_z[r_{t-1} \odot h_{t-1}, x_t] + b_h) \tag{7}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{8}$$

where, σ is the sigmoid activation function, $\odot$ denotes element-wise multiplication, x_t is the input at time step t, h_{t-1} is the hidden state at the previous time step, and z_t, r_t, w. and b. are the single update gate, reset gate, weight matrices, and bias vectors, respectively.

The integration of P-GRUs can be achieved by introducing a distribution layer to the model's architecture, responsible for characterizing

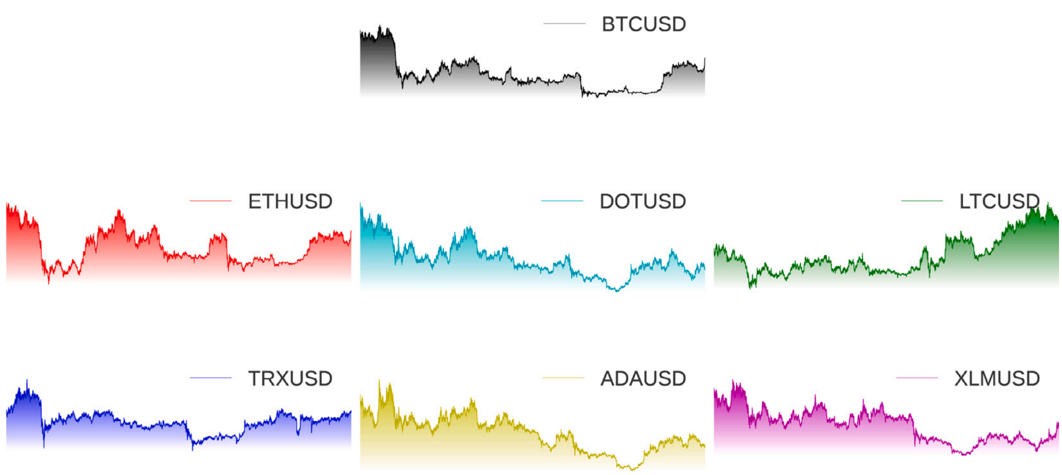


Fig. 2. Cryptocurrency price fluctuations over a span of one year (from May 16, 2022, to May 17, 2023).

the probability distribution governing the model's output, contingent upon the hidden states. The selection of this distribution layer (it can be Gaussian, Bernoulli, or tailored to the specific task) is contingent upon the context at hand. During training, the model optimizes its parameters to maximize the likelihood of observing the given data within the context of the output's probability distribution. In inference mode, the model can draw samples from the probability distribution to generate its predictions.

The mathematical formulations underpinning P-GRUs parallel those of the standard GRU equations, with a notable augmentation involving the incorporation of random variables that encapsulate the model's uncertainty. These random variables are frequently drawn from Gaussian distributions, characterized by a mean of zero and learned variances. Subsequently, the model's output undergoes processing through the distribution layer, allowing for the modeling of the output's probability distribution, predicated on the underlying hidden states. At the core of our proposed P-GRU model resides the Gaussian distribution, distinguished by its distinctive probability density function (PDF):

$$f_x = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{9}$$

where, μ symbolizes the average, while σ signifies the measure of dispersion. By framing the GRU's output as conforming to a Gaussian distribution, we gain the ability to gauge both the central tendency and the extent of spread within the projected values. To facilitate the training of our model, we employ the principle of maximum likelihood estimation (MLE). This method revolves around identifying the distribution parameters that optimize the likelihood function. The essence of MLE revolves around discerning the parameters that yield the highest likelihood of observing the actual data given the model's constructs. In contrast, Bayesian methodologies encompass the process of revising prior assumptions concerning the parameters based on the observed data, resulting in a posterior distribution of these parameters.

Considering a dataset comprising N instances, let y_i denote the authentic target value at the i_{th} observation. Further, let μ_i and σ_i denote the anticipated average and standard deviation, respectively, of the Gaussian distribution corresponding to the i_{th} observation. The likelihood function underpinning the Gaussian distribution manifests as follows:

$$L(\theta) = \prod_{i=1}^{N} \frac{1}{\sigma_i\sqrt{2\pi}} e^{\frac{-(y_i-\mu_i)^2}{2\sigma_i^2}} \tag{10}$$

Taking the natural logarithm of the likelihood function, we get the log-likelihood function:

$$\ln L(\theta) = \sum_{i=1}^{N} \ln\left(\frac{1}{\sigma_i\sqrt{2\pi}}\right) - \frac{(y_i-\mu_i)^2}{2\sigma_i^2} \tag{11}$$

The negative log-likelihood loss function is the negative of the log-likelihood function:

$$NLL(\theta) = -\ln L(\theta) = \sum_{i=1}^{N} \ln(\sigma_i\sqrt{2\pi}) + \frac{(y_i-\mu_i)^2}{2\sigma_i^2} \tag{12}$$

Minimizing the NLL loss function is equivalent to maximizing the likelihood function, which allows us to estimate the parameters of the Gaussian distribution (μ and σ) at each time step.

4. Transfer learning

Transfer learning is a methodology for creating high-performance learners trained with data from different domains, which is necessary when training data is expensive or difficult to collect (Weiss, Khoshgoftaar, & Wang, 2016). Transfer learning for knowledge acquisition represents a cutting-edge ML technique that capitalizes on the insights gained during model training for one task, thereby amplifying the overall proficiency of the model in a closely related yet distinct task. This technique harnesses the accumulated wisdom of pre-trained model iterations as a foundational basis for embarking on novel tasks. Through specialized training on task-specific sub-datasets, refined iterations of the model are crafted. This transfer learning approach, while still relatively young in the field, holds immense promise.

It wields the remarkable potential to substantially curtail the magnitude of data and computational resources necessitated for training a fresh model, concurrently bolstering the model's precision. The versatility of transfer learning is underscored by its successful implementation across a vast array of applications, encompassing domains such as computer vision, natural language processing, and speech recognition. Consequently, it has ascended as a pivotal instrument in the arsenal for constructing ML models that exhibit exceptional overall performance.

To employ this approach, we initiated by training models on the BTC dataset and subsequently identified the most effective one. In the subsequent phase, we utilized the best model by freezing all layers except the final one. This methodology involves the fixation of layers to extract optimal features from the input data, while concurrently training the last layer using a new dataset to generate the most accurate output based on the gleaned features from the frozen layers.

5. Gathering data on cryptocurrency prices

The core objective of this investigation revolves around attaining a heightened level of precision in prognosticating the price trajectories of BTC, along with an ensemble of six distinct cryptocurrencies — Ethereum, Litecoin, Tron, Polkadot, Cardano, and Stellar. This analytical endeavor delves into a year-long span, spanning from May 16, 2022, to May 17, 2023, with five-minute intervals.

The bedrock of this research is empirical foundation rests upon reputable data repositories, specifically, Yahoo Finance and Bitfinex. Renowned for their steadfast commitment to disseminating timely and accurate cryptocurrency market intelligence, these sources provide the fundamental bedrock for our data acquisition. However, a data cleansing and preprocessing method is instituted. This involves a multifaceted approach encompassing the rectification of anomalies, the imputation of missing data through alternative sources, the amalgamation of duplicate entries, and the meticulous curation of data outliers or irregularities. An integral facet of data preparation resides in the normalization procedure, which acts as a critical pivot to homogenize the disparate feature scales, thereby preventing any one feature from dominating the model's predictions (Singh & Singh, 2020).

In order to preprocess the data for utilization in ML models, we employed the subsequent normalization formula:

$$x_{norm} = \frac{x - (\alpha \times x_{min})}{(\beta \times x_{max}) - (\alpha \times x_{min})} \quad 0 < \alpha \leq 1 \quad , \quad 1 \leq \beta \tag{13}$$

where, x represents the raw data points, x_{min} is the minimum value of the data, and x_{max} is the maximum value of the data. The inclusion of α and β in the normalization process serves a crucial purpose. α allows for shifting the data by a fraction of its range, while β controls the scaling factor applied to the data. By adjusting these two constants the range and distribution of the normalized data can be tailored to meet the requirements of the ML models.

The rationale behind using α and β for normalization is to prevent any single feature from dominating the model's predictions due to variations in the data's magnitude (high-peak and low-peak oscillations). Moreover, the choice of α and β values facilitates handling extreme values and oscillations in the data, allowing the model to effectively learn from the information without being overly influenced by outliers or highly volatile periods.

Upon undergoing the normalization process, the dataset is effectively divided into three distinct categories: training (60%), validation (20%), and testing (20%). This strategic partitioning allows the model to acquire knowledge from a specific set of data during the training phase, all the while assessing its performance on previously unseen data through the validation and test sets. The training set serves as the bedrock for model training, while the validation set takes on a multifaceted role. It facilitates fine-tuning of hyperparameters, acts as a bulwark against overfitting, and gauges the model's predictive prowess using a custom-designed R2-score callback. Throughout each epoch, the model's predictive capacity is monitored using customized callback using validation data within the training section, and the optimal model weights are sorted. This pivotal phase furnishes a dependable yardstick for assessing the model's ability to generalize. Subsequently, the testing set serves to evaluate the model's performance on previously unseen data, thereby providing a reliable measure of the model's generalization capability.

Table 1 presents the correlation values within our dataset for each cryptocurrency. The analyses of charts and tables reveal a positive correlation between BTC and all other cryptocurrencies, except for Litecoin. Notably, Litecoin exhibits a negative correlation with most cryptocurrencies, with the exceptions being Ethereum and Tron. Table 2 provides informative statistics, such as the minimum, maximum, and mean values for each cryptocurrency across train, validation, and test subsets. These statistical insights offer valuable information about the distribution and characteristics of data for each cryptocurrency within the designated subsets.

6. Outcome assessment

6.1. Mean absolute percentage error

In the realm of data analysis and ML endeavors, the evaluation of forecasting models assumes a pivotal role. In this pursuit, a metric of evaluation often sought is the mean absolute percentage error (MAPE), a metric of considerable repute. It unfurls as a metric of accuracy assessment for forecasting models, offering an intuitive gauge of their performance. The essence of MAPE resides in its adeptness at quantifying the average percentage divergence between prognostications and actualities. Its computation entails taking the absolute difference between projected and observed values and subsequently normalizing it by the actual value. The outcome unfurls as a percentage error, an illuminating figure that facilitates the comparative analysis of precision amongst various predictive models. The formula that gives life to MAPE takes the following form:

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_{true,i} - y_{pred,i}}{y_{true,i}} \right| \times 100 \tag{14}$$

where, n is the number of data points, y_{true} represents the actual values, and y_{pred} denotes the predicted values. A lower MAPE value indicates better model performance.

6.2. Explained variance score

The explained variance score (EVS) stands as a pivotal metric in the evaluation of predictive models, bestowing a profound understanding of their performance. It quantifies the extent to which the model's predictions contribute to the overall variance observed in the data, shedding light on the model's adeptness in capturing the intrinsic structure of the dataset. A heightened EVS value serves as a harbinger of the model's enhanced capability to elucidate the fluctuations within the data. Beyond doubt, the EVS serves as an indispensable tool for the meticulous evaluation of predictive model performance. In essence, the formula for calculating the EVS is:

$$EVS = 1 - \frac{Var(y_{true} - y_{pred})}{Var(y_{true})} \tag{15}$$

where, Var denotes the variance, y_{true} represents the actual values, and y_{pred} denotes the predicted values. An EVS closer to 1 indicates better model performance.

6.3. R2-score

The R2-score, often known as the coefficient of determination, emerges as a significant metric in the appraisal of regression models. It serves to quantify the proportion of overall variance in the data that the model can elucidate, thereby offering a quantifiable gauge of how adeptly the model encapsulates the underlying structural patterns in the dataset. An R2-score of 1 signifies an impeccable alignment, symbolizing that the model impeccably accounts for all variations within the data. Conversely, an R2-score of 0 indicates an inability of the model to expound upon any of the data's variability, rendering it ineffectual for predictive purposes. Through a juxtaposition of the variance expounded by the model against the total variance, the R2-score delivers a robust assessment of the model's capacity to encompass the spectrum of data fluctuations. In practical terms, an R2-score is computed through the following formula:

$$R2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{16}$$

where, SS_{res} is the sum of squared residuals $\sum_{i=1}^{n}(y_{true,i} - y_{pred,i})^2$, and SS_{tot} is the total sum of squares $\sum_{i=1}^{n}(y_{true,i} - \bar{y}_{true})^2$, with $\bar{y}_{true}$ being the mean of the actual values.

6.4. Mean pairwise distance

An additional widely employed yardstick for gauging the effectiveness of prediction and regression models is the mean pairwise distance (MPD). This metric ascertains the average separation between pairs of data points found within the expected and observed values. A

Table 1
Correlation analysis of cryptocurrencies.

	BTC	ETH	TRX	DOT	ADA	LTC	XLM
BTC	**1.00000**	0.80849	0.66526	0.82841	0.77552	−0.00806	0.69686
ETH	0.80849	**1.00000**	0.43519	0.67836	0.59823	0.18218	0.45398
TRX	0.66526	0.43519	**1.00000**	0.58997	0.44592	0.18170	0.46413
DOT	0.82841	0.67836	0.58997	**1.00000**	0.85709	−0.20017	0.81922
ADA	0.77552	0.59823	0.44592	0.85709	**1.00000**	−0.36473	0.84129
LTC	−0.00806	0.18218	0.18170	−0.20017	−0.36473	**1.00000**	−0.52600
XLM	0.69686	0.45398	0.46413	0.81922	0.84129	−0.52600	**1.00000**

Table 2
Summary of descriptive data statistics.

Data type		BTC	ETH	ADA	LTC	TRX	DOT	XLM
Train	Min	15 580.0	889.4	0.23982	40.53	0.04540	4.429	0.07065
	Max	32 294.0	2121.6	0.68035	84.02	0.09173	11.504	0.15378
	Mean	20 925.0	1438.8	0.43597	60.03	0.06467	7.238	0.11045
Validation	Min	16 687.0	1239.8	0.24387	72.89	0.04977	4.233	0.07079
	Max	25 227.0	1739.8	0.41915	105.33	0.07297	7.816	0.09789
	Mean	22 100.6	1567.5	0.35664	90.60	0.06107	5.863	0.08662
Test	Min	19 622.0	1377.1	0.29812	65.50	0.05514	5.159	0.07461
	Max	30 954.0	2135.3	0.46094	103.10	0.07210	7.556	0.11350
	Mean	27 354.2	1815.8	0.37265	87.55	0.06637	6.141	0.09383

diminished MPD signifies heightened precision in the model's predictions, thereby hinting at superior performance. This is accomplished by evaluating the mean of the distances between all conceivable pairs of projected and authentic values. In essence, the MPD offers an assessment of the model's ability to apprehend and replicate the intricate patterns intrinsic to the dataset. Mathematically, the MPD can be ascertained through the following formula:

$$MPD = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \left(d(y_{true,i}, y_{true,j}) - d(y_{pred,i}, y_{pred,j}) \right) \tag{17}$$

where, n is the number of data points, $d(.,.)$ denotes the distance function (e.g., Euclidean distance), and the summations are taken over all pairs of distinct data points (i, j) with $i \neq j$. A lower MPD value indicates better model performance.

It is worth noting that several reputable studies, spanning contexts similar to ours (Deebadi, 2020; Gholipour, 2023; Mittal, Dhiman, Singh, & Prakash, 2019; Roy, Kumar, Singh, & Sangaiah, 2023; Singh, Kumar, & Akhtar, 2021; Sumi et al., 2023), have utilized these metrics, highlighting their versatile applicability and reliability in diverse research scenarios. After training the P-GRU model and obtaining predictions, it becomes imperative to assess the model's performance using pertinent metrics. The outcomes of these performance assessments are subsequently juxtaposed against those derived from the other forecasting models. This comparative analysis aims to unveil the P-GRU model's prowess in predictive capabilities, despite the inherent undulations and unpredictability innate to the data. In the face of this innate data diversity, a robust predictive model should demonstrate its adeptness in assimilating this inherent uncertainty, thereby engendering dependable prognostications.

7. Methodology

To imbue the GRU and LSTM models with probabilistic prowess, we seamlessly incorporate TensorFlow Probability into their architectural framework. This harmonious integration empowers us to tap into the probabilistic paradigm facilitated by TensorFlow Probability, thereby encapsulating uncertainties inherent in the models' predictions. These probabilistic models are armed with the negative log-likelihood as their loss function. In contrast, the other GRU and LSTM models resort to the mean squared error (MSE) for their loss function, quantifying the average squared divergence between the anticipated and true target values. Fig. 3 illustrates a simplified architecture of the proposed model. It employs a GRU layer specifically designed to process sequential data, enabling the model to capture long-term dependencies within the sequences. The information is then processed by fully-connected layers, transforming it for the intended task. Finally, a distribution layer outputs the model's prediction as a probability distribution.

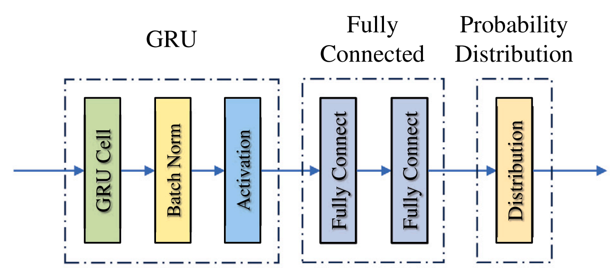


Fig. 3. Simplified P-GRU model architecture.

To further augment the prowess of our models, we introduce a customized callback mechanism during the training phase. This specialized callback undertakes a perpetual assessment with monitoring of the R2-score and MAPE over the validation data after each epoch. These metrics furnish invaluable insights into the model's ability to discern and encapsulate the inherent patterns and dynamics within the data, alongside its accuracy in making predictions. Moreover, this callback dynamically orchestrates adjustments to the learning rate, leveraging an exponential decay strategy. By adeptly modulating the learning rate, we meticulously fine-tune the model's training trajectory, fostering augmented convergence and fortifying its capacity for generalization. The astute oversight of the R2-score metric throughout this process ensures that only the optimal model weights are retained, forestalling overfitting and preventing the model from rote memorization of the training data.

Fig. 4 outlines the training process flowchart incorporating a bespoke callback mechanism. Data is initially split into distinct training, validation, and test sets (see Figs. 5 and 6). Following preprocessing, the training data undergoes iterative training. Within each epoch, the model is trained and its performance is evaluated using R2-score and MAPE on the validation data. A custom callback assesses if both the R2-score is higher and the MAPE is lower than the previously observed best. If these criteria are met, the model's weights are saved, signifying the current iteration as the best-performing model encountered so

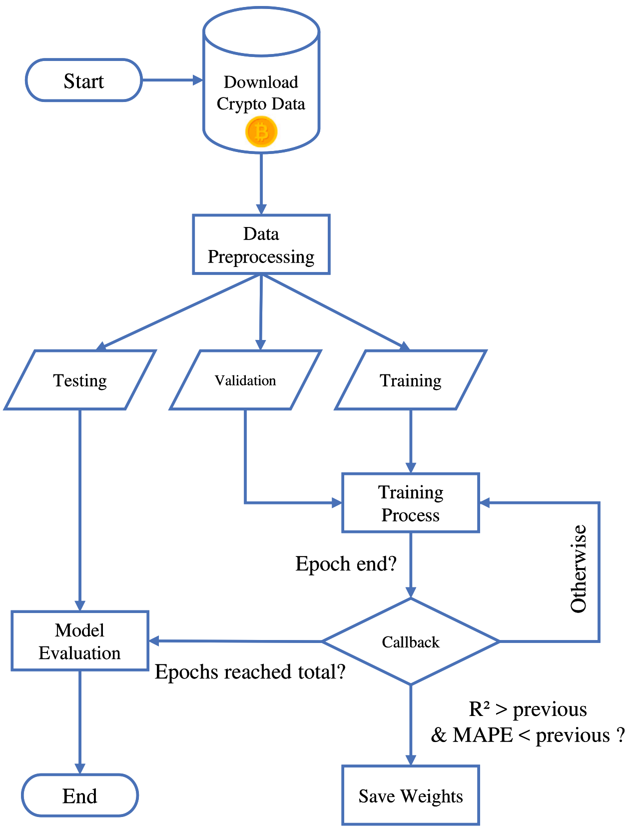


Fig. 4. Flowchart of the proposed training process.

far. This process continues until the designated number of epochs is reached, marking the completion of training.

Upon traversing a span of 100 epochs, we discern the most efficacious model based on its performance vis-à-vis the validation data. Subsequently, this chosen model undergoes a rigorous evaluation utilizing the test dataset. This exhaustive scrutiny serves to assess the model's capacity for transposing its acumen onto unseen data, while simultaneously furnishing a robust estimate of its predictive potential. In succession, the pre-trained model metamorphoses into a transfer model, tasked with the prediction projection of data from six additional cryptocurrencies. This transfer learning methodology capitalizes on the reservoir of knowledge amassed from the pre-trained model's exposure to BTC data, channeling it to prophesy the price dynamics of other cryptocurrencies. By transferring these acquired insights, the model attains an elevated level of performance across novel datasets, vividly showcasing the efficacy of transfer learning within the realm of cryptocurrency price prediction.

Overall, this approach not only facilitates the effective training and evaluation of multiple models but also elevates the predictive faculties of our models when confronted with supplementary datasets. The seamless assimilation of probabilistic capabilities and transfer learning stands as a seminal milestone within the sphere of cryptocurrency price prediction research. The ability to apprehend uncertainties and harness the wisdom garnered from pre-trained models augurs promising avenues for subsequent advancements, engendering more precise and dependable forecasts within the ever-evolving realm of cryptocurrencies.

The demo source code for implementing the proposed cryptocurrency price prediction is openly available on GitHub at https://github.com/amingolnari/Demo-BTCUSD-PricePred-ProbabilisticDL-TransferLearning. To facilitate accessibility and ease of use, readers are encouraged to run the code seamlessly through Google Colab.

8. Results

We embark on a comprehensive evaluation of the performance of four distinct types of recurrent neural networks. These networks are subjected to scrutiny both with and without the incorporation of, probability, bidirectional, and time-distribution techniques. The focal point of this assessment revolves around their predictive prowess in relation to the price trajectories of seven prominent cryptocurrencies. During the training phase, distinct loss functions are employed based on the model type. The P-GRU model is trained using the NLL as its loss function, while the simple GRU and LSTM models resort to the MSE for this purpose. In order to gauge the efficacy of these models, we employ two key performance metrics: R2-score and MAPE. These metrics, computed with respect to the validation dataset, stand as pivotal benchmarks for assessing the models' predictive prowess. The outcomes of these evaluations are elegantly presented in Table 3, encapsulating the essence of our analyses.

The analysis reveals a distinct trend, highlighting the favorable performance of the P-GRU model in relation to its counterparts. It garners noteworthy scores across various metrics, including a high R2-score of 0.99973, a minimal MAPE of 0.00190, an impressive EVS of 0.99968, and a modest MPD of 0.10156. This underscores the effectiveness of incorporating probabilistic attributes into the model's structure, with a notable emphasis on the role of a meticulously customized callback during the training phase to optimize model performance. In a broader context, these findings serve as evidence of the potential efficacy of RNNs in accurately predicting cryptocurrency prices. Specifically, the P-GRU model emerges as a standout performer. It is important to observe that the integration of probabilistic features into the GRU model brings about a notable enhancement in predictive capacity. Furthermore, the strategic implementation of a tailored callback during training adds a layer of optimization, ensuring precise cryptocurrency price predictions with commendable accuracy.

In summary, these findings not only underscore the potential for progress in the realm of cryptocurrency price prediction but also highlight the significant role played by RNNs, notably demonstrated by the P-GRU model. This approach suggests possibilities for further improvement, including the judicious application of transfer learning techniques to enhance model performance across the remaining six cryptocurrencies. For a perspective on the outcomes stemming from the application of the transfer learning technique to the P-GRU model for predicting the prices of other cryptocurrencies, please refer to Table 4.

The ensuing visualizations portray the outcomes of each employed model when applied to BTC data. A comprehensive perspective on the performance of different LSTM and GRU models can be gleaned from Figs. 7 and 8, respectively.

The residuals vs. predicted price plot, presented in the upcoming figure, serves as an illuminating window into the performance of each implemented model. This visual exploration revolves around dissecting the interplay between residuals (the differences between actual and predicted values) and anticipated prices. Through this scrutiny, we gain the ability to gauge the accuracy and efficacy of the models under examination. When the intervals between residuals and projected prices are minimized, it signifies an elevated level of precision and accuracy within the model's predictions.

This observation accentuates the significance of curtailing these residuals and endeavoring to achieve a cohesive clustering of data points around the projected price line. Models characterized by narrower intervals within this context emerge as more dependable and robust. Their ability to closely encapsulate the inherent trends and dynamics of the BTC data is underscored, bolstering their reliability (see Fig. 9).

The application of transfer learning to six prominent cryptocurrencies, namely Ethereum, Litecoin, Tron, Polkadot, Cardano, and Stellar, has yielded compelling outcomes, serving to accentuate the efficacy

Table 3
Evaluation metrics for BTC price prediction models.

Model	R2-score	MAPE	EVS	MPD
LSTM	0.93244	0.02163	0.98398	14.16147
Bidirectional LSTM	0.91187	0.02461	0.98086	18.41146
Bidirectional Probabilistic LSTM	0.99803	0.00343	0.99837	0.42596
Time Distributed LSTM	0.94552	0.02000	0.99010	11.42742
Bidirectional Time Distributed LSTM	0.90496	0.02562	0.97712	19.86073
Probabilistic LSTM	0.99843	**0.00145**	0.99847	0.12652
GRU	0.90810	0.02525	0.97813	19.20545
Bidirectional GRU	0.89530	0.02677	0.97807	21.82890
Bidirectional Probabilistic GRU	0.99898	0.00197	0.99952	0.17940
Time Distributed GRU	0.99092	0.00803	0.99360	2.06814
Bidirectional Time Distributed GRU	0.97560	0.01332	0.98915	5.25566
Probabilistic GRU	**0.99973**	0.00190	**0.99968**	**0.10156**

Table 4
Evaluation metrics for the P-GRU model's price prediction performance using transfer learning across six cryptocurrencies.

	ETH	TRX	DOT	ADA	LTC	XLM
R2-score	0.99895	0.99781	0.99847	0.99902	0.99765	0.99868
MAPE	0.00188	0.00220	0.00224	0.00203	0.00282	0.00252
EVS	0.99896	0.99762	0.99845	0.99903	0.99842	0.99905
MPD	0.01192	0.51227×10^{-6}	0.54469×10^{-4}	0.28499×10^{-5}	0.00118	0.10167×10^{-5}

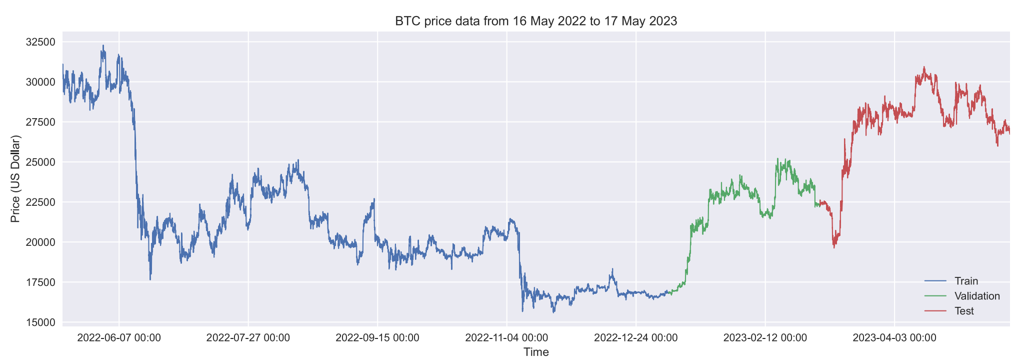
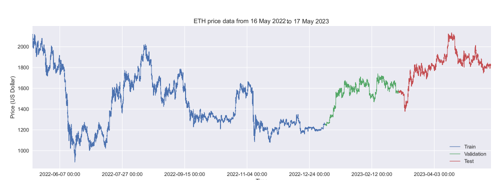
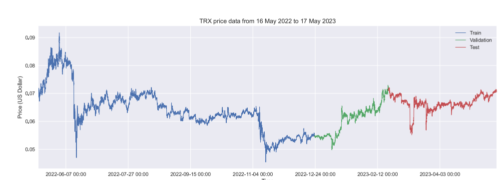


Fig. 5. Historical price of BTC from May 2022 to May 2023, segmented into training, validation, and test phases. The training segment is depicted in blue, validation in green, and test in red.


(a) Ethereum price history.


(b) Tron price history.

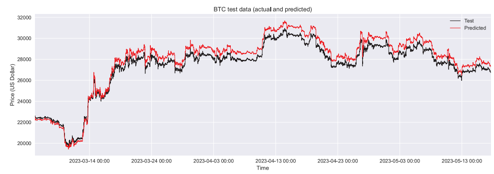

(c) Litecoin price history.


(d) Stellar price history.

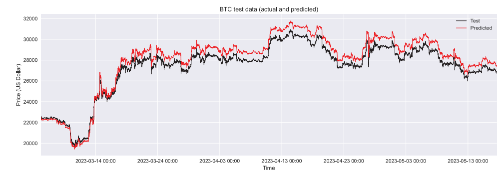

(e) Cardano price history.


(f) Polkadot price history.

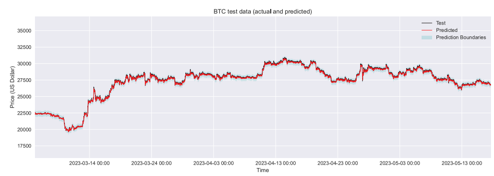
Fig. 6. Historical prices of cryptocurrency from May 2022 to May 2023, segmented into training, validation, and test phases. The training segment is depicted in blue, validation in green, and test in red.



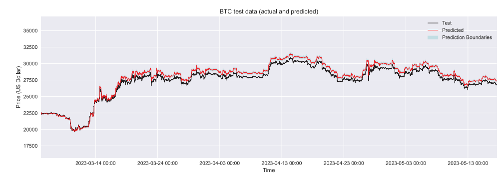
(a) Simple LSTM.



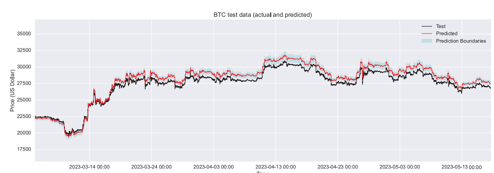
(b) Bidirectional LSTM.



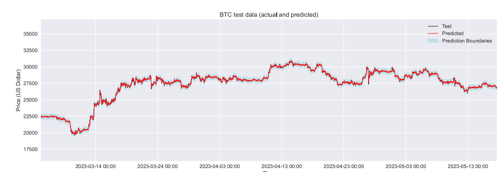
(c) Bidirectional probabilistic LSTM.



(d) Time-distributed LSTM.

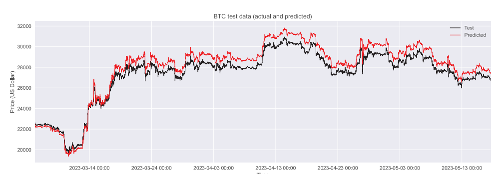


(e) Bidirectional time-distributed LSTM.

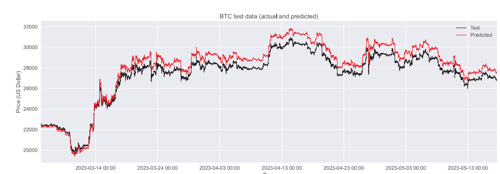


(f) Probabilistic LSTM.

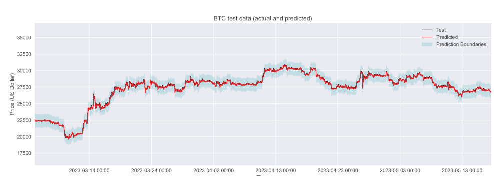
Fig. 7. BTC price and its prediction (LSTM), with the test dataset fluctuations in black and model-predicted values in red. Additionally, shaded blue boundaries (mean $\pm$ 2 $\times$ stdv) around the predicted values depict the distribution of expected prices, offering a depiction of the expected price range and uncertainty.



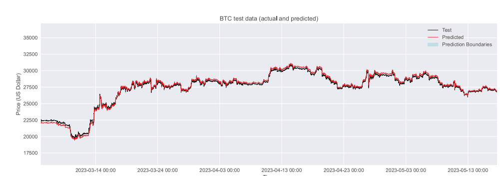
(a) Simple GRU.



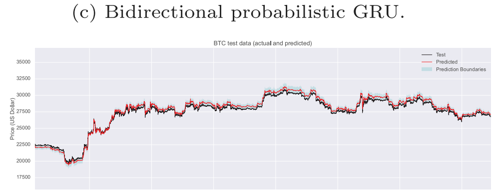
(b) Bidirectional GRU.



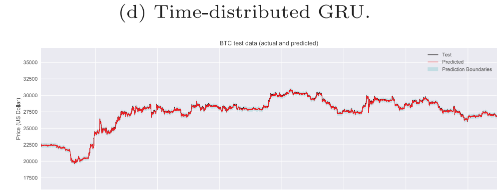
(c) Bidirectional probabilistic GRU.



(d) Time-distributed GRU.



(e) Bidirectional time-distributed GRU.



(f) Probabilistic GRU.

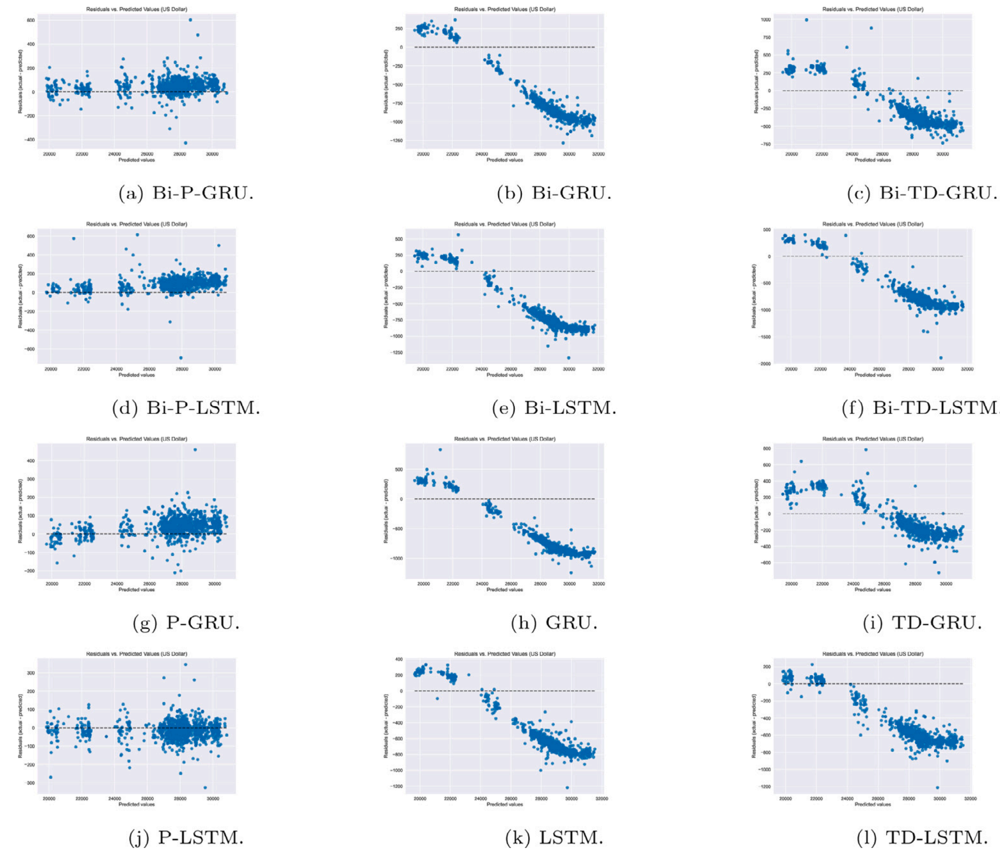
Fig. 8. BTC price and its prediction (GRU), with the test dataset fluctuations in black and model-predicted values in red. Additionally, shaded blue boundaries (mean $\pm$ 2 $\times$ stdv) around the predicted values depict the distribution of expected prices, offering a depiction of the expected price range and uncertainty.

of the proposed methodology. This approach capitalizes on the pre-trained BTC model to craft individual models for each cryptocurrency. This strategic utilization of BTC's knowledge repository forms the foundation of these new models.

Upon subjecting these models to rigorous evaluation against actual price data, the results are both intriguing and insightful. The visual representation of each cryptocurrency's price alignment with its projected values, as depicted in the following six plots, conveys a tangible sense of accuracy and trustworthiness inherent in the transfer learning strategy. This congruence between predicted and actual prices serves as a concrete validation of the resilience of the transfer learning framework (see Figs. 10 and 11). It further underscores the potential it holds for furnishing precise and consistent price predictions across an array of diverse cryptocurrencies.

9. Conclusion

The primary objective of this study was to introduce a DL model tailored specifically for accurately predicting the price of digital currencies, with a focus on BTC. The cryptocurrency market is known for



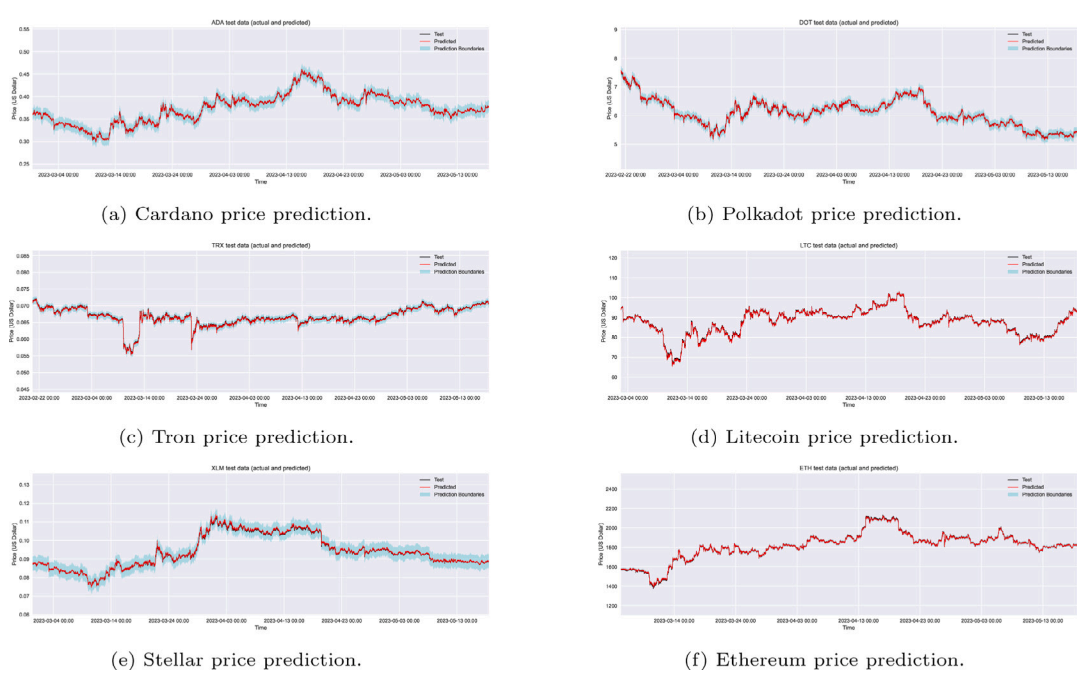
(a) Bi-P-GRU. (b) Bi-GRU. (c) Bi-TD-GRU.

(d) Bi-P-LSTM. (e) Bi-LSTM. (f) Bi-TD-LSTM.

(g) P-GRU. (h) GRU. (i) TD-GRU.

(j) P-LSTM. (k) LSTM. (l) TD-LSTM.

Fig. 9. Relationship between residuals and predicted prices on BTC dataset and different models, these plots provide a concise visual assessment of the model's accuracy and potential patterns in prediction errors.



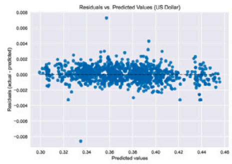
(a) Cardano price prediction. (b) Polkadot price prediction.

(c) Tron price prediction. (d) Litecoin price prediction.

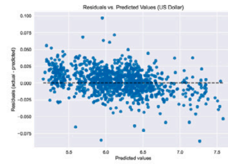
(e) Stellar price prediction. (f) Ethereum price prediction.

Fig. 10. Six cryptocurrencies price prediction using transfer learning (P-GRU), with the test dataset fluctuations in black and model-predicted values in red. Additionally, shaded blue boundaries (mean $\pm$ 2 $\times$ stdv) around the predicted values depict the distribution of expected prices, offering a depiction of the expected price range and uncertainty.

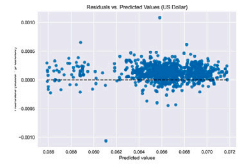
its uncertainty and volatility, making price forecasting a challenging task. To address this challenge, in this paper, we proposed a novel approach that incorporates P-GRU into the forecasting model, enabling the generation of probability distributions for predicted values. This



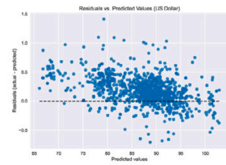
(a) Cardano P-GRU.



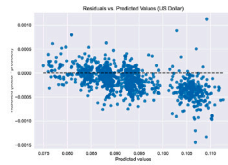
(b) Polkadot P-GRU.



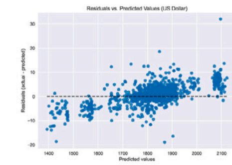
(c) Tron P-GRU.



(d) Litecoin P-GRU.



(e) Stellar P-GRU.



(f) Ethereum P-GRU.

Fig. 11. Relationship between residuals and predicted prices on six cryptocurrency datasets and the P-GRU model, these plots provide a concise visual assessment of the model's accuracy and potential patterns in prediction errors.

probabilistic feature provides valuable insights into the uncertainty associated with price predictions, offering a more robust and reliable forecasting framework.

The proposed P-GRU approach has been evaluated using one year of BTC price data collected at a five-minute timescale. The performance of the model has been compared with other well-established models such as GRU, LSTM, time-distributed, bidirectional, and simple models. The results demonstrated that the P-GRU approach outperformed traditional models, showcasing its superior accuracy in predicting cryptocurrency prices, especially in the context of uncertain market conditions.

To further optimize the model's performance, a customized callback was implemented using R2-score tracking. This callback effectively captured the optimal model weights based on validation data, preventing overfitting and enhancing the model's generalization capabilities. The utilization of this callback contributed to the robustness and reliability of the P-GRU model's predictions.

Furthermore, the scope of this research was expanded through a transfer learning approach. By leveraging a model pre-trained on BTC data, we successfully predicted the prices of six other prominent cryptocurrencies, namely Ethereum, Litecoin, Tron, Polkadot, Cardano, and Stellar. The application of transfer learning allowed for the transfer of knowledge from the BTC model to these additional cryptocurrencies, resulting in separate models for each of them. The findings demonstrated that the transfer learning approach significantly enhanced the model's performance on novel datasets, indicating the potential for broad applications in the cryptocurrency market.

In summary, the results of the study emphasize the effectiveness of the suggested P-GRU model in precisely predicting cryptocurrency prices, especially when confronted with market unpredictability and instability. By including probabilistic characteristics, the model offers a valuable indication of uncertainty in its forecasts, thereby enhancing its dependability. Additionally, the model's performance was further strengthened through the implementation of a customized callback and transfer learning, clearly showcasing its superiority over conventional models. All in all, the discoveries from this study make a substantial contribution to the progress of cryptocurrency price prediction models and hold promising prospects for future research and practical applications in the ever-evolving realm of cryptocurrencies.

CRediT authorship contribution statement

Amin Golnari: Software, Investigation, Methodology, Data curation, Visualization, Writing – Original Draft, Writing – review & editing. **Mohammad Hossein Komeili:** Investigation, Writing – review & editing. **Zahra Azizi:** Investigation, Writing – review & editing.

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.

Data availability

The data that has been used is confidential.

References

Ahmed, N. K., Atiya, A. F., Gayar, N. E., & El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, *29*(5–6), 594–621.

Alonso-Monsalve, S., Suárez-Cetrulo, A. L., Cervantes, A., & Quintana, D. (2020). Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators. *Expert Systems with Applications*, *149*, Article 113250.

Chang, T.-J., Lee, T.-S., Yang, C.-T., & Lu, C.-J. (2023). A ternary-frequency cryptocurrency price prediction scheme by ensemble of clustering and reconstructing intrinsic mode functions based on CEEMDAN. *Expert Systems with Applications*, *233*, Article 121008.

Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics*, *365*, Article 112395.

Chen, W., Xu, H., Jia, L., & Gao, Y. (2021). Machine learning model for Bitcoin exchange rate prediction using economic and technology determinants. *International Journal of Forecasting*, *37*(1), 28–43.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Deebadi, A. (2020). Understanding impact of Twitter feed on Bitcoin price and trading patterns.

Drahokoupil, J. (2022). Application of the XGBoost algorithm and Bayesian optimization for the Bitcoin price prediction during the COVID-19 period. FFA Working Papers 4.

Geweke, J., & Amisano, G. (2010). Comparing and evaluating Bayesian predictive distributions of asset returns. *International Journal of Forecasting*, *26*(2), 216–230.

Gholipour, M. (2023). Leveraging the power of hybrid models: Combining ARIMA and LSTM for accurate Bitcoin price forecasting.

Gradojevic, N., Kukolj, D., Adcock, R., & Djakovic, V. (2023). Forecasting Bitcoin with technical analysis: A not-so-random forest? *International Journal of Forecasting*, *39*, 1–17.

Hamayel, M., & Owda, A. (2021). A novel cryptocurrency price prediction model using GRU, LSTM and bi-LSTM machine learning algorithms. *AI*, *2*, 477–496.

Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, *37*(1), 388–427.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780.

Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, *32*(10), 2513–2522.

Jospin, L. V., Laga, H., Boussaid, F., Buntine, W., & Bennamoun, M. (2022). Hands-on Bayesian neural networks—A tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, *17*(2), 29–48.

Kim, Y. B., Kim, J., Kim, W., Im, J., Kim, T., Kang, S., et al. (2016). Predicting fluctuations in cryptocurrency transactions based on user comments and replies. *PLoS One*, *11*, Article e0161197.

Kuan, C.-M., & White, H. (1994). Artificial neural networks: An econometric perspective. *Econometric Reviews*, *13*(1), 1–91.

Liu, M., Li, G., Li, J., Zhu, X., & Yao, Y. (2021). Forecasting the price of Bitcoin using deep learning. *Finance Research Letters*, *40*, Article 101755.

Mallqui, D. C., & Fernandes, R. A. (2019). Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques. *Applied Soft Computing*, *75*, 596–606.

McNally, S., Roche, J., & Caton, S. (2018). Predicting the price of Bitcoin using machine learning. In *2018 26th Euromicro international conference on parallel, distributed and network-based processing* (pp. 339–343).

McNelis, P. D. (2005). *Neural networks in finance: gaining predictive edge in the market*. Academic Press.

Meng, X., Babaee, H., & Karniadakis, G. E. (2021). Multi-fidelity Bayesian neural networks: Algorithms and applications. *Journal of Computational Physics*, *438*, Article 110361.

Mittal, A., Dhiman, V., Singh, A., & Prakash, C. (2019). Short-term Bitcoin price fluctuation prediction using social media and web search data. In *2019 twelfth international conference on contemporary computing* (pp. 1–6). IEEE.

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

Oyedele, A. A., Ajayi, A. O., Oyedele, L. O., Bello, S. A., & Jimoh, K. O. (2023). Performance evaluation of deep learning and boosted trees for cryptocurrency closing price prediction. *Expert Systems with Applications*, *213*, Article 119233.

Patel, M. M., Tanwar, S., Gupta, R., & Kumar, N. (2020). A deep learning-based cryptocurrency price prediction scheme for financial institutions. *Journal of Information Security and Applications*, *55*, Article 102583.

Pour, E. S., Jafari, H., Lashgari, A., Rabiee, E., & Ahmadisharaf, A. (2022). Cryptocurrency price prediction with neural networks of LSTM and Bayesian optimization. *European Journal of Business and Management Research*, *7*(2), 20–27.

Rajabi, S., Roozkhosh, P., & Farimani, N. M. (2022). MLP-based learnable window size for Bitcoin price prediction. *Applied Soft Computing*, *129*, Article 109584.

Ranjan, S., Kayal, P., & Saraf, M. (2023). Bitcoin price prediction: A machine learning sample dimension approach. *Computational Economics*, *61*, 1617–1636.

Roy, P. K., Kumar, A., Singh, A., & Sangaiah, A. K. (2023). Forecasting Bitcoin prices using deep learning for consumer centric industrial applications. *IEEE Transactions on Consumer Electronics*.

Singh, A., Kumar, A., & Akhtar, Z. (2021). Bitcoin price prediction: A deep learning approach. In *2021 8th international conference on signal processing and integrated networks* (pp. 1053–1058). IEEE.

Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, Article 105524.

Sujatha, R., Mareeswari, V., Chatterjee, J. M., Abd Allah, A. M., & Hassanien, A. E. (2021). A Bayesian regularized neural network for analyzing Bitcoin trends. *IEEE Access*, *9*, 37989–38000.

Sumi, U. S., Akter, R., Ahamed, K. A., Sutradhar, S., Ahamed, S., & Elias, T. (2023). Analysis of machine learning and deep learning to forecast prices on several crypto exchanges. In *2023 fifth international conference on electrical, computer and communication technologies* (pp. 1–8). IEEE.

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2020). Using deep learning for price prediction by exploiting stationary limit order book features. *Applied Soft Computing*, *93*, Article 106401.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, *3*(1), 1–40.

Wen, B., Wang, Y., Ding, Y., Zheng, H., Qin, B., & Yang, C. (2023). Security and privacy protection technologies in securing blockchain applications. *Information Sciences*, Article 119322.