Applied Soft Computing Journal 186 (2026) 114043

Contents lists available at ScienceDirect

Applied Soft Computing

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

# Quantile deep learning models for multi-step ahead time series prediction

Jimmy Cheung, Smruthi Rangarajan, Amelia Maddocks, Rohitash Chandra*

*Transitional Artificial Intelligence Research Group, School of Mathematics and Statistics, University of New South Wales, Sydney, Australia*

## HIGHLIGHTS

- We present a novel quantile regression deep learning framework for multi-step time series prediction.
- We elevate the capabilities of deep learning models by incorporating quantile regression.
- We include multivariate and univariate modelling strategies and compare with conventional deep learning models.
- The model provide additional predictions without a loss in the prediction accuracy when compared to the literature.

## ARTICLE INFO

*Keywords:*
Deep learning
Time series prediction
Quantile regression
Multivariate modelling
Multi-step ahead prediction

## ABSTRACT

Uncertainty quantification is crucial in time series prediction, and quantile regression offers a valuable mechanism for uncertainty quantification, which is useful for extreme value forecasting. Although deep learning models have been prominent in multi-step ahead prediction, the development and evaluation of quantile deep learning models have been limited. We present a novel quantile regression deep learning framework for multi-step time series prediction. In this way, we elevate the capabilities of deep learning models by incorporating quantile regression, thus providing a more nuanced understanding of predictive values. We provide an implementation of prominent deep learning models for multi-step ahead time series prediction and evaluate their performance under high volatility and extreme conditions. We include multivariate and univariate modelling, strategies, and provide a comparison with conventional deep learning models from the literature. Our models are tested on two cryptocurrencies: Bitcoin and Ethereum, using daily close-price data and selected benchmark time series datasets. The results show that integrating a quantile loss function with deep learning provides additional predictions for selected quantiles without a loss in prediction accuracy. Our quantile model can handle volatility more effectively and provides uncertainty quantification through the use of quantiles when compared to conventional deep learning models.

## 1. Introduction

In time series forecasting, uncertainty quantification is a critical component for informed decision-making, particularly in high volatility problems such as financial markets, energy demand, and weather forecasting. Although conventional deep learning models are powerful in multi-step-ahead forecasting, they often fall short in providing comprehensive measures of uncertainty. This gap can be addressed by integrating quantile regression, a statistical technique that offers a mechanism for extreme forecasting by predicting the conditional quantiles of a response variable. Koenker and Bassett [1] introduced the quantile regression model in the mid-1970s to estimate conditional quantiles, offering a measure of uncertainty rather than single-point predictions as in conventional linear regression models. Quantile regression has been widely used in statistical analysis [2] and finds applications in various fields, including epidemiology [3], economics [4,5], ecology [6], and finance [7]. In the field of economics, it has been employed to study salary distributions influenced by returns to education and student experience [8]. In medicine, quantile regression has been used to analyse the effects of different local anesthetics on the duration of nerve blocks [9]. Unlike traditional linear models such as least squares regression [10,11], quantile regression provides more comprehensive information about the conditional distribution, revealing data characteristics across different quantiles as well as the average of the data. Hence, this approach offers a methodology for projecting uncertainties in prediction [12,13].

* Corresponding author.
*Email address:* rohitash.chandra@unsw.edu.au (R. Chandra).

https://doi.org/10.1016/j.asoc.2025.114043
Received 18 February 2025; Received in revised form 4 September 2025; Accepted 2 October 2025
Available online 15 October 2025

Extreme value prediction [14] focuses on forecasting rare and significant events, which are often outliers or extreme values in a dataset and have a low probability of occurrence but can cause major consequences [15]. In a meteorological context, an example is the rapid intensification of cyclones [16]. Given the distribution of a dataset, this statistical modelling approach targets the tail of the distribution where extreme events reside, allowing for the estimation of their probability. It finds applications in various fields, including natural disasters [17–19], financial crises [20], and system failures [21]. Extreme value forecasting (prediction) is crucial for enhancing risk management as it provides a foundation for developing emergency plans [22] and preventive measures [23]. For instance, it can be used to assess potential casualties in earthquake disasters [19], helping to minimise losses. Additionally, whilst extreme value prediction targets the rare extreme events that are in the tails of the distribution, quantile regression provides a more generalised approach to estimate various quantiles, such as the median and 90th percentile of the response variable's conditional distribution [24]. However, classic quantile regression can perform poorly for extreme values. Quantile regression, when integrated with extreme value theory (EVT) [25,26], can estimate conditional quantiles that extend beyond the observed data range [27]. Cai et al. [24] developed extreme value prediction using a quantile function model.

Deep learning models can handle complex, high-dimensional data and extract hidden patterns and features, making them particularly effective for time series prediction, especially with nonlinear and multivariate data [28]. These models have been extensively used for time series forecasting, including univariate, multivariate, single-step, and multi-step predictions [29]. In the meteorological field, deep learning can be used to predict extreme weather events [30] such as smog [31], heavy rainfall [32], and declining groundwater levels [33]. By analysing historical meteorological data and satellite images, deep learning models can also identify early signals of extreme weather, enabling advanced preparation to mitigate potential damage [30].

The combination of quantile regression with deep learning [34,35] has been applied to survival analysis problems such as breast cancer, utilising the Huber check function and inverse probability censoring weights (IPCW) function [36]. Recent studies have utilised the quantile regression forests (QRF) model to predict road traffic volume, showing significant implications for regional development [37]. Furthermore, integrating quantile regression with deep learning models, such as the long short-term memory (LSTM) network has significantly improved the accuracy and reliability of river runoff predictions [38]. Hu et al. [39] presented the monotone quantile regression neural network (MQRNN) to address the quantile crossing problem in time series prediction by considering the monotonicity of quantiles. An improved quantile regression neural network (iQRNN) [40] was used for probabilistic load forecasting that utilised strategies such as batch training, early stopping, and dropout regularisation to improve the training efficiency and prediction stability of the model. Recent advancements, including MQRNN [41] and the deep partially linear quantile regression neural network (DPLQR) model [42] have addressed quantile crossover, where different quantile estimation lines (e.g. 10 % quantile, 50 % quantile) may cross or stagger during the prediction process in the quantile regression. These models highlight the potential of combining deep learning with quantile regression for enhanced uncertainty quantification.

The integration of deep learning and extreme value prediction has demonstrated significant application potential across various fields. Deep learning has been leveraged to predict extreme market fluctuations, aiding investors in anticipating the risks of financial crises or market crashes [43]. The combination of extreme value theory (EVT) with neural networks has significantly improved the accuracy of predicting extreme events in financial markets [44]. A hybrid model framework that combines EVT and machine learning [45] can more accurately estimate stock market risks by processing multivariate and high-frequency data, thereby enhancing risk management and investment decision-making accuracy. Furthermore, there is limited work in the area of quantile regression for multi-step ahead forecasting.

In this study, we present a novel quantile regression deep learning framework for multi-step time series prediction. In this way, we elevate the capabilities of deep learning models by incorporating quantile regression, thus providing a more nuanced understanding of predictive values. We evaluate the framework using univariate and multivariate benchmark datasets and focus on multi-step ahead time series prediction. The datasets exhibit conditions of high volatility and extremes, including cryptocurrency and benchmark time series. We evaluate the framework with selected deep learning models, which have been demonstrated to be very promising for multi-step ahead forecasting [46]. We provide open-source Python code and data so that our framework can be extended and applied to various fields that feature extreme values and require uncertainty quantification.

The rest of the paper is organised as follows. In Section 2, we provide background and related work, and in Section 3, we present the methodology. Section 4 presents the results, and Sections 5 and 6 provide the discussion and conclusions, respectively.

## 2. Background and related work

### 2.1. Quantile regression

The quantile regression model is an extension of linear regression that estimates the conditional median (other quantiles) of the response variable using the conditional quantile function. For the $\tau$-th quantile ($0 < \tau < 1$), the quantile model is:

$$Q_y(\tau|X) = X\beta(\tau) \tag{1}$$

where: $Q_y(\tau|X)$ represents the $\tau$-th quantile of the dependent variable $y$ given the independent variables $X$. $X$ is the vector of independent variables, containing $n$ observations. $\beta(\tau)$ is the vector of coefficients associated with the $\tau$-th quantile. Quantile regression estimates the conditional distribution of the dependent variables under the different quantiles. It is different from ordinary least squares regression (OLS), which is based on the conditional mean of the estimated dependent variable. Therefore, quantile regression can provide a more comprehensive understanding of the data and offer a means of uncertainty quantification in predictions. Through quantile regression, we can obtain a more detailed description of the entire distribution of a given dataset by estimating different quantiles (such as the 10th, 50th, and 90th quantiles).

Quantile regression is less sensitive to outliers because its estimation is based on minimising the absolute error with a specialised loss function, rather than the conventional squared error loss in linear models. The quantile loss function is given as follows:

$$\rho_\tau(u) = u(\tau - 1_{u<0)}) \tag{2}$$

where: $u = y - X\beta(\tau)$ represents the residuals. $\rho_\tau(u)$ is the asymmetric absolute loss function, also known as the "check loss function." $I(\cdot)$ is the indicator function, which takes the value 1 if the condition inside is true, otherwise 0.

A simple linear regression example can be used to visualise the concept of quantiles, where regression lines for different quantiles are displayed on the same graph as shown in Fig. 1.

Quantile regression is also known to be applicable for data that is heteroskedastic, i.e., the variance of the residuals in the regression model is not constant across all data points. Moreover, quantile regression can capture the potential nonlinear and heterogeneous effects of explanatory variables on the dependent variable, allowing for enhanced insights on the given dataset [47,48].

The application of quantile regression in time series analysis has garnered widespread attention [49] by expanding modelling options through allowing for modelling of local and quantile-specific dynamics. For example, the weighted Nadaraya-Watson (WNW) regression has

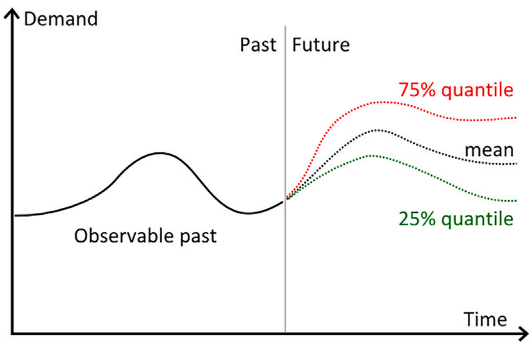



**Fig. 1.** The uncertainty range provided in the demand forecast using quantile regression includes the forecast mean and the trend of the upper and lower quantiles (25 % and 75 %).

been effectively used to implement quantile regression in time series data [49]. Additionally, quantile regression can be applied to interval forecasting, structural change detection, and portfolio construction [50]. Koenker [51] provided a comprehensive review of the development and applications of quantile regression over the past forty years, which was further extended by Tyralis et al. [35].

### 2.2. Extreme value theory

EVT [25,26] forms the basis of extreme value prediction, which focuses on modelling the probability distribution of the tail of the data. It provides a set of methods and distribution models for modelling and analysing extreme events. The generalised extreme value distribution (GEV) [52] and the generalised Pareto distribution (GPD) [53] are two important tools to define and estimate models in EVT. The block maxima (BM) [54] method and the peak over threshold (POT) [25] method are two common model parameter estimation methods for extreme value analysis.

The application of the two main theorems of extreme value theory includes (i) the main limit theorem of EVT (proved by Fréchet [55] in 1927 for the Pareto-type limit distribution and by Fisher and Tippet [56] in 1928 for the Weibull and Gumbel limit distributions) leading to the GEV distribution, and (ii) the Gnedenko-Pickands-Balkema-de Haan theorem [57] leading to the GPD distribution. In the case when the extreme value distribution of a random variable meets certain conditions, exceedance over a high threshold can be approximated by the GPD, while the GEV distribution can approximate the block maxima. The GEV distribution is used in the BM method to analyse extremes [54], while the GPD distribution is used in the POT method to analyse threshold exceedances [25]. POT is more suitable for quantile estimation as it can better utilise extreme observations with a larger sample size [58], whilst BM is more suitable for estimating the return level, which refers to the threshold value that is expected to be exceeded once within a particular time period.

There are various methods for extreme value prediction, including parametric [59], non-parametric [60], and semi-parametric approaches [61]. Parametric methods assume that the data follow a specified extreme value distribution and make predictions by estimating parameters. Non-parametric methods use the data directly for prediction and do not rely on a specific distribution structure. Semi-parametric methods combine the advantages of both approaches, utilising data characteristics whilst also considering specific distributions.

### 2.3. Multi-step time series prediction

Single-step prediction refers to a model prediction one step ahead in time, while multi-step time series is a task that aims to predict multiple time steps into the future [62]. This becomes increasingly complex with the number of forecast steps, particularly with time series data. The strengths and weaknesses of different neural network architectures vary significantly for time series prediction [46]. Since time series prediction depends on temporal patterns, it's essential to carefully select the optimal neural network architecture and training method. The prediction errors can accumulate over time, especially when dealing with chaotic time series datasets. This implies that to produce precise results, the predictive capabilities and hyperparameters of different models need to be considered and customised to the given dataset. Chandra et al. [46] evaluated a variety of deep learning models, including simple recurrent neural networks (RNN), long short-term memory networks (LSTM), bidirectional LSTM networks (BD-LSTM), encoder-decoder LSTM networks (ED-LSTM), and convolutional neural networks (CNN). These models have then been compared on their performance on univariate time series datasets. They reported that bidirectional LSTM and encoder-decoder LSTM networks performed the best in terms of their prediction accuracy, which highlights the advantages of utilising deep learning models in handling multi-step ahead time series prediction.

Furthermore, Chang et al. [63] used real-time recurrent learning for training RNNs for flood forecasting, using an iterative approach for two-step-ahead forecasts. Additionally, Khedkar et al. [64] incorporated EVT into deep learning models to address extreme flooding issues across Australia's major catchments. They utilised multivariate and multi-step time series prediction and reported that quantile-LSTM outperformed the baseline deep learning models while providing uncertainty estimates in hydrological forecasting.

In recent years, deep learning models have shown considerable potential in predicting cryptocurrency prices, which is an area characterised by high volatility and unpredictability. Bayesian neural networks (BNNs) [65] have been used to deal with volatility and uncertainty quantification in predictions by treating model parameters as probability distributions rather than fixed values [66]. In cryptocurrency price prediction, BNNs provide a way to quantify the uncertainty of predictions [67], which is particularly important in financial applications that require risk assessment. Chandra and He [68] employed BNNs to investigate the performance of related multi-step-ahead forecasting models for stock prices, during the COVID-19 pandemic and reported that accurate forecasting was challenging due to the high volatility of the stock market. These areas are precisely where BNNs may perform well in volatile markets by providing more reliable uncertainty estimates.

In addition to the use of BNNs, Wang et al. [69] applied machine learning techniques to forecast cryptocurrency volatility utilising intrinsic features (internal and external determinants). Their findings revealed that LSTM networks significantly outperformed traditional volatility models such as Generalised Autoregressive Conditional Heteroskedasticity (GARCH). Wu et al. [70] evaluated selected deep learning models, including CNNs, Transformer models, and LSTM variants, using datasets from both before and during the COVID-19 pandemic for cryptocurrency price prediction. They emphasised the importance of evaluating models in different scenarios, and identified the convolutional LSTM with a multivariate approach as the most accurate model.

## 3. Methodology

### 3.1. Deep learning models

Recurrent Neural Networks (RNNs) feature recurrent connections in the hidden layer to represent temporal data [71], and are designed to handle sequential data [72] and thus are extensively utilised for time series forecasting [46]. LSTM network [73] is a variant of an RNN that addresses the problem of learning long-term dependencies by conventional RNNs such as the Elman RNN [71]. LSTMs are particularly effective for handling temporal data, since they can retain information

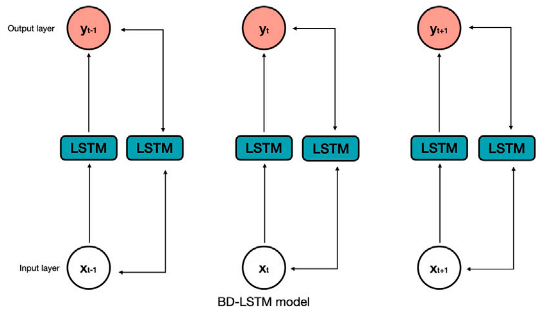



**Fig. 2.** Bidirectional-LSTM network showing the flow of information.

over longer periods, outperforming conventional RNNs. LSTM models enhance traditional RNNs by incorporating memory cells that feature multiple gates to manage information flow. There are 4 components in an LSTM memory cell (unit): the input gate, the forget gate, the output gate and the cell state. The interaction of these gates is the crucial part in updating the cell state, which aids in combating the issues related to the vanishing and exploding gradients [73] faced by conventional RNNs [72].

The bidirectional long short-term memory (BD-LSTM) is an advanced LSTM model that handles information in both forward and backward directions through two independent hidden layers, as shown in Fig. 2. Unlike canonical LSTM models that process information in a single direction, each input sequence is passed through the RNN twice; once in the forward direction and once in reverse [74]. This made them prominent for language modelling and natural language processing (NLP) tasks [75], and also for multi-step ahead time series forecasting [46,70]. The encoder-decoder long short-term memory (ED-LSTM) model was designed to handle language modelling tasks [76], which is also effective for time series prediction due to its ability to capture complex temporal patterns and dependencies over long sequences. Although the convolutional LSTM (Conv-LSTM) network was initially used for weather forecasting problems [77], it is also capable of handling a wide range of time series-related data. Conv-LSTM can effectively harness both spatial and temporal dependencies in data by combining the strengths of CNNs [78] and LSTM networks. This capability makes Conv-LSTM particularly suited for tasks involving multivariate time series forecasting, such as predicting cryptocurrency prices. Therefore, we used these models for our quantile deep learning framework.

### 3.2. Quantile deep learning model

The key and unique feature of these model implementations is the use of the quantile loss function. For each defined model, there will be a 'classic' version with a standard loss function and another version utilising the quantile loss function. This approach allows us to evaluate which set of models performs better, offering more comprehensive predictions that account for the inherent volatility in cryptocurrency markets.

The quantile loss function helps in making predictions that are more tailored to specific sections of the dataset. Instead of predicting only the average outcome, it allows for the prediction of a set of defined quantiles. Although the quantile loss function does not predict an exact value, we assume the median values for each time step (prediction horizon) as the main predicted value.

$$\ell_q(y, \hat{y}) = \begin{cases} q \cdot (y - \hat{y}) & \text{if } y \geq \hat{y} \\ (q-1) \cdot (y - \hat{y}) & \text{if } y < \hat{y} \end{cases} \tag{3}$$

where, $y$ is the true value, $\hat{y}$ is the predicted value and $q$ is the specific quantile (e.g. $q = 0.95$). We can interpret that if $y \geq \hat{y}$, the actual value is greater than or equal to the predicted value. The loss is given as $q$ times the difference between the true and predicted values. Therefore, for higher quantiles, the error is higher when the prediction is less than the actual value. In the case that $y < \hat{y}$, the actual value is less than the predicted value. In this case, the loss is $(q-1)$ times the difference between the actual and predicted values.

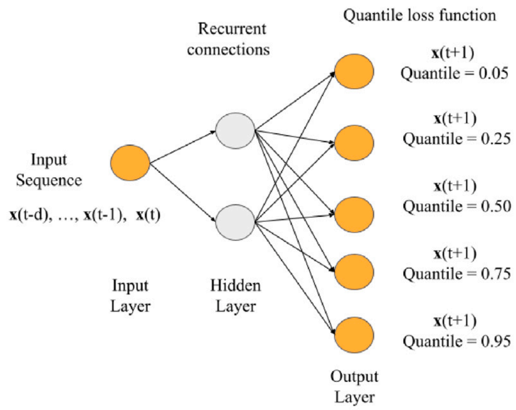
Applying the quantile loss function to time series data allows for a broader range of predicted values and enables an overview of uncertainties. Instead of predicting a single close price, our implementation will use the quantile loss function that considers a set of quantiles of a prediction horizon (step). Fig. 3 presents a quantile RNN for one-step ahead and multi-step-ahead prediction using two strategies, i.) grouped percentiles (Panel b) and ii.) vector-based quantiles (Panel c). $x$ represents the time series data indexed by time $t$ that is windowed by size $d$ for $m$ step-ahead prediction. Note that the vector-based quantiles have further connections to the hidden neurons, which are not explicitly shown. Furthermore, the time-based input and recurrent connections are also not explicitly shown in the RNN. Fig. 3 highlights the interaction between the quantile loss function at the output layer of a simple RNN, which is also applicable to other deep learning models (CNN and LSTM models). We use the quantile loss function instead of the mean squared error loss for the output layer. We are interested in capturing the uncertainty in predictions at the 5th, 25th, 50th, 75th and 95th percentiles, hence we use the quantile values of 0.05, 0.25, 0.5, 0.75 and 0.95. Note that other quantile values can be defined, as long as they fall within the given range (0 to 1). After we have defined our quantile values, we present the data sample to the input layer of the deep learning model, which will be propagated through the hidden layers, and finally to the output layer (neurons). Depending on the quantile value $\tau$, we assign a weight to the quantile loss function, and the further $\tau$ deviates from 0.5, the more biased the loss function, producing a lower ($\tau < 0.5$) or upper ($\tau > 0.5$) value than the median prediction. The number of quantile values determines the length of each output neuron. In Fig. 3, each output neuron features the predicted values from all the defined quantiles. In the case of the multivariate features, additional neurons in the input layer can be added for each feature based on the architecture given in Fig. 3.

### 3.3. Framework

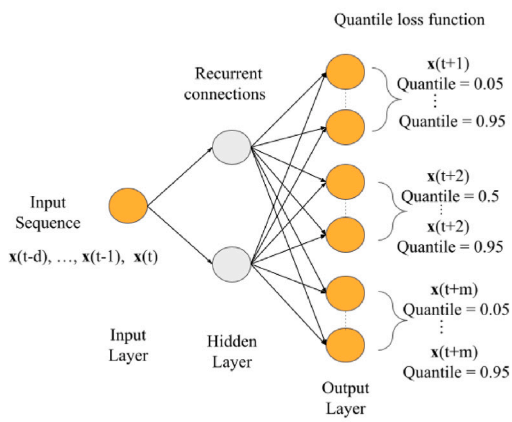
We outline the key stages in the framework (Fig. 4) that include data processing and prediction using deep learning models. In Stage 1, we begin by extracting and processing the selected datasets and applying exploratory data analysis. We need to transform the original time series data into sequences that can be used for prediction.

Stage 2 involves preparing the data for model training. In the case of deep learning models, we need to process the data depending on their nature, i.e. univariate and multivariate data for associated models, as shown in our framework. This sliding window technique ensures that the model learns from a variety of overlapping sequences, capturing the temporal dependencies in the data. These sequences are then normalised and split into training and testing datasets, as done in previous works from the related literature [70]. The univariate time series is divided into overlapping windows, each window contains an input sequence vector of a fixed number of consecutive time points (size $d = 6$) and an output sequence vector (size $m = 5$) for the future predicted time points. In the case of the multivariate strategy, the model input features include (*high, low, open, close price* and *volume*) to predict the close price for five days (steps). The input features are crucial factors that affect the future closing price of the given cryptocurrency, and the previous high and low prices also support estimating the quantiles. In the case of univariate models, we selected the close price, as this was determined to be the most important feature in earlier work [70].

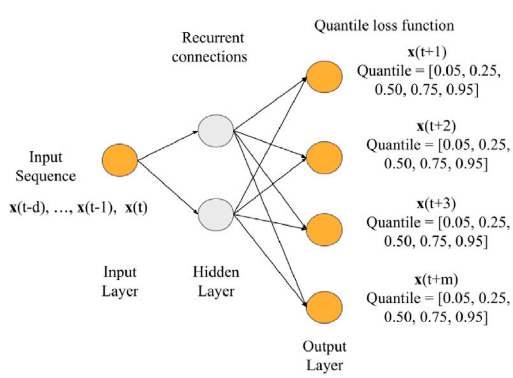
In Stage 3, we reviewed the literature to find the most appropriate deep learning models. We selected BD-LSTM, Conv-LSTM, and ED-LSTM and defined their hyperparameters from the literature [70]. We defined




(a) One-step ahead prediction.




(b) Multi-step ahead prediction using grouped quantiles.




(c) Multi-step ahead prediction using vector-based quantiles for each prediction horizon.

**Fig. 3.** Quantile recurrent neural network for one-step ahead and multi-step ahead prediction using two strategies, i.) grouped percentiles (Panel b) and ii.) vector-based quantiles (Panel c). Note that the vector-based quantiles have further connections to the hidden neurons, which are not explicitly shown. In the case of the multivariate features, additional neurons in the input layer can be added for each feature. The time-based input and recurrent connections are also not explicitly used in the recurrent neural network. $x$ represents the time series data index by time $t$ that is windowed by size $d$ for $m$ step-ahead prediction.

the deep learning model architectures, such as the input size and output size, as shown in Fig. 3.

In Stage 4, using the three models, we developed a quantile loss function as shown in Fig. 3. The most complex part of our framework is setting up and training the multivariate multi-step ahead quantile-based deep learning models. In both cases, the multi-step ahead predictions are handled by defining multiple output neurons, with each output neuron representing a distinct step-ahead prediction, along with each output neuron presenting a quantile as shown in Fig. 3-Panel (b) employing the quantile loss function. We create both a standard and a quantile deep learning model and use the Adam optimiser for training them.

In Stage 5, we provide an analysis of the predictions and review the strengths and weaknesses of the respective models and training strategies. We can facilitate a comprehensive comparison of model performance using different metrics, including root-mean-square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). In our evaluation, we specifically use the RMSE as given below.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4}$$

where, $n$ is the number of data points (samples), $y$ and $\hat{y}$ are the actual and predicted values, respectively. In the case of multi-step-ahead prediction, we take the mean of the RMSE of the $m$ steps. We report RMSE of the different quantiles, where the $\hat{y}$ is of a specific quantile (e.g. quantile value of 0.95).

### 3.4. Data

We demonstrate the effectiveness of quantile deep learning models using multivariate and univariate time series datasets. We feature cryptocurrency stock price data from previous studies [70] (Bitcoin and Ethereum) and benchmark time series, including Sunspot, Mackey-Glass and Lorenz time series [46] (Fig. 5).

1. Bitcoin is a multivariate dataset that contains daily entries of Bitcoin prices - high, low, open and close prices - along with trade volume and market capitalisation. There are 2991 data points, dating from April 2013 to July 2021.
2. Ethereum is a multivariate dataset that contains daily entries of Ethereum prices - high, low, open and close price - along with trade volume and market capitalisation. There are 2160 data points dating from August 2015 to July 2021. Both cryptocurrency datasets contain columns such as serial number, name, symbol and date, but we will omit them as we won't be needing those in our models.
3. Sunspot is a univariate dataset that records monthly observations of the sun's surface dating from 1749 to 2021 with 3265 data points, where the number of sunspots fluctuates and follows an approximate 11-year cycle.
4. The Mackey-Glass is a univariate dataset that features a continuous chaotic time series, computed using the following delayed differential equation [79]. In this study, we use Mackey-Glass [80] parameters ($a = 0.2$, $b = 0.1$ and $\tau = 10$) and generate 3000 data points.
5. The Lorenz equations [81] are a three-dimensional chaotic time series composed of ordinary differential equations, inherently unpredictable over long periods. We have used the default values of the Lorenz system ($\rho = 28$, $\sigma = 10$ and $\beta = 2.667$) and generated 10,000 data points.

### 3.5. Experiment setup

After developing the initial models, we considered several factors for selecting the appropriate hyperparameters for each model type. Since Bitcoin and Ethereum are highly volatile, training on continuous data would not adequately prepare the model for handling such fluctuations. Therefore, we created the training dataset using a split that was randomly selected, i.e. 80:20 ratio. The reason for the random train test split is to ensure the models account for data across all time periods. For

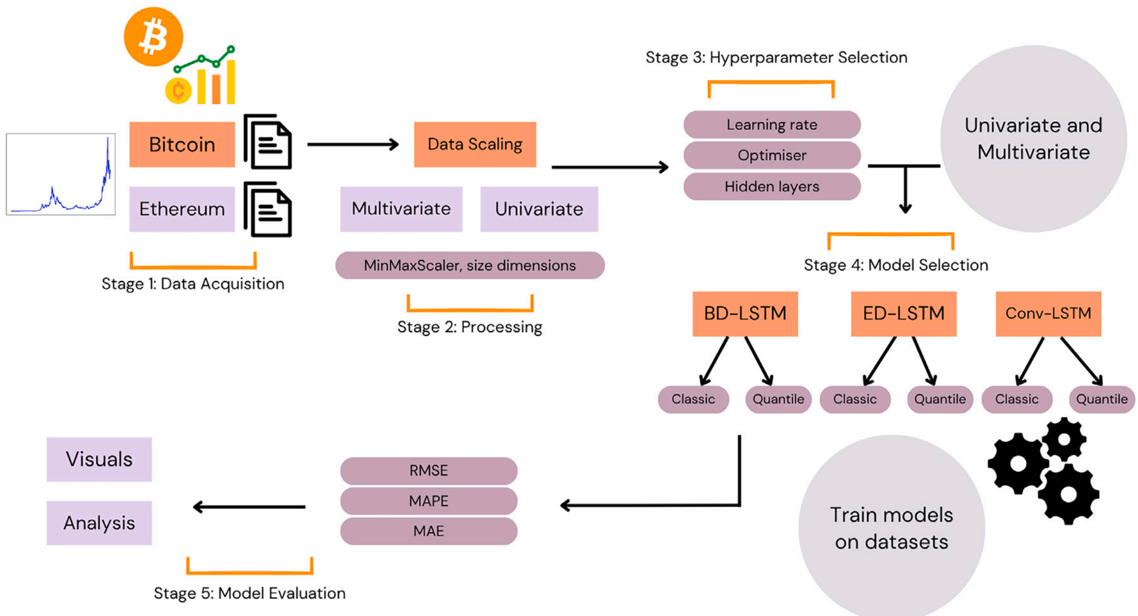
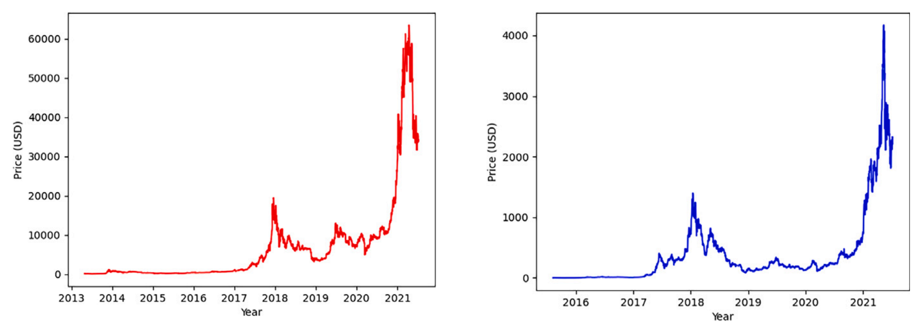



**Fig. 4.** Framework diagram showing the key stages that include data processing model training and evaluation. We present a quantile-based implementation for a set of deep learning models, including BD-LSTM, Conv-LSTM, and ED-LSTM.




(a) Bitcoin (b) Ethereum

**Fig. 5.** Cryptocurrency time series reporting daily close prices.

instance, cryptocurrency data is especially volatile during the COVID-19 pandemic period, which falls only in the test dataset if the train-test split wasn't implemented. Our goal is to ensure that the respective models have the ability to manage volatile data effectively.

We kept the models consistent with previous work ([46]) and hence used the hyperparameters presented in Table 1. In the respective deep learning models, we use adaptive moment estimation (Adam) [82] optimiser for training with a learning rate of 0.0001. In the case of the cryptocurrency datasets (Bitcoin and Ethereum), we use 6 as the input window size with 5 outputs (5 prediction horizons) as done by Wu et al. [70]. In other real-world and simulated time series datasets, the input and output window sizes were adjusted to allow comparison with related work by Chandra et al. [46], where the input window is fixed at 5, and the output window at 10 (10 prediction horizons). Furthermore, the following needs to be taken into account along with the information in Table 1.

- The BD-LSTM model includes both the forward and backward LSTM layer.
- ED-LSTM includes two LSTM networks with a time distributed layer, in the Encoder and Decoder submodels.
- The Conv-LSTM includes a 1D convolutional layer for the univariate time series and a 2D layer for the multivariate time series. In the convolutional layer, we use 64 filters with a kernel size of 2. It also utilises an LSTM network and a dense layer.

**Table 1**
Model architecture for univariate/multivariate strategy for the respective deep learning models for the cryptocurrency datasets. We define the number of neurons in the input layer (number of features $f$ and window size $d$), hidden layers $h_1, h_2$, and the output layer. Note that the number of neurons in the input layer and output layer vary across the different datasets.

| Model | Strategy | Input | Hidden layers | Output |
|---|---|---|---|---|
| BD-LSTM | Univariate | $[f = 1, d = 6]$ | $[h_2 = 50, h_1 = 50]$ | 5 |
| | Multivariate | $[f = 6, d = 6]$ | $[h_1 = 50, h_2 = 50]$ | 5 |
| ED-LSTM | Univariate | $[f = 1, d = 6]$ | $[h_1 = 100, h_2 = 100]$ | 5 |
| | Multivariate | $[f = 6, d = 6]$ | $[h_1 = 100, h_2 = 100]$ | 5 |
| Conv-LSTM | Univariate | $[f = 1, d = 6]$ | $[h_2 = 20, h_1 = 20]$ | 5 |
| | Multivariate | $[f = 6, d = 6]$ | $[h_1 = 20, h_2 = 20]$ | 5 |

We report the RMSE mean and 95 % confidence intervals from the test dataset based on 30 independent experimental runs. We note that a lower RMSE indicates better model performance and high uncertainty is indicated by a high confidence interval. In the case of our quantile-based deep learning models, we calculate the average RMSE across the

**Table 2**
Prediction accuracy on the Bitcoin dataset, reporting accuracy (mean RMSE and ± 95 % confidence interval) for 30 independent model training runs. * median across quantile (0.5).

| Model | Mean | Step 1 | Step 2 | Step 3 | Step 4 | Step 5 |
|---|---|---|---|---|---|---|
| Linear regression | | | | | | |
| Univariate | 0.0062 ± 0.0058 | 0.0068 ± 0.0105 | 0.0076 ± 0.0185 | 0.0076 ± 0.0149 | 0.0062 ± 0.0110 | 0.0029 ± 0.0040 |
| Quantile Univariate | **0.0059 ± 0.0055*** | 0.0067 ± 0.0103 | **0.0070 ± 0.0172** | 0.0075 ± 0.0152 | **0.0058 ± 0.0099** | 0.0027 ± 0.0044 |
| Multivariate | 0.0061 ± 0.0056 | **0.0066 ± 0.0101** | 0.0079 ± 0.0185 | **0.0069 ± 0.0126** | 0.0064 ± 0.0123 | **0.0027 ± 0.0042** |
| Quantile Multivariate | 0.0158 ± 0.0021* | 0.0092 ± 0.0011 | 0.0129 ± 0.0013 | 0.0163 ± 0.0017 | 0.0193 ± 0.0017 | 0.0215 ± 0.0022 |
| Univariate deep learning models | | | | | | |
| BD-LSTM | 0.0155 ± 0.0018 | 0.0098 ± 0.0009 | 0.0130 ± 0.0012 | 0.0159 ± 0.0017 | 0.0185 ± 0.0018 | 0.0204 ± 0.0014 |
| Quantile BD-LSTM | 0.0153 ± 0.0018* | 0.0096 ± 0.0011 | 0.0127 ± 0.0013 | 0.0157 ± 0.0018 | 0.0182 ± 0.0018 | 0.0203 ± 0.0015 |
| Conv-LSTM | 0.0152 ± 0.0018 | 0.0093 ± 0.0009 | 0.0127 ± 0.0011 | 0.0156 ± 0.0012 | 0.0182 ± 0.0016 | 0.0202 ± 0.0016 |
| Quantile Conv-LSTM | 0.0153 ± 0.0019* | **0.0092 ± 0.0009** | 0.0127 ± 0.0011 | 0.0157 ± 0.0012 | 0.0183 ± 0.0017 | 0.0204 ± 0.0018 |
| ED-LSTM | 0.0108 ± 0.0006 | 0.0106 ± 0.0013 | 0.0107 ± 0.0012 | 0.0110 ± 0.0011 | **0.0110 ± 0.0012** | 0.0109 ± 0.0016 |
| Quantile ED-LSTM | **0.0107 ± 0.0006*** | 0.0103 ± 0.0012 | **0.0105 ± 0.0011** | **0.0109 ± 0.0012** | **0.0110 ± 0.0012** | **0.0107 ± 0.0014** |
| Multivariate deep learning models | | | | | | |
| BD-LSTM | 0.0163 ± 0.0017 | 0.0109 ± 0.0013 | 0.0139 ± 0.0015 | 0.0167 ± 0.0014 | 0.0191 ± 0.0016 | 0.0209 ± 0.0015 |
| Quantile BD-LSTM | 0.0159 ± 0.0018* | 0.0102 ± 0.0013 | 0.0137 ± 0.0014 | 0.0164 ± 0.0016 | 0.0187 ± 0.0015 | 0.0205 ± 0.0016 |
| Conv-LSTM | 0.0160 ± 0.0017 | 0.0107 ± 0.0014 | 0.0137 ± 0.0015 | 0.0164 ± 0.0015 | 0.0186 ± 0.0016 | 0.0204 ± 0.0015 |
| Quantile Conv-LSTM | 0.0165 ± 0.0017* | 0.0113 ± 0.0015 | 0.0144 ± 0.0017 | 0.0170 ± 0.0014 | 0.0191 ± 0.0015 | 0.0209 ± 0.0015 |
| ED-LSTM | 0.0113 ± 0.0007 | **0.0101 ± 0.0013** | 0.0112 ± 0.0012 | 0.0118 ± 0.0013 | 0.0119 ± 0.0014 | 0.0117 ± 0.0018 |
| Quantile ED-LSTM | **0.0112 ± 0.0005*** | 0.0110 ± 0.0009 | **0.0110 ± 0.0011** | **0.0112 ± 0.0012** | **0.0114 ± 0.0012** | **0.0113 ± 0.0013** |

number of time steps at each quantile (0.05, 0.25, 0.5, 0.75, and 0.95), with the mean representing the median value (0.5).

## 4. Results

We evaluate quantile deep learning models for time series prediction, including BD-LSTM, ED-LSTM, and Conv-LSTM as the base models and their corresponding quantile versions (e.g. Quantile BD-LSTM).

### 4.1. Cryptocurrency datasets

Table 2 presents the performance (RMSE) of univariate and multivariate linear regression and deep learning models (BD-LSTM, ED-LSTM, Conv-LSTM) for the Bitcoin test dataset. We highlight in bold the best performance for the respective prediction horizons. We observe that the quantile linear regression accuracy (RMSE) is similar to linear regression (mean and prediction horizons given by the steps). This implies that quantile regression can effectively handle the volatility of cryptocurrency data while providing predictions of the respective quantiles, which accounts for uncertainty quantification. An interesting observation can be seen for the multivariate strategy, where there is a higher mean RMSE but a more condensed confidence interval for the multivariate quantile linear model.

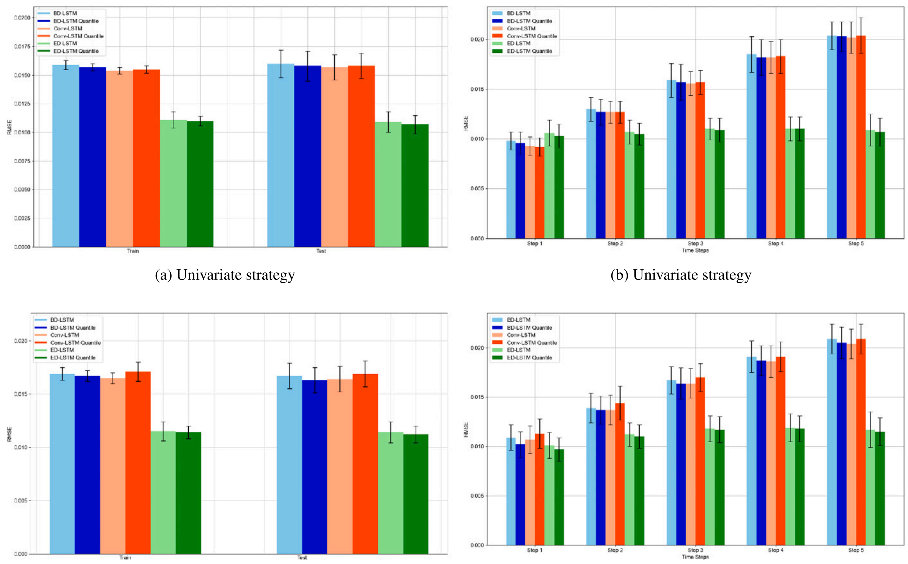
Greaves et al. [83] demonstrated that neural networks are a superior model for classification than linear regression for Bitcoin price prediction; hence, we move on to deep learning models. Earlier, Wu et al. [70] showed that the BD-LSTM, ED-LSTM, and Conv-LSTM networks provided the best accuracy ranks in the univariate and multivariate strategies for a wider range of deep learning models.

Across both univariate and multivariate strategies in Table 2, the ED-LSTM and quantile ED-LSTM models provide the highest prediction accuracy, and consistently, the ED-LSTM models outperform BD-LSTM and Conv-LSTM. Specifically, the Quantile-ED-LSTM model provides the best accuracy for all prediction horizons, except step one in the univariate and multivariate strategies. Additionally, in Fig. 6 (b), (d), we can observe that ED-LSTM and Quantile-ED–LSTM both provide consistent accuracy as the prediction horizon changes, thus being the most robust and stable model. We also find that BD-LSTM and Conv-LSTM provide similar performance, but the Quantile-BD-LSTM model consistently provides higher accuracy than its counterpart (BD-LSTM) across both univariate and multivariate strategies. In Table 3, we can see the accuracy (mean RMSE) for each quantile, and not only quantile models often provide similar predictions, but they also provide further information (quantiles) for uncertainty quantification.

In the Ethereum dataset, Table 5 presents the performance (RMSE) of linear regression and deep learning models for the test datasets, with the best performance highlighted in bold. We observe that the Univariate quantile models provide the best accuracy (RMSE), and the Multivariate quantile models provide the most robustness, as indicated by consistently low confidence intervals. Fig. A.10 presents a visualisation of predictions for the respective models and quantiles for the Ethereum time series, where we observe that the quantiles well capture the actual data points. In Figs. 7 (c) and (d), the Multivariate strategy shows that consistently the ED-LSTM models outperform BD-LSTM and Conv-LSTM. In contrast to Bitcoin (Fig. 6, the classic ED-LSTM model provides the most accurate predictions overall for all time horizons, with the lowest RMSE value.

Moreover, in Fig. 7 (a) and (b), ED-LSTM models in the univariate strategy consistently outperform BD-LSTM and Conv-LSTM in prediction accuracy (except on step one), and Quantile-BD-LSTM provides the best prediction accuracy. Furthermore, the Quantile-ED-LSTM is the most robust univariate model for predicting Bitcoin as the prediction horizon increases. We note that although BD-LSTM and Conv-LSTM present consistent results, as the number of prediction days increases, the forecast accuracy gradually decreases. The Quantile-BD-LSTM and Quantile-Conv-LSTM models also consistently provide higher accuracy than their counterparts for the univariate strategy. Finally, Conv-LSTM exhibits better performance than the Quantile-Conv-LSTM for multivariate strategies.

Table 4 outlines multivariate and univariate strategies for the Ethereum dataset (test dataset - mean RMSE across 5 time steps at different quantiles) for 30 independent model training runs. Since the median quantile is our prediction, it has the lowest RMSE compared to any other quantiles. This is logically consistent, as other quantiles cover more extreme prediction values. Additionally, it has a much smaller confidence interval, which highlights that quantile models excel in reducing percentage-based errors, making them particularly effective in dealing with price fluctuations and the inherent volatility of cryptocurrency markets. In Fig. A.10, we present selected predictions for the given quantiles, where we observe that in Fig. A.10 (c), the Quantile-Conv-LSTM model performs accurately. Although the Conv-LSTM model failed to capture



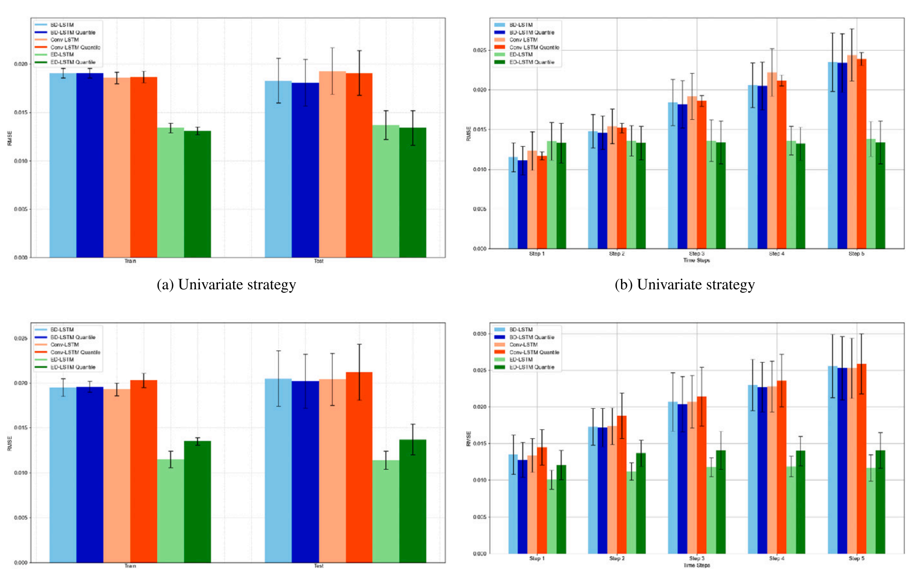
(a) Univariate strategy (b) Univariate strategy

(c) Multivariate strategy (d) Multivariate strategy

**Fig. 6.** Bitcoin time series: prediction plots of respective univariate and multivariate strategies (RMSE mean with 95 % confidence interval given as error bar).

**Table 3**
Bitcoin prediction accuracy (mean RMSE across 5 time steps) at different quantiles.

| Strategy | Model | Quantile | | | | |
|---|---|---|---|---|---|---|
| | | 0.05 | 0.25 | 0.5 | 0.75 | 0.95 |
| Univariate | Quantile BD-LSTM | 0.0339 ± 0.0032 | 0.0195 ± 0.0022 | 0.0158 ± 0.0013 | 0.0201 ± 0.0021 | 0.0374 ± 0.0039 |
| | Quantile Conv-LSTM | 0.0307 ± 0.0022 | 0.0189 ± 0.0014 | 0.0158 ± 0.0011 | 0.0190 ± 0.0015 | 0.0296 ± 0.0022 |
| | Quantile ED-LSTM | 0.0272 ± 0.0032 | 0.0153 ± 0.0019 | 0.0134 ± 0.0018 | 0.0161 ± 0.0026 | 0.0258 ± 0.0037 |
| Multivariate | Quantile BD-LSTM | 0.0313 ± 0.0033 | 0.0198 ± 0.0025 | 0.0163 ± 0.0012 | 0.0197 ± 0.0019 | 0.0314 ± 0.0030 |
| | Quantile Conv-LSTM | 0.0322 ± 0.0038 | 0.0203 ± 0.0029 | 0.0169 ± 0.0012 | 0.0201 ± 0.0021 | 0.0318 ± 0.0033 |
| | Quantile ED-LSTM | 0.0226 ± 0.0028 | 0.0133 ± 0.0019 | 0.0112 ± 0.0008 | 0.0131 ± 0.0014 | 0.0206 ± 0.0023 |



(a) Univariate strategy (b) Univariate strategy

(c) Multivariate strategy (d) Multivariate strategy

**Fig. 7.** Ethereum performance evaluation of respective Univariate and Multivariate deep learning models for 5 prediction horizons (mean RMSE with 95 % confidence interval as error bar).

**Table 4**
Results for the Ethereum dataset reporting accuracy (mean RMSE and ± 95 % confidence interval) for 30 experimental runs for each model. * represents the median across quantiles (0.5).

| Model | Mean | Step 1 | Step 2 | Step 3 | Step 4 | Step 5 |
|---|---|---|---|---|---|---|
| Linear regression | | | | | | |
| Univariate | 0.0079 ± 0.0060 | 0.0071 ± 0.0093 | 0.0070 ± 0.0111 | 0.0092 ± 0.0126 | 0.0105 ± 0.0187 | 0.0055 ± 0.0124 |
| Quantile Univariate | **0.0075 ± 0.0061*** | **0.0068 ± 0.0097** | **0.0067 ± 0.0111** | **0.0089 ± 0.0130** | **0.0103 ± 0.0195** | **0.0050 ± 0.0126** |
| Multivariate | 0.0090 ± 0.0077 | 0.0071 ± 0.0098 | 0.0090 ± 0.0149 | 0.0102 ± 0.0126 | 0.0112 ± 0.0214 | 0.0074 ± 0.0228 |
| Quantile Multivariate | 0.0201 ± 0.0030* | 0.0115 ± 0.0025 | 0.0162 ± 0.0027 | 0.0213 ± 0.0042 | 0.0242 ± 0.0039 | 0.0272 ± 0.0052 |
| Univariate deep learning models | | | | | | |
| BD-LSTM | 0.0178 ± 0.0022 | 0.0115 ± 0.0018 | 0.0149 ± 0.0021 | 0.0184 ± 0.0029 | 0.0206 ± 0.0028 | 0.0235 ± 0.0037 |
| Quantile BD-LSTM | 0.0176 ± 0.0023* | **0.0112 ± 0.0018** | 0.0146 ± 0.0021 | 0.0182 ± 0.0030 | 0.0205 ± 0.0029 | 0.0233 ± 0.0038 |
| Conv-LSTM | 0.0187 ± 0.0023 | 0.0123 ± 0.0024 | 0.0154 ± 0.0022 | 0.0193 ± 0.0029 | 0.0221 ± 0.0030 | 0.0244 ± 0.0033 |
| Quantile Conv-LSTM | 0.0186 ± 0.0022* | 0.0124 ± 0.0024 | 0.0154 ± 0.0023 | 0.0192 ± 0.0029 | 0.0218 ± 0.0027 | 0.0241 ± 0.0031 |
| ED-LSTM | 0.0136 ± 0.0010 | 0.0135 ± 0.0024 | 0.0136 ± 0.0019 | 0.0136 ± 0.0026 | 0.0136 ± 0.0018 | 0.0138 ± 0.0022 |
| Quantile ED-LSTM | **0.0133 ± 0.0011*** | 0.0133 ± 0.0025 | **0.0133 ± 0.0021** | **0.0134 ± 0.0027** | **0.0132 ± 0.0021** | **0.0134 ± 0.0027** |
| Multivariate deep learning models | | | | | | |
| BD-LSTM | 0.0200 ± 0.0024 | 0.0134 ± 0.0024 | 0.0173 ± 0.0024 | 0.0206 ± 0.0039 | 0.0229 ± 0.0034 | 0.0256 ± 0.0044 |
| Quantile BD-LSTM | 0.0197 ± 0.0025* | 0.0128 ± 0.0025 | 0.0171 ± 0.0024 | 0.0204 ± 0.0038 | 0.0228 ± 0.0035 | 0.0252 ± 0.0042 |
| Conv-LSTM | 0.0199 ± 0.0024 | 0.0134 ± 0.0023 | 0.0174 ± 0.0025 | 0.0207 ± 0.0036 | 0.0228 ± 0.0035 | 0.0253 ± 0.0041 |
| Quantile Conv-LSTM | 0.0208 ± 0.0024* | 0.0145 ± 0.0024 | 0.0188 ± 0.0031 | 0.0214 ± 0.0040 | 0.0236 ± 0.0036 | 0.0259 ± 0.0041 |
| ED-LSTM | **0.0113 ± 0.0007** | **0.0101 ± 0.0013** | **0.0112 ± 0.0012** | **0.0118 ± 0.0013** | **0.0119 ± 0.0014** | **0.0117 ± 0.0018** |
| Quantile ED-LSTM | 0.0126 ± 0.0008* | 0.0121 ± 0.0020 | 0.0121 ± 0.0020 | 0.0119 ± 0.0014 | 0.0132 ± 0.0016 | 0.0135 ± 0.0018 |

**Table 5**
Performance evaluation of Multivariate and Univariate strategies for the Ethereum dataset (test dataset mean RMSE across 5-time steps at different quantiles) for 30 independent model training runs.

| Strategy | Model | Quantile | | | | |
|---|---|---|---|---|---|---|
| | | 0.05 | 0.25 | 0.5 | 0.75 | 0.95 |
| Univariate | Quantile BD-LSTM | 0.0372 ± 0.0039 | 0.0219 ± 0.0025 | 0.0181 ± 0.0024 | 0.0221 ± 0.0035 | 0.0390 ± 0.0045 |
| | Quantile Conv-LSTM | 0.0368 ± 0.0052 | 0.0226 ± 0.0034 | 0.0191 ± 0.0023 | 0.0222 ± 0.0028 | 0.0353 ± 0.0040 |
| | Quantile ED-LSTM | 0.0272 ± 0.0032 | 0.0153 ± 0.0019 | 0.0134 ± 0.0018 | 0.0161 ± 0.0026 | 0.0258 ± 0.0037 |
| Multivariate | Quantile BD-LSTM | 0.0385 ± 0.0042 | 0.0228 ± 0.0025 | 0.0202 ± 0.0030 | 0.0249 ± 0.0053 | 0.0383 ± 0.0056 |
| | Quantile Conv-LSTM | 0.0392 ± 0.0041 | 0.0241 ± 0.0025 | 0.0212 ± 0.0031 | 0.0267 ± 0.0057 | 0.0405 ± 0.0063 |
| | Quantile ED-LSTM | 0.0265 ± 0.0031 | 0.0155 ± 0.0020 | 0.0137 ± 0.0017 | 0.0168 ± 0.0029 | 0.0270 ± 0.0039 |

the actual values, its quantile counterpart improved the model's ability to capture the true values.

### 4.2. Benchmark datasets

Next, we evaluate our framework for the benchmark datasets that include Sunspots, Mackey-Glass and Lorenz. Chandra et al. [46] demonstrated that BD-LSTM and ED-LSTM provided the best accuracy ranks in the evaluation of selected univariate deep learning models. Fig. 8 presents 10-step prediction horizons of the BD-LSTM and ED-LSTM models along with their quantile variants. In Figs. 8 (a)-(c), we observe that the BD-LSTM is the least robust and has the largest error margins (95 % confidence intervals). Additionally, the ED-LSTM and Quantile-ED-LSTM provide a consistent level of accuracy across the prediction horizon, whereas BD-LSTM models are less accurate as time steps increase.

Table 6 presents the prediction accuracy for the ED-LSTM and Quantile-ED-LSTM models; both models provided very similar results. In the Sunspot and Lorenz datasets, ED-LSTM models provide the best mean RMSE accuracy, and Quantile ED-LSTM models exhibit the best performance for the Mackey-Glass time series. We note that there is a distinct difference between ED-LSTM models and the rest of the field, clearly demonstrating that it is the favourable model across various time series datasets. Note that our goal for the Quantile-ED-LSTM is to achieve a similar level of performance to ED-LSTM while providing predictions for the different quantiles as provided.

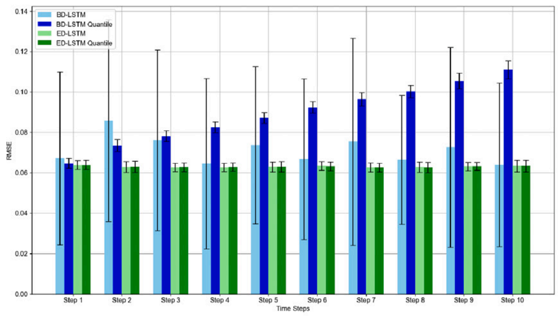
We provide prediction visualisations for all datasets in Figs. A.9–A.11. All models and their prediction prowess were showcased, often times falling into the quantile range. Note that models tend to perform well for large, consistent prices. For low values outputs such as sunspots, Mackey-Glass and Lorenz datasets, we noticed that quantile prediction often deviates from actual values.

## 5. Discussion

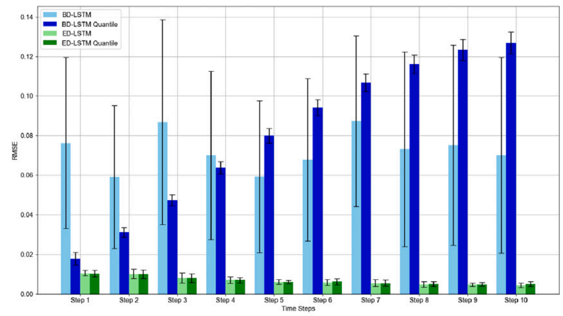
### 5.1. Summary of results

This study explored quantile deep learning models for multivariate and multi-step ahead time series prediction with univariate and multivariate models for selected cryptocurrency and time series prediction datasets. Table 7 presents a summary of the results, highlighting that both the conventional ED-LSTM and Quantile-ED-LSTM models consistently deliver the strongest predictive performance across all datasets. We find that for the cryptocurrency datasets, neither the BD-LSTM nor the Conv-LSTM (including their quantile variants) shows a clear performance hierarchy. This aligns with the RMSE results outlined in Tables 2 and 4, where all four models exhibit competitive accuracy (close performance). Additionally, the primary goal of this study is to enhance the representation of uncertainty with the quantile loss function, rather than to improve the forecast accuracy of the conventional models. Therefore, we expected a similar performance of quantile models relative to conventional deep learning models, and has been demonstrated across all datasets in Figs. 6–8.

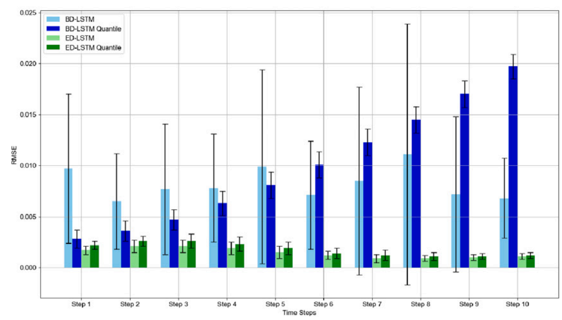
After obtaining the model rankings, we aim to quantify the difference in improvement by running statistical tests. By deriving approximate p-values [84] from the mean and 95 % confidence interval, we take the




(a) Sunspot




(b) Mackey-Glass 10 step-ahead prediction




(c) Lorenz 10 step-ahead prediction

**Fig. 8.** Performance evaluation of respective Univariate deep learning models, showing 10 step-ahead prediction horizons for benchmark datasets for 30 independent model training runs (mean RMSE with 95 % confidence interval as error bar).

**Table 6**
Benchmark time series datasets reporting model test accuracy (mean RMSE and ± 95 % confidence interval for 30 experimental runs) for univariate deep learning models. * represents the median prediction quantile (0.5).

| Model | Mean | Step 2 | Step 5 | Step 8 | Step 10 |
|---|---|---|---|---|---|
| Sunspot | | | | | |
| BD-LSTM | 0.0712 ± 0.0139 | 0.0858 ± 0.0499 | 0.0736 ± 0.0389 | 0.0664 ± 0.0319 | 0.0640 ± 0.0405 |
| Quantile BD-LSTM | 0.0892 ± 0.0045* | 0.0735 ± 0.0029 | 0.0871 ± 0.0027 | 0.1003 ± 0.0030 | 0.1111 ± 0.0045 |
| ED-LSTM | **0.0630** ± **0.0008** | **0.0628** ± **0.0027** | **0.0629** ± **0.0025** | 0.0627 ± 0.0026 | 0.0633 ± 0.0029 |
| Quantile ED-LSTM | 0.0630 ± 0.0024* | 0.0629 ± 0.0028 | 0.0630 ± 0.0026 | **0.0626** ± **0.0026** | **0.0633** ± **0.0029** |
| Mackey-Glass | | | | | |
| BD-LSTM | 0.0725 ± 0.0145 | 0.0591 ± 0.0362 | 0.0592 ± 0.0385 | 0.0732 ± 0.0492 | 0.0701 ± 0.0495 |
| Quantile BD-LSTM | 0.0807 ± 0.0119* | 0.0311 ± 0.0024 | 0.0800 ± 0.0037 | 0.1161 ± 0.0046 | 0.1269 ± 0.0055 |
| ED-LSTM | 0.0071 ± 0.0011 | 0.0101 ± 0.0023 | 0.0060 ± 0.0011 | **0.0048** ± **0.0014** | **0.0049** ± **0.0012** |
| Quantile ED-LSTM | **0.0071** ± **0.0010*** | **0.0100** ± **0.0021** | **0.0059** ± **0.0008** | 0.0049 ± 0.0012 | 0.0050 ± 0.0012 |
| Lorenz | | | | | |
| BD-LSTM | 0.0097 ± 0.0017 | 0.0072 ± 0.0053 | 0.0111 ± 0.0097 | 0.0097 ± 0.0086 | 0.0076 ± 0.0055 |
| Quantile BD-LSTM | 0.0140 ± 0.0013* | 0.0044 ± 0.009 | 0.0104 ± 0.0013 | 0.0177 ± 0.0018 | 0.0239 ± 0.0022 |
| ED-LSTM | **0.0015** ± **0.0004** | **0.0022** ± **0.0007** | **0.0014** ± **0.0006** | **0.0009** ± **0.0004** | **0.0010** ± **0.0004** |
| Quantile ED-LSTM | 0.0021 ± 0.0004* | 0.0028 ± 0.0005 | 0.0020 ± 0.0007 | 0.0013 ± 0.0004 | 0.0013 ± 0.0004 |

**Table 7**
Performance (rank) of different models for respective time-series problems. Note lower rank denotes better performance.

| Data | Strategy | BD-LSTM | Quantile BD-LSTM | Conv-LSTM | Quantile Conv-LSTM | ED-LSTM | Quantile ED-LSTM |
|---|---|---|---|---|---|---|---|
| Bitcoin | Univariate | 6 | 5 | 3 | 4 | 1 | 2 |
| Ethereum | | 4 | 3 | 6 | 5 | 2 | 1 |
| Mean Rank | | 5 | 4 | 4.5 | 4.5 | **1.5** | **1.5** |
| Bitcoin | Multivariate | 5 | 3 | 4 | 6 | 2 | 1 |
| Ethereum | | 5 | 3 | 4 | 6 | 2 | 1 |
| Mean Rank | | 5 | 3 | 4 | 6 | 2 | **1** |
| Sunspot | Univariate | 3 | 4 | – | – | 1 | 2 |
| Mackey-Glass | | 3 | 4 | – | – | 2 | 1 |
| Lorenz | | 3 | 4 | – | – | 1 | 2 |
| Mean Rank | | 3 | 4 | – | – | **1.33** | 1.67 |

mean RMSE values from each univariate model for the Ethereum dataset to compare (Table 4). Although mean RMSE suggested that quantile models performed better than their counterparts, this difference in results is not significant (p ≈ 0.69, 0.90, 0.95 for ED-LSTM, BD-LSTM and Conv-LSTM, respectively). Therefore, the quantile models provide the same performance when compared to the conventional models, while projecting risk and uncertainty using the quantiles. Furthermore, we acknowledge that there are more rigorous paired analyses that can be performed, such as the paired *t*-test [85], the Wilcoxon signed rank test [86], ANOVA [87] and the Friedman test [88].

In both univariate and multivariate cases, the quantile models performed similarly to the conventional deep learning models. In the case of Ethereum, the performance for the quantile models is slightly poorer than that of conventional models (Table 4). However, in the Bitcoin dataset (Table 2), the predictions remained similar to the univariate datasets, demonstrating that deep learning models have different predictive abilities depending on the dataset. We can gather that the model with the most consistent and accurate predictions is the Quantile-ED-LSTM model. The best models in the cryptocurrency datasets do not automatically imply they are the best across all other datasets. We have run similar experiments on three other volatile datasets (i.e. sunspots, Mackey-Glass and Lorenz), and in particular, the BD-LSTM standard model proved to be particularly unreliable, with very large confidence intervals and RMSE values (Fig. 8). The BD-LSTM quantile model deteriorated in RMSE value across the time steps but featured a small confidence interval, demonstrating the model's robustness. Consistent with the cryptocurrency results, the ED-LSTM models outperformed the BD-LSTM models as seen in Table 6, where the ED-LSTM models have a higher mean rank than the BD-LSTM models across all datasets.

We next review the multivariate results for the cryptocurrency datasets (Tables 2 and 4), where the distinguishing features between each model and their quantile counterparts have more clarity and definition. The multivariate results remained consistent with the univariate parts, where the ED-LSTM outperforms both the Conv-LSTM and BD-LSTM models. We can observe the contrast between the three models in Figs. 6 and 7. Since ED-LSTM models process the entirety of the given historical data through the encoder and then make predictions through the decoder, they can take into account the entire history of the dataset and hence make more reliable predictions. BD-LSTM model also performs poorly in comparison to the ED-LSTM models (Table 7), as they are not as capable in handling long-term dependencies in the data, which is a key feature in volatile datasets.

We note that our findings in this paper were given in RMSE and its 95 % confidence interval for 30 experimental runs. Among them, the median predictions ($\alpha = 0.5$) returned the lowest RMSE values compared to other quantile values $\alpha$. This result is to be expected since we have not yet applied our pinball loss function (Equation 3) to skew the quantile weight function up or down. Following a similar logic, we observed that the RMSE values increase for more extreme quantiles since those values serve as the higher or lower range of the prediction, hence they will have a larger difference from the true value.

### 5.2. Significance of study

Our findings have better results than those reported in the related study [70], as their test mean for the ED-LSTM multivariate Bitcoin data was 0.0373 compared to our quantile ED-LSTM result of 0.0112, as seen in Table 2. However, do note that the train test split ratios are different across the two papers, where we used 80:20 in comparison to 70:30 by Wu et al [70]. The lower mean RMSE from our ED-LSTM quantile regression analysis particularly emphasises that the median (0.5 quantile) results yield superior predictive accuracy compared to traditional methods. This enhancement in performance underscores the robustness of quantile regression for time series forecasting. Furthermore, quantile regression not only improves prediction accuracy but also offers a probabilistic interpretation by providing a spectrum of potential outcomes, thereby enriching the decision-making process with a more comprehensive risk assessment. In the literature, conventional deep learning models have been evaluated for multi-step ahead time series prediction on Sunspots, Mackey-Glass and Lorenz systems [46], where BD-LSTM had the best accuracy for univariate time series data. Furthermore, Chandra et al. [46] reported a common trend where the predictive accuracy decreases across higher steps-ahead prediction. Note that in their results, in the case of Mackey-Glass, ED-LSTM was the best-performing model, and for the Lorenz system, both ED-LSTM and LSTM outperformed BD-LSTM. This is in line with the observations in our study, where ED-LSTM is the best-performing model. We recall that the process of extreme value forecasting (EVF) [91,92] does not directly calculate extreme values [93], and instead calculates uncertainty bounds [94], which implicitly addresses extreme values. Our quantile deep learning models for multi-step ahead prediction serve as an example of EVF in action, as the quantile nature of the model indicates uncertainty.

### 5.3. Limitations and future work

Although the quantile models are marginally better as they showcase risk and uncertainty to the problem at hand, the computational overhead (training/inference time) tradeoffs are noticeable. We were essentially training five models for quantile regression due to the input of the tiled loss function, and hence, the computational time increased five-fold. However, with better code implementation and parallel computing [99], we can decrease the difference in computation time. Another issue to address is quantile crossing, where quantile curves are crossing each other. For example, $q_{0.9}(\hat{x}) < q_{0.5}(\hat{x})$, which violates the basic property as higher quantiles should never be lower than smaller quantiles. Quantile crossing is common in quantile regression because the models are trained separately for each quantile level $\tau$; thus, it is prone to fluctuate independently. This issue can be resolved by post-processing and setting constraints during estimation [100].

Furthermore, we need to compute the difference in each predicted $\hat{x}_i$ and actual values $x_i$ to quantify the quality of quantile predictions, such as interval coverage [89] and interval score [90]. We also need to determine whether the true value falls into the estimated range of $[q_\alpha(\hat{x}_i), q_{1-\alpha}(\hat{x}_i)]$. We can deploy quantile coverage to assess the accuracy of the predicted quantiles, which is similar to interval coverage. It determines the percentage that the actual value falls between the predicted intervals. For example, if we have an observation with a predicted interval of $[q_{0.25}(\hat{x}_i), q_{0.75}(\hat{x}_i)]$, we should expect the actual value to be in this interval 50 % of the time.

Regularisation techniques such as Lasso [96] and ridge regression [97] can be incorporated into our framework to enhance model accuracy. Dropout-based regularisation [98] has been prominent in deep learning, and this can also be incorporated in the respective model architectures with further hyperparameter tuning [95]. Extensive evaluation using different training-test split ratios, cross-validation, and different input dimension windows for time series data embedding can be considered. Finally, model outputs and feature variable assumptions need to be clearly defined and constrained to ensure quantile predictions adequately cover the expected range.

In future work, several strategies can be taken to enhance our quantile deep learning framework further. The quantile deep learning model for predicting cryptocurrency can be further enhanced using a multimodal [101] framework that considers text data, such as data from news media and Twitter about cryptocurrency markets. Sentiment analysis using natural language processing and large language models [102] can be useful in providing further information for the deep learning models. Sentiment analysis in combination with a quantile deep learning can be instrumental in improving future predictions with quantiles for robust uncertainty quantification.

There are other approaches to uncertainty quantification, such as Bayesian inference [103], where the posterior distribution is obtained by prior distribution and the likelihood function [104]. In particular, the uncertainty bound can be calculated by sampling from the posterior distribution using Markov Chain Monte Carlo (MCMC) [105]. There have been efforts in developing Bayesian neural networks and Bayesian deep learning models with MCMC [106] and variational-Bayes sampling strategies [107]. There exists a study on multi-step ahead stock price forecasting using the Bayesian neural networks taking into account the price fluctuations given uncertainties during COVID-19 [68]. Although our study provides a frequentist approach [108] to uncertainty quantification using quantile regression in deep learning, our framework can be extended using Bayesian deep learning. This can be done by sampling from the posterior distribution (model weights and biases) using MCMC or variational Bayes. Although this would be an 'overkill' in conventional problems, it would be useful in problems where risk analysis is vital, such as medical diagnosis. Furthermore, quantile deep learning models can be utilised for time series data imputation tasks [109], where uncertainties obtained from different quantiles can be useful in producing different versions of imputed datasets.

## 6. Conclusions

In this study, we investigated the combination of quantile regression in selected deep learning models for multi-step ahead time series prediction. Our results demonstrated that combining quantile regression with deep learning models has been very effective, even in volatile environments such as the cryptocurrency markets. In the case of the cryptocurrency datasets, the quantile ED-LSTM model has consistently outperformed traditional methods, highlighting their ability to effectively handle multi-step forecast uncertainty and volatility. Although our current models have shown strong predictive capabilities, it is still possible to improve them further through further hyperparameter tuning and incorporating novel architectural and training strategies. Not only do our quantile models provide exceptional accuracy, but they also demonstrate remarkable stability and robustness across various prediction horizons. Our results also show that combining the quantile loss function with deep learning does occasionally provide slightly more accurate predictions. The quantile models offer a more reliable prediction by embracing the inherent uncertainties with the quantiles and being less sensitive to outliers. This allows us to be more certain within our predictions, making them invaluable tools for navigating datasets that are unpredictable and volatile in nature.

Our study shows that quantile deep learning models provide an assesment of uncertainty, which is useful in risk assessment and decision-making processes. This process is critical in dealing with high volatility and extreme data in risk-sensitive environments and has potential for modelling climate extreme events. Future research can explore model optimisation and applications to other forecasting problems and regression tasks.

## CRediT authorship contribution statement

**Jimmy Cheung:** Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation. **Smruthi Rangarajan:** Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation. **Amelia Maddocks:** Writing – original draft, Software, Investigation, Data curation. **Rohitash Chandra:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Conceptualization.

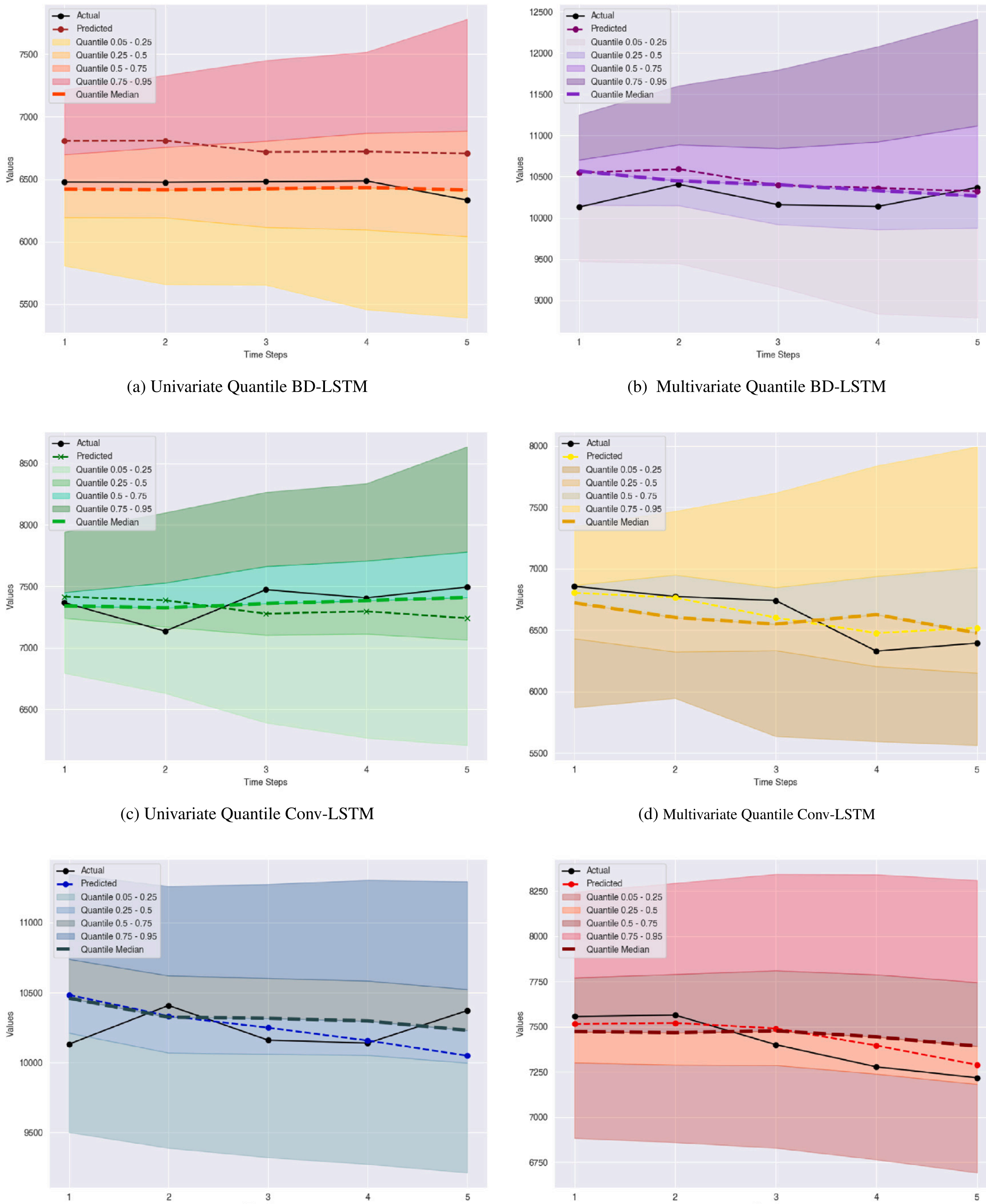
## Declaration of competing interest

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

We thank Arpit Kapoor from UNSW Sydney for earlier discussions. We thank Xizhe Chen from UNSW Sydney for earlier contributions to this paper.

## Appendix A. Prediction quantiles




(a) Univariate Quantile BD-LSTM

(b) Multivariate Quantile BD-LSTM

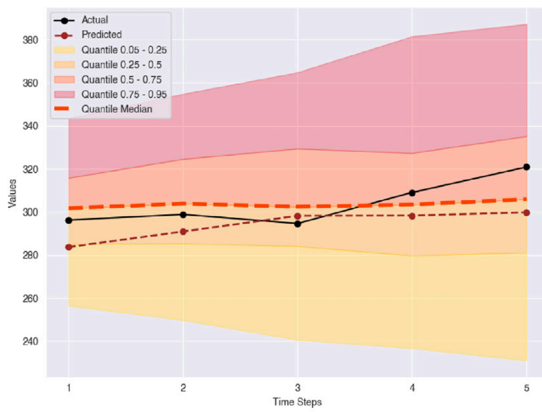
(c) Univariate Quantile Conv-LSTM

(d) Multivariate Quantile Conv-LSTM

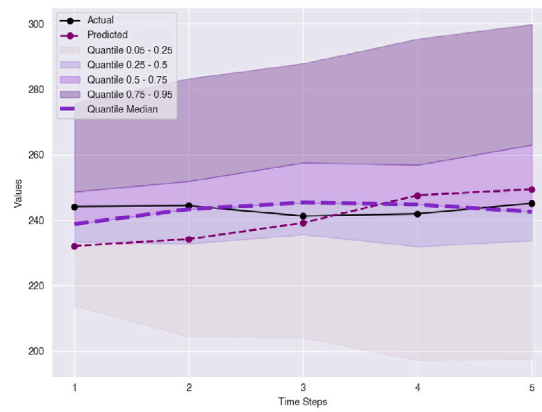
(e) Univariate Quantile ED-LSTM

(f) Multivariate Quantile ED-LSTM

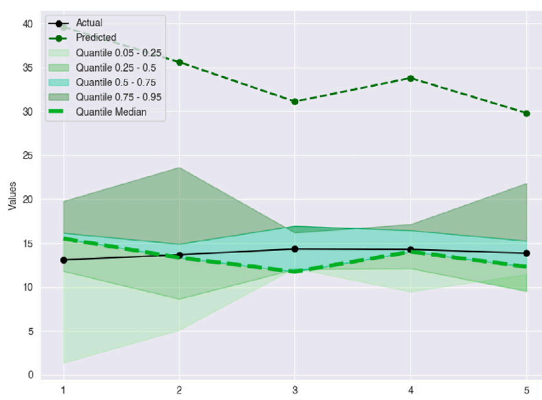
**Fig. A.9.** Prediction for the Bitcoin time series, showing quantiles for univariate and multivariate strategies for the Quantile-ED-LSTM (e.g. Quantiles 0.05–0.25) and ED-LSTM (Predicted).



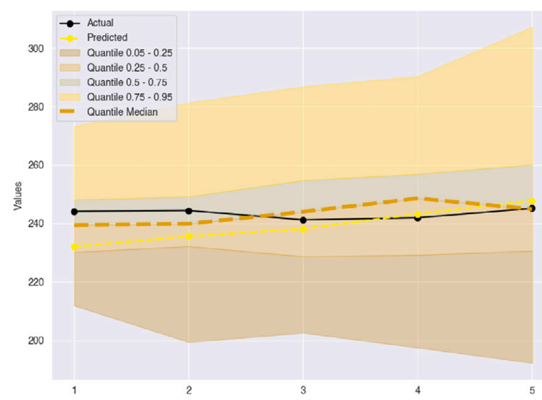

(a) Univariate Quantile BD-LSTM



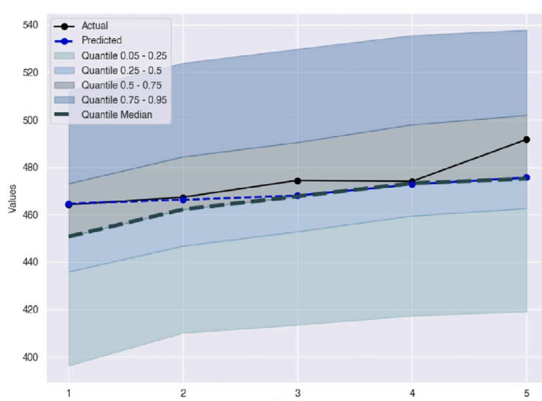

(b) Multivariate Quantile BD-LSTM



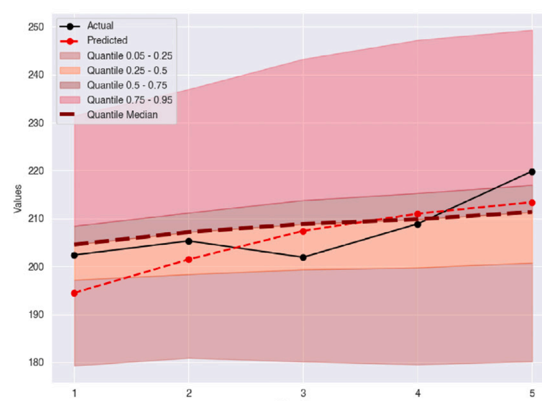

(c) Univariate Quantile Conv-LSTM




(d) Multivariate Quantile Conv-LSTM

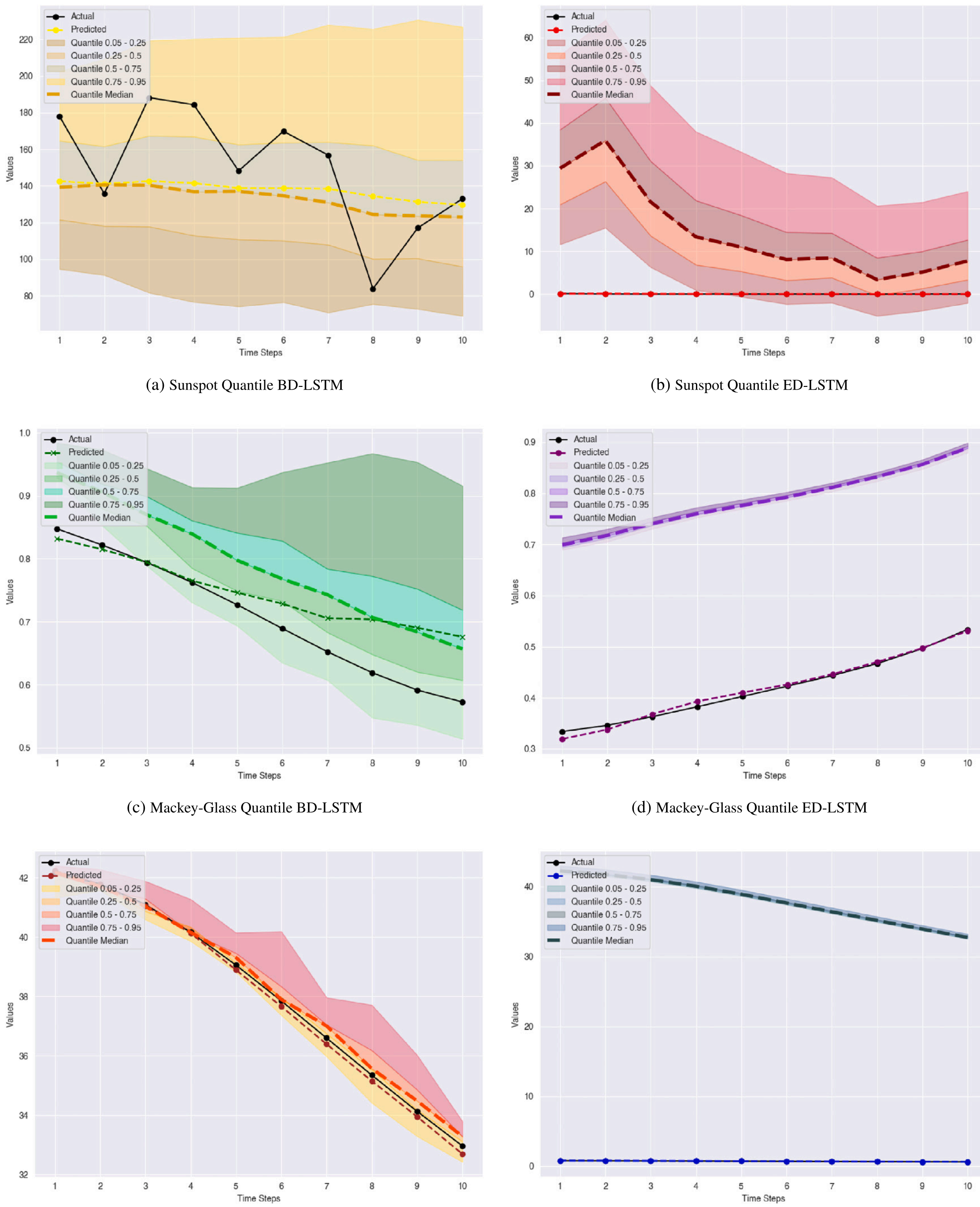



(e) Univariate Quantile ED-LSTM




(f) Multivariate Quantile ED-LSTM

**Fig. A.10.** Prediction for the ethereum time series, showing quantiles for univariate and multivariate strategies for the quantile-ED-LSTM (e.g. Quantile 0.05–0.25) and ED-LSTM (Predicted).




(a) Sunspot Quantile BD-LSTM

(b) Sunspot Quantile ED-LSTM

(c) Mackey-Glass Quantile BD-LSTM

(d) Mackey-Glass Quantile ED-LSTM

(e) Lorenz Quantile BD-LSTM

(f) Lorenz Quantile ED-LSTM

**Fig. A.11.** Predictions for univariate Quantile-ED-LSTM models (e.g. Quantile 0.05–0.25) and ED-LSTM (predicted) for Mackey-Glass, sunspot and lorenz time series.

## Data availability

We provide Python code and the data using our GitHub repository.[1]

## References

[1] R. Koenker, G. Bassett Jr, Regression quantiles, Econometrica J. Econ. Soc. (1978) 33–50, https://www.jstor.org/stable/1913643
[2] E. Waldmann, Quantile regression: a short story on how and why, Stat. Model. 18 (3–4) (2018) 203–218, https://doi.org/10.1177/1471082X18759142
[3] Y. Wei, R.D. Kehm, M. Goldberg, M.B. Terry, Applications for quantile regression in Epidemiology, Curr. Epidemiol. Rep. 6 (2019) 191–199.
[4] B. Fitzenberger, R. Koenker, J.A.F. Machado, Economic Applications of Quantile Regression, Springer Science & Business Media, 2013.
[5] B. Tillaguango, et al., Impact of oil price, economic globalization, and inflation on economic output: evidence from Latin American oil-producing countries using the quantile-on-quantile approach, Energy (2024) https://www.sciencedirect.com/science/article/pii/S0360544224015597
[6] L. Briollais, G. Durrieu, Application of quantile regression to recent genetic and-omic studies, Hum. Genet. 133 (8) (2014) 951–966.
[7] D.E. Allen, P. Gerrans, R. Powell, et al., Quantile regression: its application in investment analysis, Jassa 4 (2009) 7–12.
[8] M. Buchinsky, Changes in the US wage structure 1963-1987: application of quantile regression, Econometrica (1994) 405–458.

1 https://github.com/sydney-machine-learning/quantiledeeplearning

[9] S.J. Staffa, D.S. Kohane, D. Zurakowski, Quantile regression and its applications: a primer for anesthesiologists, Anesth. Analg. 128 (4) (2019) 820–830.
[10] G.S. Watson, Linear least squares regression, Ann. Math. Stat. (1967) 1679–1699.
[11] P. Geladi, B.R. Kowalski, Partial least-squares regression: a tutorial, Anal. Chim. Acta 185 (1986) 1–17.
[12] K. Vaysse, P. Lagacherie, Using quantile regression forest to estimate uncertainty of digital soil mapping products, Geoderma 291 (2017) 55–64.
[13] N. Dogulu, P. López López, D.P. Solomatine, A.H. Weerts, D.L. Shrestha, Estimation of predictive hydrologic uncertainty using the quantile regression and UNEEC methods and their comparison on contrasting catchments, Hydrol. Earth Syst. Sci. 19 (7) (2015) 3181–3201.
[14] M.K. Ochi, On prediction of extreme values, J. Ship Res. 17 (1) (1973) 29–37.
[15] R.P. Ribeiro, N. Moniz, Imbalanced regression and extreme value prediction, Mach. Learn. 109 (2020) 1803–1835.
[16] B. Wang, X. Zhou, Climate variation and prediction of rapid intensification in tropical cyclones in the western North Pacific, Meteorol. Atmos. Phys. 99 (1) (2008) 1–16.
[17] L. Makkonen, Problems in the extreme value analysis, Struct. Saf. 30 (5) (2008) 405–419.
[18] J.H. Lee, H. Kim, H.J. Park, et al., Temporal prediction modeling for rainfall-induced shallow landslide hazards using extreme value distribution, Landslides 18 (2021) 321–338.
[19] H. Xing, S. Junyi, H. Jin, The casualty prediction of earthquake disaster based on extreme learning machine method, Nat. Hazards 102 (3) (2020) 873–886.
[20] X. Zhao, C. Scarrott, L. Oxley, et al., Extreme Value Modelling for Forecasting Market Crisis Impacts, Routledge, 2014.
[21] C. Wang, L. Zhang, G. Tao, Quantifying the influence of corrosion defects on the failure prediction of natural gas pipelines using generalized extreme value distribution (GEVD) model and copula function with a case study, Emerg. Manag. Sci. Technol. 4 (1) (2024).
[22] P. Cumperayot, R. Kouwenberg, Early warning systems for currency crises: a multivariate extreme value approach, J. Int. Money Finance 36 (2013) 151–171.
[23] T.B. Messervey, D.M. Frangopol, S. Casciati, Application of the statistics of extremes to the reliability assessment and performance prediction of monitored highway bridges, in: Structures and Infrastructure Systems, Routledge, 2019, pp. 287–299.
[24] Y. Cai, D.E. Reeve, Extreme value prediction via a quantile function model, Coast. Eng. 77 (2013).
[25] A.J. McNeil, Extreme value theory for risk managers, Departement Mathematik ETH Zentrum 12 (5) (1999) 217–237.
[26] J. Pickands Iii, Statistical inference using extreme order statistics, Ann. Stat. (1975) 119–131.
[27] J. Velthoen, C. Dombry, J.J. Cai, et al., Gradient boosting for extreme quantile regression, Extremes 77 (2023) 639–667.
[28] B. Lim, S. Zohren, Time-series forecasting with deep learning: a survey, Philos. Trans. R. Soc. A 379 (2194) (2021) 20200209.
[29] J.F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, A. Troncoso, Deep learning for time series forecasting: a survey, Big Data 9 (1) (2021) 3–21.
[30] W. Fang, Q. Xue, L. Shen, et al., Survey on the application of deep learning in extreme weather prediction, Atmosphere 12 (6) (2021) 661.
[31] Z. Chen, H. Yu, Y. Geng, et al., Evanet: an extreme value attention network for long-term air quality prediction, in: 2020 IEEE International Conference on Big Data (Big Data), IEEE, 2020, pp. 4545–4552.
[32] S. Gope, S. Sarkar, P. Mitra, et al., Early prediction of extreme rainfall events: a deep learning approach, in: Advances in Data Mining. Applications and Theoretical Aspects: 16th Industrial Conference, ICDM 2016, New York, NY, USA, July 13-17, 2016. Proceedings 16, Springer International Publishing, 2016, pp. 154–167.
[33] E. Chen, M.S. Andersen, R. Chandra, Deep learning framework with Bayesian data imputation for modelling and forecasting groundwater levels, Environ. Model. Softw. 178 (2024) 106072.
[34] G. Papacharalampous, et al., Uncertainty estimation in spatial interpolation of satellite precipitation with ensemble learning (2024) https://arxiv.org/abs/2403.10567
[35] H. Tyralis, G. Papacharalampous, A review of predictive uncertainty estimation with machine learning, Artif. Intell. Rev. 57 (4) (2024) 94, https://link.springer.com/article/10.1007/s10462-023-10698-8#Abs1
[36] Y. Jia, J.H. Jeong, Deep learning for quantile regression under right censoring: DeepQuantReg, Comput. Stat. Data Anal. 165 (2022) 107323.
[37] M.J. Van Strien, A. Grêt-Regamey, A global time series of traffic volumes on extra-urban roads, Sci. Data 11 (1) (2024) 470, https://www.nature.com/articles/s41597-024-03287-z
[38] S. Zhu, M. Zhang, C. Wang, et al., A probabilistic runoff prediction model based on improved long short-term memory and interval correction, J. Hydrol. Eng. 29 (4) (2024) 04024018.
[39] J. Hu, J. Tang, Z. Liu, A novel time series probabilistic prediction approach based on the monotone quantile regression neural network, Inf. Sci. 654 (2024) 119844, https://www.sciencedirect.com/science/article/pii/S0020025523014299
[40] W. Zhang, H. Quan, D. Srinivasan, An improved quantile regression neural network for probabilistic load forecasting, IEEE Trans. Smart Grid 10 (4) (2018) 4425–4434.
[41] J. Hu, J. Tang, Z. Liu, A novel time series probabilistic prediction approach based on the monotone quantile regression neural network, Inf. Sci. 119844 (2023).
[42] J. Tang, et al., Neural networks for partially linear quantile regression, J. Bus. Econ. Stat. (2023) https://www.tandfonline.com/doi/abs/10.1080/07350015.2023.2208183
[43] M. et al. A conceptual model of investment-risk prediction in the stock market using extreme value theory with machine learning: a semisystematic literature review, Risks 11 (3) (2023) 60 , https://www.mdpi.com/2227-9091/11/3/60
[44] A.-A. Ibn Musah, et al., The asymptotic decision scenarios of an emerging stock exchange market: extreme value theory and artificial neural network, Risks 6 (4) (2018) 132, https://www.mdpi.com/2227-9091/6/4/132
[45] M. Melina, et al., Modeling of machine learning-based extreme value theory in stock investment risk prediction: a systematic literature review, Big Data (2024) https://www.liebertpub.com/doi/abs/10.1089/big.2023.0004
[46] R. Chandra, S. Goyal, R. Gupta, Evaluation of deep learning models for multi-step ahead time series prediction, IEEE Access 9 (2021) 83105–83123.
[47] H. Zhu, H. Xia, Y. Guo, C. Peng, The heterogeneous effects of urbanization and income inequality on CO2 emissions in BRICS economies: evidence from panel quantile regression, Environ. Sci. Pollut. Res. 25 (2018) 17176–17193.
[48] R. Koenker, Quantile regression for longitudinal data, J. Multivar. Anal. 91 (1) (2004) 74–89.
[49] Z. Cai, Regression quantiles for time series, Econom. Theory 18 (1) (2002) 169–192, https://www.cambridge.org/core/journals/econometric-theory/article/abs/regression-quantiles-for-time-series/485A99DE06615C32AA41D737D8E29A77
[50] Z. Xiao, Time series quantile regressions, in: Handbook of Statistics, vol. 30, Elsevier, 2012, pp. 213–257, https://www.sciencedirect.com/science/article/abs/pii/B9780444538581000090
[51] R. Koenker, Quantile regression: 40 years on, Annu. Rev. Econ. 9 (2017) 155–176, https://www.annualreviews.org/content/journals/10.1146/annurev-economics-063016-103651#right-ref-B44
[52] T.G. Bali, The generalized extreme value distribution, Econ. Lett. 79 (3) (2003) 423–427.
[53] E. Castillo, A.S. Hadi, Fitting the generalized Pareto distribution to data, J. Am. Stat. Assoc. 92 (440) (1997) 1609–1620.
[54] A. Ferreira, L. De Haan, On the block maxima method in extreme value theory: PWM estimators, Ann. Stat. (2015) 276–298, http://www.jstor.org/stable/43556515
[55] M. Fréchet, Sur la loi de probabilité de l'écart maximum, Ann. de la Soc. Polonaise de Math. (1927).
[56] R.A. Fisher, L.H.C. Tippett, Limiting forms of the frequency distribution of the largest and smallest member of a sample, Proc. Camb. Philos. Soc. 24 (1928) 180–190.
[57] V.F. Pisarenko, A. Sornette, D. Sornette, et al., Characterization of the tail of the distribution of earthquake magnitudes by combining the GEV and GPD descriptions of extreme value theory, Pure Appl. Geophys. 171 (2014) 1599–1624.
[58] A. Bücher, C. Zhou, A horse race between the block maxima method and the peak–over–threshold approach, Statistical Science 36 (3) (2021) 360–378. https://projecteuclid.org/journals/statistical-science/volume-36/issue-3/A-Horse-Race-between-the-Block-Maxima-Method-and-the/10.1214/20-STS795.short.
[59] T.H. Soukissian, C. Tsalis, The effect of the generalized extreme value distribution parameter estimation methods in extreme wind speed prediction, Nat. Hazards 78 (2015) 1777–1809.
[60] J. Schaumburg, Predicting extreme value at risk: nonparametric quantile regression with refinements from extreme value theory, Comput. Stat. Data Anal. 56 (12) (2012) 4081–4096.
[61] F.F. Do Nascimento, D. Gamerman, H.F. Lopes, A semiparametric Bayesian approach to extreme value estimation, Stat. Comput. 22 (2012) 661–675.
[62] H.B. Sandya, P.H. Kumar, S.B. Patil, Feature extraction, classification and forecasting of time series signal using fuzzy and GARCH techniques, in: Proceedings of the National Conference on Challenges in Research and Technology in the Coming Decades (CRT), 2013, pp. 1–7.
[63] L.C. Chang, P.A. Chen, F.J. Chang, Reinforced two-step-ahead weight adjustment technique for online training of recurrent neural networks, IEEE Trans. Neural Netw. Learn. Syst. 23 (8) (2012) 1269–1278, https://doi.org/10.1109/TNNLS.2012.2200695
[64] R.W.V.S. Khedkar, R. Chandra, Evaluation of deep learning models for Australian climate extremes: prediction of streamflow and floods, arXiv:2407.15882 (2024) https://arxiv.org/abs/2407.15882
[65] J. Lampinen, A. Vehtari, Bayesian approach for neural networks—review and case studies, Neural Networks 14 (3) (2001) 257–274.
[66] I. Kononenko, Bayesian neural networks, Biol. Cybern. 61 (5) (1989) 361–370.
[67] H. Jang, J. Lee, An empirical study on modeling and prediction of Bitcoin prices with Bayesian neural networks based on blockchain information, IEEE Access 6 (2017) 5427–5437.
[68] R. Chandra, Y. He, Bayesian neural networks for stock price forecasting before and during COVID-19 pandemic, PLoS One 16 (7) (2021) e0253217, https://doi.org/10.1371/journal.pone.0253217
[69] Y. Wang, G. Andreeva, B. Martin-Barragan, Machine learning approaches to forecasting cryptocurrency volatility: considering internal and external determinants, Int. Rev. Financ. Anal. 90 (2023) 102914.
[70] J. Wu, X. Zhang, F. Huang, H. Zhou, R. Chandra, Review of deep learning models for crypto price prediction: implementation and evaluation, arXiv (2024) https://arxiv.org/abs/2405.11431
[71] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (2) (1990) 179–211.
[72] G. Van Houdt, C. Mosquera, G. Nápoles, A review on the long short-term memory model, Artif. Intell. Rev. 53 (Dec 2020) https://doi.org/10.1007/s10462-020-09838-1
[73] S. Hochreiter, The vanishing gradient problem during learning recurrent neural nets and problem solutions, Int. J. Uncertain. Fuzziness Knowl. Based Syst. 6 (2) (1998) 107–116.
[74] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Networks 18 (5–6) (2005) 602–610.

[75] Y. Yu, X. Si, C. Hu, J. Zhang, A review of recurrent neural networks: LSTM cells and network architectures, Neural Comput. 31 (7) (2019) 1235–1270.
[76] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems, vol. 27, 2014.
[77] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-C. Woo, Convolutional LSTM network: a machine learning approach for precipitation nowcasting, in: Advances in Neural Information Processing Systems, vol. 28, 2015.
[78] L. Alzubaidi, J. Zhang, A.J. Humaidi, et al., Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, J. Big Data 8 (2021) 1–74.
[79] X. Hinaut, N. Trouvain, Mackey-Glass timeseries dataset (2021) https://reservoirpy.readthedocs.io/en/latest/api/generated/reservoirpy.datasets.mackey_glass.html
[80] M.C. Mackey, L. Glass, Oscillation and chaos in physiological control systems, Science 197 (4300) (1977) 287–289, https://doi.org/10.1126/science.267326
[81] E.N. Lorenz, The statistical prediction of solutions of dynamic equations, Proc. Int. Symp. Numer. Weather Predict. (1960) 628–635, https://web.archive.org/web/20190523190103/http://eaps4.mit.edu/research/Lorenz/The_Statistical_Prediction_of_Solutions_1962.pdf
[82] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980 (2017) https://arxiv.org/abs/1412.6980
[83] A. Greaves, B. Au, Using the bitcoin transaction graph to predict the price of bitcoin, 2015. https://api.semanticscholar.org/CorpusID:18038866
[84] M.S. Thiese, B. Ronna, U. Ott, P value interpretations and considerations, J. Thorac. Dis. 8 (9) (2016) E928–E931, https://doi.org/10.21037/jtd.2016.08.16
[85] H. Hsu, P.A. Lachenbruch, Paired t test, in: Wiley Statsref: Statistics Reference Online, Wiley, 2014, https://doi.org/10.1002/9781118445112.stat06558.pub2
[86] R.F. Woolson, Wilcoxon signed-rank test, in: Wiley Encyclopedia of Clinical Trials, Wiley, 2007, pp. 1–3, https://doi.org/10.1002/9780471462422.eoct979
[87] L. St, S. Wold, Analysis of variance (ANOVA), Chemom. Intell. Lab. Syst. 6 (4) (1989) 259–272, https://doi.org/10.1016/0169-7439(89)80195-6
[88] D.W. Zimmerman, B.D. Zumbo, Relative power of the Wilcoxon test, the Friedman test, and repeated-measures ANOVA on ranks, J. Exp. Educ. 62 (1) (1993) 75–86, https://doi.org/10.1080/00220973.1993.9943838
[89] T. Gneiting, M. Katzfuss, Probabilistic forecasting, Annu. Rev. Stat. Its Appl. 1 (2014) 125–151, https://doi.org/10.1146/annurev-statistics-062713-085831
[90] T. Gneiting, A.E. Raftery, Strictly proper scoring rules, prediction, and estimation, J. Am. Stat. Assoc. 102 (477) (2007) 359–378, https://doi.org/10.1198/016214506000001437
[91] R.Y. Chou, Forecasting financial volatilities with extreme values: the conditional autoregressive range (CARR) model, J. Money Credit Bank. (2005) 561–582.
[92] X. Zhao, C. Scarrott, L. Oxley, M. Reale, Extreme value modelling for forecasting market crisis impacts, in: The Global Financial Crisis, Routledge, 2014, pp. 61–70.
[93] W.J. Dixon, Analysis of extreme values, Ann. Math. Stat. 21 (4) (1950) 488–506.
[94] B. Merz, A.H. Thieken, Flood risk curves and uncertainty bounds, Nat. Hazards 51 (2009) 437–458.
[95] L. Yang, A. Shami, On hyperparameter optimization of machine learning algorithms: theory and practice, Neurocomputing 415 (2020) 295–316.
[96] R. Muthukrishnan, R. Rohini, Lasso: a feature selection technique in predictive modeling for machine learning, in: 2016 IEEE International Conference on Advances in Computer Applications (ICACA), IEEE, 2016, pp. 18–20.
[97] W.N. van Wieringen, Lecture notes on ridge regression, arxiv Preprint arXiv:1509.09169 (Sep 2015).
[98] S. Wager, S. Wang, P.S. Liang, Dropout training as adaptive regularization, Adv. Neural Inf. Process. Syst. 26 (2013).
[99] T. Boiński, P. Czarnul, Optimization of data assignment for parallel processing in a hybrid heterogeneous environment using integer linear programming, Comput. J. 65 (6) (2022) 1412–1433, https://doi.org/10.1093/comjnl/bxaa187
[100] Y. Park, J. Kim, H. Kim, K. Shin, S. Kim, Learning quantile functions without quantile crossing for distribution-free time series forecasting, in: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), Proceedings of Machine Learning Research vol. 151 of PMLR, 2022, pp. 2432–2448.
[101] I.S. Al-Tameemi, M.R. Feizi-Derakhshi, S. Pashazadeh, M. Asadpour, Multi-model fusion framework using deep learning for visual-textual sentiment classification, Comput. Mater. Contin. 76 (2) (2023) 2145–2177.
[102] W. Zhang, Y. Deng, B. Liu, S.J. Pan, L. Bing, Sentiment analysis in the era of large language models: a reality check, arxiv Preprint arXiv:2305.15005 (May 2023).
[103] R.M. Neal, Bayesian Learning for Neural Networks, vol. 118, Springer New York, 1996.
[104] Y. Yoon, G. Swales Jr, T.M. Margavio, A comparison of discriminant analysis versus artificial neural networks, J. Oper. Res. Soc. 44 (1) (1993) 51–60, https://doi.org/10.1057/jors.1993.6
[105] D.J.C. MacKay, A practical Bayesian framework for backpropagation networks, Neural Comput. 4 (3) (1992) 448–472, https://doi.org/10.1162/neco.1992.4.3.448
[106] R. Zhang, C. Li, J. Zhang, C. Chen, A.G. Wilson, Cyclical stochastic gradient mcmc for bayesian deep learning, arXiv preprint arXiv:1902.03932 (Feb 2019).
[107] H. Attias, Inferring parameters and structure of latent variable models by variational Bayes, arxiv Preprint arXiv:1301.6676 (Jan 2013).
[108] B. Chaput, J.C. Girard, M. Henry, Frequentist approach: modelling and simulation in statistics and probability teaching, in: Teaching Statistics in School Mathematics—Challenges for Teaching and Teacher Education: A Joint ICMI/IASE Study: the 18th ICMI Study, 2011, pp. 85–95.
[109] E. Afrifa-Yamoah, U.A. Mueller, S.M. Taylor, A.J. Fisher, Missing data imputation of high-resolution temporal climate time series data, Meteorol. Appl. 27 (1) (2020) e1873.