Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 5, 2026

# A HYBRID STATE-SPACE AND ATTENTION FRAMEWORK FOR CRYPTOCURRENCY TIME-SERIES FORECASTING USING CRYPTOMAMBA AND ATTENTIONMLP

**Abdul Sattar Chan[1], Zainab Umair Kamangar[2], Umair Ayaz Kamangar[*3], Junaid Ahmed[4], Mumtaz Ali[5]**

[1,*3,4,5]*Department of Computer Systems Engineering, Sukkur IBA University, Pakistan*
[2]*Department of Computer Science, Sukkur IBA University, Pakistan*

[1]abdul.sattar@iba-suk.edu.pk, [2]zainabumair.phdcss22@iba-suk.edu.pk, [*3]umair.ayaz@iba-suk.edu.pk, [4]j.bhatti@iba-suk.edu.pk, [5]mumtaz.ali@iba-suk.edu.pk

DOI: https://doi.org/10.5281/zenodo.20389770

**Keywords**

**Article History**
*Received: 28 March 2026*
*Accepted: 07 May 2026*
*Published: 26 May 2026*


**Corresponding Author: ***
**Umair Ayaz Kamangar**

**Abstract**

*Cryptocurrency markets are unpredictable, highly non-stationary, and subject to intricate time-series patterns, which significantly complicate the forecasting task. Historically, different variants of RNN such as LSTM and BiLSTM were heavily utilized for time-series prediction, yet they often do not effectively capture long-range dependencies, while possessing high computational costs. This research suggests a comparable framework utilizing the state-space sequence modeling architecture CryptoMamba to tackle these problems. We benchmark the CryptoMamba's performance against classic RNN models and also propose a new version of CryptoMamba combined with Attention-based Multi-Layer Perceptron (AttentionMLP) to further boost the prediction accuracy. Experiments were performed over a dataset of cryptocurrency with identical features and evaluation criteria. Based on the evaluation, it can be observed that the model CryptoMamba + AttentionMLP provides the best prediction with lowest RMSE, MAE and MAPE, while outperforming LSTM, BiLSTM and GRU.*

## 1. INTRODUCTION

The merger among technology and financial sectors has greatly affected the means markets operate and the way investments are carried out. The invention of online trading facilities, algorithms to trade at extremely high frequency and increasing use of Artificial Intelligence are contributing to the digital evolution of the markets and expanding new dimensions for opportunities in stock market for investors and analysts for data processing, prediction and risk mitigation. Against this background, the research and application of forecasting models based on AI technology have attracted a large amount of attention [1].

Stock market is an exchange where lots of investors make transactions of sell and buy of shares in different companies which in essence is portion of a business's worth. The primary purpose of the stock market is to leverage the changes in asset values. Investment in stock market for most individuals is a vital part of their financial planning, for it is the arena where wealth can grow over time; a passive investment which offers greater return than that achieved through savings or any other method of investment, while on the other hand it is also subject to much uncertainty and risk. Forecasting the market is an attractive but nevertheless, daunting prospect. All investors aspire to develop

a successful method of anticipating the movement of market and help him invest more efficiently.

Historically, various methods and techniques have been deployed to forecast stock market movements; from the fundamental analysis of financial data to technical analysis of past stock prices. In spite of all these attempts at forecasting market direction, accurate prediction of market behavior has remained an elusive open issue.

Recently, new avenues have opened up for the application of Deep learning (DL) techniques for stock market prediction following technological advancements and the increased availability of data. DL models, belonging to the field of ML, focus on developing algorithms that can learn from data, detect subtle correlations, and forecast data based on the patterns they identify [2].

In this paper, we seek to ascertain how ML algorithms can assist in stock market forecasting by utilizing prediction models to characterize the complex dynamics of market behavior. Initial endeavors at market prediction employed statistical methods; however, DL models have proven to be highly effective in modeling complex patterns and, recently, the research on the scalability and the limited parallelization capability of recurrent neural networks like LSTM, BiLSTM, and GRU has become popular and leading to the introduction of linear complexity state-space models like Mamba.

This project aims to explore the potential of enhancing the cryptocurrency forecast performance through the combined use of CryptoMamba with a powerful MLP based prediction head.

The main contributions of this work are as follows:

- To evaluate classical RNN-based models (LSTM, BiLSTM, GRU) for cryptocurrency prediction.
- To implement CryptoMamba as a modern sequence modeling backbone.
- To design and integrate multiple prediction heads, including MLP, Gated MLP, FourierMLP, TCN, and AttentionMLP.
- To perform a fair comparative analysis using identical datasets, features, and evaluation metrics.

## 2. Literature Review

Recent applications of deep learning and machine learning models to financial markets have gained significant attention, especially for cryptocurrency price forecasting.

Awoke et al. (2020) evaluated the use of deep learning models for Bitcoin price forecasting with recurrent structures including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures. The study found that both models successfully captured the temporal dependence of Bitcoin price time series. GRU was faster and less computationally intensive than LSTM.[3]

Chang et al. (2024) assessed the capabilities of deep learning and state-of-the-art machine learning methods for predicting trends in economy and stock market. Recurrent models such as LSTM, GRU are used together with classical statistical models (ARIMA, Prophet) and ensemble methods to predict the financial time-series data. In terms of RMSE and MAE values, GRU model shows superior performance over all classical methods.[4]

Han et al. (2025) introduced MFB as a generalized multimodal fusion framework with both time-lagged sentiment and technical indicators for predicting Bitcoin prices. BiLSTM and BiGRU architectures combined with attention mechanisms and feature selection methods can effectively capture the delayed effects from the textual sentiment data onto the price prediction. MFB demonstrated significant improvement compared with traditional deep learning and unimodal methods in forecasting accuracy and prediction error.[5]

Xie et al. (2022) proposed MARINA, an MLP-attention based architecture designed for multivariate time-series forecasting and anomaly detection. The model decomposes spatio-temporal learning into a temporal module, which is based on MLP, and a spatial module, which is based on self-attention, without using any recurrent or transformer-based architectures. MARINA accurately captures long-term temporal dependencies and inter-variable correlations.

MARINA achieved state-of-the-art forecasting accuracy with significantly lower computational costs than state-of-the-art methods on various benchmarks. Hence, MARINA shows its potential for large-scale cryptocurrency price time series forecasting.[6]

Sepehri et al. (2025) introduced CryptoMamba, a State Space Model (SSM) based on the Mamba architecture designed for highly volatile cryptocurrency price forecasting. Leveraging selective state-space dynamics with linear computational complexity allows capturing long-range temporal dependencies. Combining hierarchical Mamba layers and lightweight MLP structures enables CryptoMamba to extract multi-scale temporal features. Comparing CryptoMamba with existing methods such as LSTM, Bi-LSTM, GRU, iTransformer and S-Mamba using real-world Bitcoin data on RMSE, MAE and MAPE metrics demonstrated significant improvements over all other methods while using a fraction of parameters. Trading simulations indicate the effectiveness of state-space models for cryptocurrency forecasting due to high profitability and low risk.[7]

Bai et al. (2018) presented a Temporal Convolutional Network (TCN) for sequence modeling, including time-series prediction. TCN employs causality and dilatation coupled with residual connections which enable capturing of long-range temporal dependencies effectively. Furthermore, TCN also supports parallelism, making it faster and more memory efficient compared to RNNs. Experimental results demonstrated that TCN is superior over RNNs such as LSTM and GRU for tasks requiring long-term dependencies. Thus, TCN can be a good candidate for time series forecasting applications like Bitcoin price prediction.[8]

Teixeira and Barbosa (2025) provide a comprehensive comparison of machine learning and deep learning techniques in financial time series forecasting applied to stock price prediction. They tested classical methods (XGBoost), recurrent models (RNN, LSTM, GRU) and hybrid models using multiple technical indicators, macro-economic variables, and market index as input features to the models. Their experimental results revealed that both GRU and XGBoost are able to model the highly non-linear dynamics of the financial data. Their work shows that combining different models would achieve better performance. Their study focusing on stock prices could be very useful in cryptocurrency price forecasting since these two market types share many common characteristics such as volatility and non-stationarity which justifies the investigation of hybrid, MLP, and state-space based architectures for long-horizon forecasting.[9]

Kim et al. (2022) suggested a deep learning framework that utilizes on-chain information such as volume, miner behaviour and network participation for cryptocurrency price forecasting. In order to effectively handle the non-stationarity of the Bitcoin price series and avoid the sudden changes caused by regime shifts, they proposed a change point detection approach for segmentation and normalization of the data and a Self-Attention-based Multiple LSTM (SAM-LSTM) architecture for price prediction. The SAM-LSTM model comprises multiple LSTM units for group processing of the features which is followed by attention mechanisms and aggregation through a MLP network. Experiments conducted on real Bitcoin price data demonstrated significant improvements over the traditional LSTM based methods in terms of MAE, RMSE, MSE and MAPE. This study shows the significance of incorporating blockchain features along with a deep neural architecture in a structured way for predicting cryptocurrency prices.[10]

Mousa et al. (2025) proposed a Bitcoin price prediction framework that integrates hashrate-based on-chain data, wavelet decomposition, and deep stacking learning architecture for efficient handling of cryptocurrency volatility and non-stationarity. While traditional models rely on historical price series, this method takes advantage of hashrate, a key blockchain characteristic, which indirectly represents miner interest and network stability. Hashrate signals are transformed using wavelet decomposition, splitting them into different frequency components to efficiently model both long-term

and short-term temporal patterns. The resulting decomposed components are then fed into a deep stacking learning framework of hierarchical base learners. Experimental results obtained from real Bitcoin market data demonstrate the superiority of the proposed hybrid model over classical machine learning and standalone deep learning methods with improved RMSE, MAE, and MAPE values. This study highlights the benefits of leveraging on-chain data and a multi-resolution approach for accurate cryptocurrency price forecasting.[11]

Liu et al. (2021) developed a Gated Multilayer Perceptron (gMLP) model as a replacement for Transformer models in sequence modeling. GMLP substitutes self-attention with a Spatial Gating Unit (SGU) that efficiently handles cross-token dependencies through linear projections and gating mechanisms. Their results in language and vision demonstrate comparable performance to Transformers with a simpler architecture and lower computational cost, highlighting its potential for time series applications [12].

Ukwuoma et al. (2025) presented an attention-gated MLP (agMLP) model for hydrogen production prediction. By integrating gated MLP architecture with attention, complex nonlinear dependencies were captured and model interpretability was improved. Experimental results indicate agMLP surpasses traditional methods in predictive accuracy. Moreover, the model's robustness was confirmed through explainable AI methods like SHAP and LIME, showcasing its reliability in sustainable energy prediction tasks [13].

Li et al. (2023) introduced a Temporal Convolutional Network (TCN)-based hybrid forecasting framework for short-term utility-scale photovoltaic (PV) power prediction. This framework combines a physics-based trend forecasting model with a data-driven fluctuation forecasting model, using multiple TCNs to effectively capture both long-term trends and short-term intra-hour variability due to cloud movements. An automated scenario-based detector site selection method was proposed to leverage spatio-temporal correlations across neighboring PV sites. Extensive experiments on real PV data confirmed that the TCN-based hybrid approach significantly improves forecasting accuracy by 20-30% over state-of-the-art methods [14].

## 3. Methodology

This approach adopts a two-stage experimental setting. In the first stage, standard RNN based sequence models (LSTM, BiLSTM, and GRU) are trained and tested as baseline methods. In the second stage, CryptoMamba acts as the backbone for sequence modeling, along with the added multiple prediction heads are applied to see how they perform. All the models are trained by using the same data, features, optimizer, and evaluation measures in order to establish fair comparisons.

### 3.1. Dataset

The first data set contains historical information on bitcoin from a period where the data is available and can be collected from Yahoo Finance [15]. For every interval of time in the data set, there is a value for each of the following five market characteristics, which are frequently used in predicting time-series finance: Open Price-The price at the opening of trading for that time interval. High Price-The maximum value reached at any point during that time interval. Low Price-The minimum value reached at any point during that time interval. Close Price-The price at which trading ends for that time interval and is used as the target value. Trading Volume-The total quantity of bitcoin traded during that time interval; it is a measure of trading liquidity and participant behavior.

### 3.2 Performance Measures

Prior to discussing the deep learning models and data preparation it was necessary to select a series of performance metrics that would be used to assess those DL models. The choice of performance metrics is of crucial importance when evaluating a model because they give a clear measure of the performance of a model; and it is, because it is crucial that those values accurately reflect the performance of different models, that it used the metrics that would take into consideration both the magnitude and the

direction of errors of predictions. Mean Absolute Error (MAE) Root and Mean Squared Error (RMSE) both were chosen by reason of simplicity of understanding and also due to the clear measure they gave of prediction accurately.

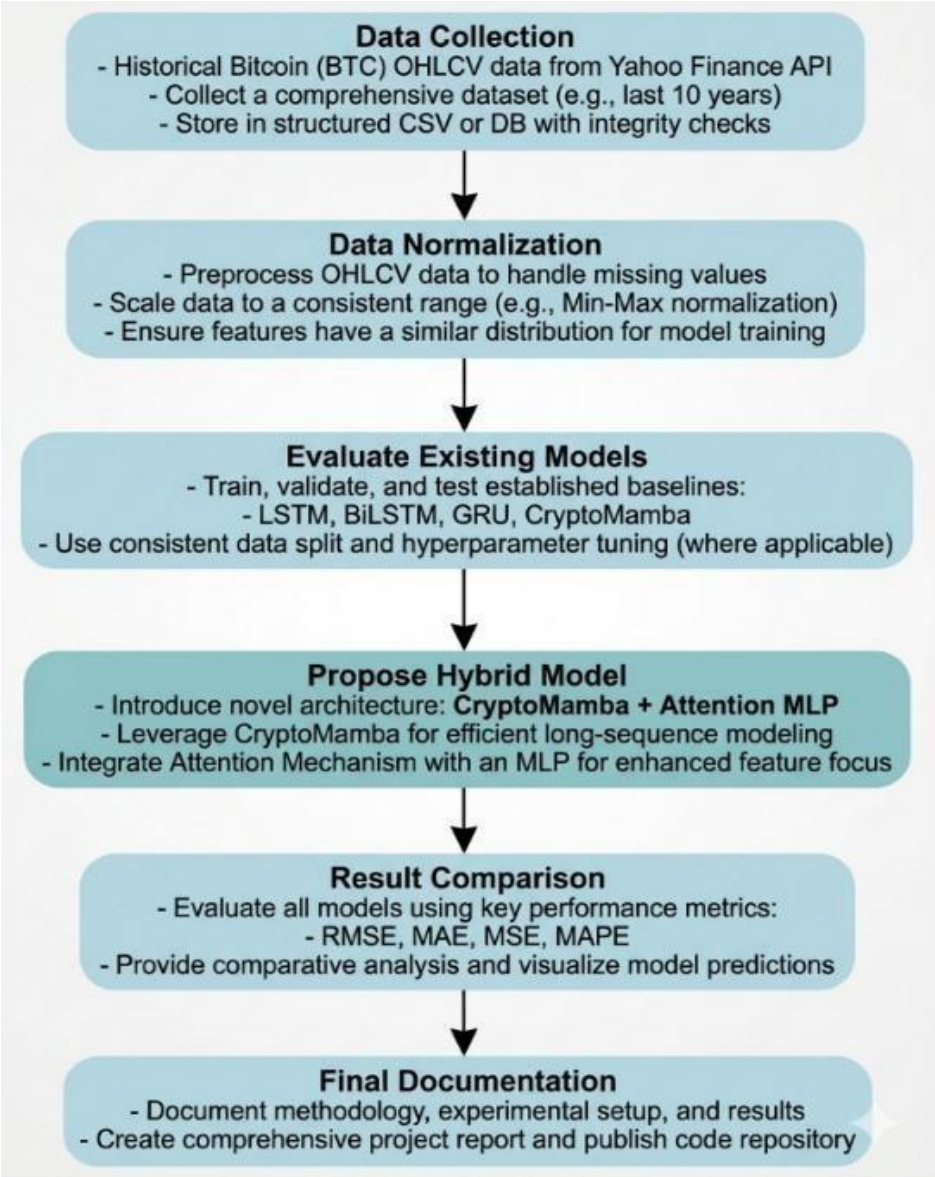



**Figure 1: Methodology**

Hence it was decided to make use of a series of 5 metrics; namely, MAE, MSE, RMSE, Mean Absolute Percentage Error (MAPE) and the Coefficient of Determination (R2). MAE-mean of the absolute values of the difference between forecasts and actuals. The MAE is a straightforward indicator which represents a clear measure of the magnitude of the forecast errors. Indeed, as seen below, MAE is computed as the average of the absolute differences between the forecast and the actual value (Eq. (1)), where lower values correspond to best forecasts [16].

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \longrightarrow 1$$

where xi denotes the actual values, yi the predicted values, and n is the total number of observations.
MSE Eq. (2)), is one of the frequently used metrics for evaluating the performance of regression models. It represents the average of the squares of the differences between the predicted and the actual value. It favors the larger errors more than the smaller errors as it squares each of the errors prior to averaging, hence more susceptible to outliers [17]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2 \longrightarrow 2$$

The RMSE also been employed. As RMSE is a derivation of MSE, it represents the square root of MSE, hence measure the typical difference between actual values and prediction errors. RMSE is common as its interpretability and gives indication how far ahead prediction can be made (Eq. (3)) The scale of RMSE is as the actual values [19]:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \longrightarrow 3$$

MAPE value which can be said to be very handy in calculating the average % error between the forecasts and actual values. In (Eq. (4)) MAPE is particularly useful where it is desirable to know the relative accuracy of the forecasts against the actual values without regard to the magnitude of the data. Likewise, if predicting stock prices, it would serve no purpose to have the absolute value of the value of any of these metrics for different stocks, or for different portions of the same stock prices; the best possible useful metric would be a percentage value, and MAPE is one of such useful values.

$$MAPE = 100\frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - y_i}{x_i}\right| \longrightarrow 4$$

Finally, $R^2$ that is a statistical measure, helps to determine how well the data are fitted to the model (Eq. (5)). A value of 1 means a perfect fit while a value less than 1 represents a poor fit. Negative values and zero also represent a poor fit [18].

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(x_i - y_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \longrightarrow 5$$

where x¯-> actual values mean.

### 3.3 Data Processing

Data processing is one of the first step to the creation of dependable models and more especially when you are comparing. Indeed, the models should always be given clean data so that when you are testing them you only get to see the model. It also ensures that data is correctly formatted, clean, and can also make model better performing and stable. In this study, the different data types obtained and collected were processed into one database which was used as the training data for machine learning models as explained above.

#### 3.3.1 Normalize

The data was normalized, by MinMaxScaler function (available in sklearn.preprocessing library, [19]). This rescales the data so it will lie between two preset numbers. In this case the data was scaled between 0 and 1. This process is essential for boosting efficiency of machine learning algorithms and for ensuring each feature has an equal input. Normalization is formulated as:

Xnorm = X—Xmin / Xmax—Xmin

Additionally, the data is organized into fixed-length sequential windows to transform the raw time series into supervised learning samples suitable for deep sequence modeling.

#### 3.3.2 Temporal Data Splitting

To ensure a realistic evaluation and avoid data leakage, the dataset is divided into training, validation, and testing sets using a strictly chronological split. This approach preserves the natural temporal ordering of financial data and reflects real-world forecasting conditions.

The dataset is split as follows:

- **Training Set:**

From **September 17, 2018** to **September 17, 2022**, used to train the model and learn historical market patterns.

- **Validation Set:**

From **September 17, 2022** to **September 17, 2023**, used for hyperparameter tuning and model selection.

- **Test Set:**

From **September 17, 2023** to **September 17, 2024**, reserved exclusively for final performance evaluation on unseen data.

This time-based partitioning ensures that future information is never used during training, thereby providing an unbiased and robust assessment of predictive performance.

In this forecasting problem, the target value is the Close Price of the next day and the model inputs historical sequences of features listed above to predict the next time step of the stock's closing price[19 , 20]. This problem becomes a supervised time-series prediction one that can model both short and long term trends in the market.

## 4. Prediction Models

Prediction model presents and describes the forecasting models which are used for prediction of prices of bitcoin. Seven different models were applied which were trained and tested according to the metrics.

### *4.1* Long Short-Term Memory (LSTM)

The LSTMs were intended to resolve the issue of long-term dependencies vanishing gradient problem by adding another layer of regulation on information and letting it be stored for a very long time [21]. Basically, the architecture of an LSTM consists of memory blocks which are recurrent sub networks; their roles are to store the network state over a period of time and to control the flow of information between them. Fig.2 shows the structure of a LSTM block with input signal

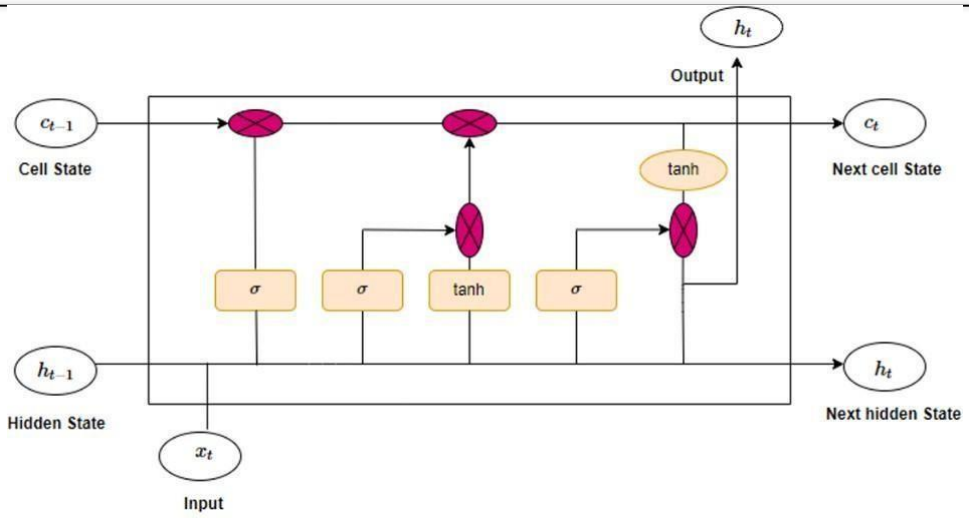



**Figure2: LSTM Architecture**

An LSTM network's forward training process can be characterized by these equations [22].

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \quad (1)$$

$$f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_t\right) \quad (2)$$

$$c_t = f_t * c_{t-1} + i_t * \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) \quad (3)$$

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \quad (4)$$

$$h_t = o_t * \tanh\left(c_t\right) \quad (5)$$

i is the input at time step t, is the hidden state at time step t, is the cell state at time step t, and are the input gate, forget gate and output gate, respectively at time
step t. W and b denote the weight matrices and bias vectors, respectively. The sigmoid function, function outputs values in [0, 1] and tanh function outputs values in [-1, 1].

### 4.2 Gated Recurrent Unit—GRU

GRUs are type of recurrent network proposed in 2014 by [23] as an evolution of LSTM network. The main advantage of both, LSTM and GRU, is their ability to handle inputs sequences of undefined size, carrying an internal state about the past, by opposition to the simple recurrent neural network. Unlike LSTM networks that use an internal memory cell and three different gates for controlling the information flow and retention, GRU networks are simpler networks characterized by two gates, update and reset gate [23]:

It should be also interesting to mention in [23] that, on the task of language modeling in Penn Treebank, the GRU performs better than the LSTM. Comparison between different NLP models [25] states that GRU can be very effective and competitive when compared to LSTMs or CNNs. One important property of the GRU network is its capability to capture long-range

dependencies, contrasting simple RNN networks. Indeed, by means of their gates, GRU networks are able to remember the input received over long sequence time steps and "forget" the information it deemed useless according to the current input and previous state. Therefore, GRUs networks are appropriate for tasks that need memory and long-range dependencies like language translation.

In Figure 3, time state is written as t [26]:

$$u_t = \sigma\left(W_u\left[h_{t-1}, x_t\right]\right) \tag{6}$$

$$r_t = \sigma\left(W_r\left[h_{t-1}, x_t\right]\right) \tag{7}$$

$$h_t = (1 - u_t) * h_{t-1} + u_t * \tanh\left(W\left[r_t * h_{t-1}, u_t\right]\right) \tag{8}$$

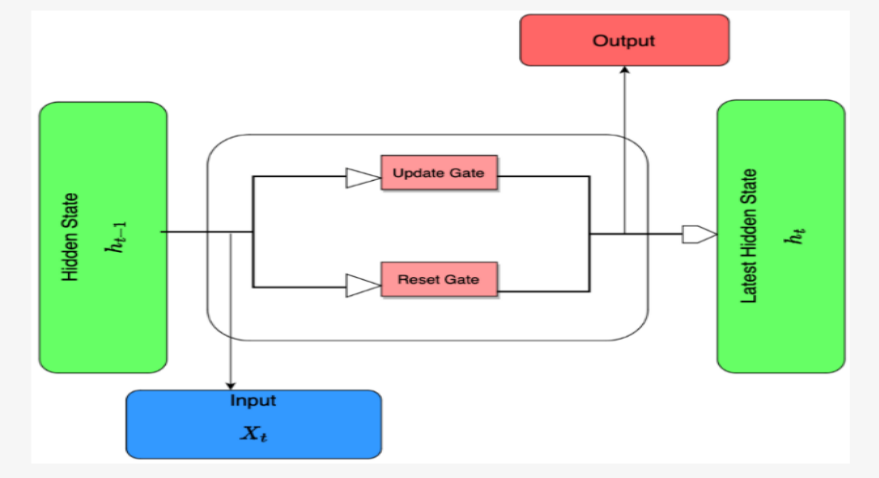



**Figure 3: GRU Architecture**

### *4.3* Bi-Directional LSTM

Bi-LSTM (as shown in Figure 4) is a type of recurrent neural network which is fed with sequence data both forward and backward, so that information both from the past and the future can be applied in classification or prediction. This would probably be appropriate for a task when the information around the current moment might depend on the past events, as well as the future events.

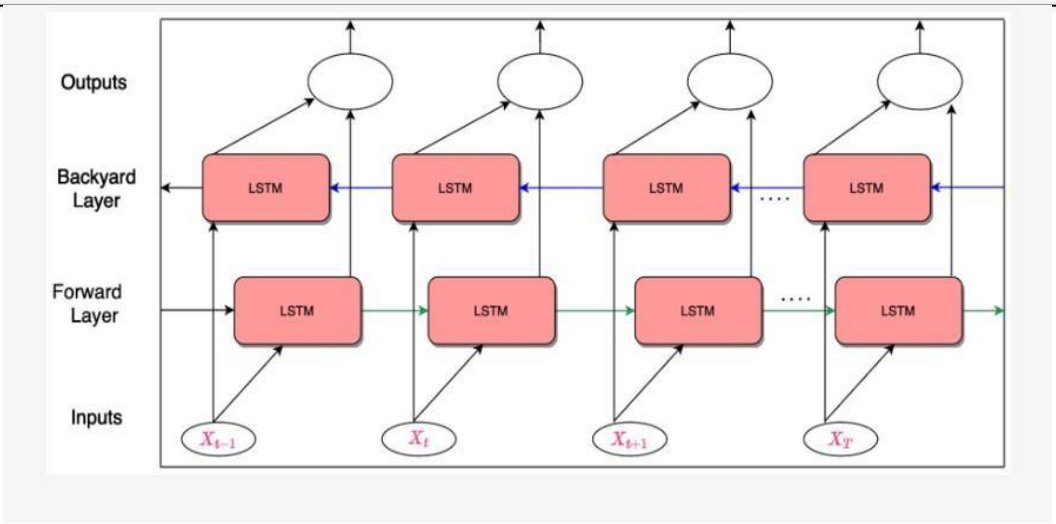



**Figure 4:BiLSTM Architecture**

The Bi-LSTM was first proposed in a 1997 article "Bi-directional Recurrent Neural Networks" [27], where the authors proposed using forward and backward LSTMs to learn past and future context of speech signal processing tasks. After that, Bi-LSTMs implemented on numerous NLP applications such as sentiment analysis, language translation and text classification. Bi-LSTMs also proved to be successful for time series estimation tasks in multiple researches [24,28,29] by implementing Bi-LSTM and achieving great results. In similar fashions, Refs. [22,30] also implemented Bi-LSTM and achieve promising results on time series data.

### *4.4* CryptoMamba Architecture

CryptoMamba is an architecture of Mamba intended to be used in financial time-series forecasting. It uses Mamba blocks to capture long-range dependencies on sequences of data. The model consists of a stack of several computational blocks C-Block, and then the final Merge block which outputs the prediction. The input of the CryptoMamba is the set of features of some fixed number of days in the past and the output is the predicted closing value on the following day. The structure of CryptoMamba is displayed in figure 5. Every C-Block contains a few CMBlocks and an MLP. Each CMBlock is a block of normalization and a Mamba block. The output of each CMBlock is fed to the following CMBlock of the C-Block, making the model hierarchical. The MLP is a linear layer which changes the sequence dimension to match with the next C-Block. Each C-Block's outputs are gathered using the Merge block, which is a simple linear layer that combines the learned features.

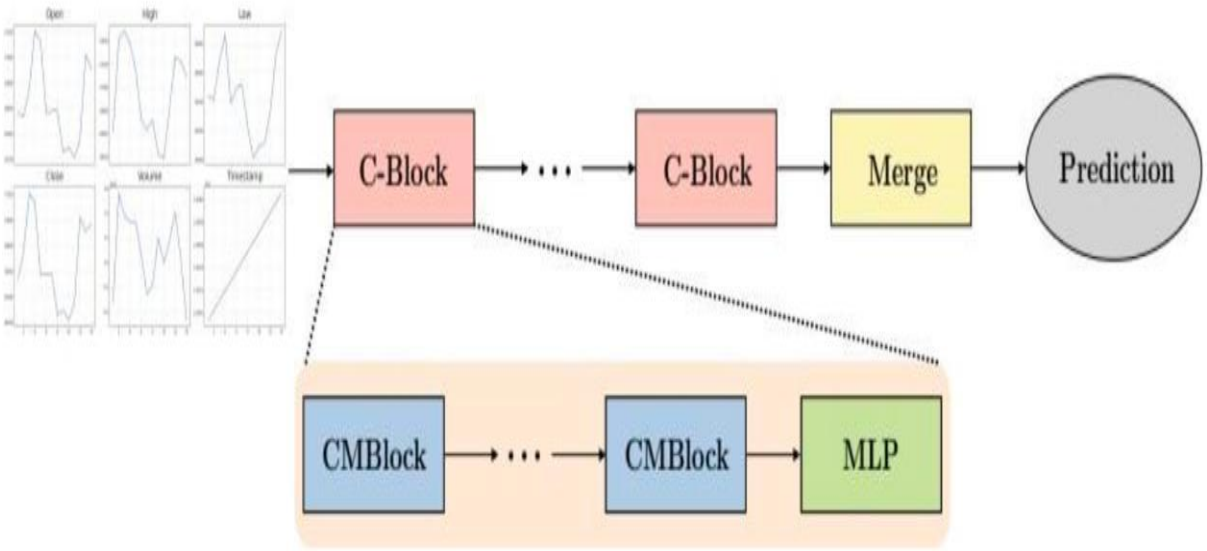



**Figure 5: CryptoMamba Architecture**

The CryptoMamba model is a succession of some C-Blocks and a Merge block. In the C-Block there are some CM Blocks and a final MLP.

The structure of CryptoMamba in the form of hierarchy would capture the dependencies of different time scale; namely short- term and long-term by refining the features across C-Blocks; the model of Mamba is able to learn efficiently because its input- dependent dynamics may enable the model to adopt to financial time-series dynamic and stochasticity[31].In our experiments, every model uses 14 days past as input to predict 1- day ahead close price. And to achieve a fair comparison, the Adam optimizer [29] is used as optimizer along with RMSE loss as the loss function. Every model uses batch size 32 and a learning rate scheduler with weight decay to prevent overfitting. Early stopping is used to prevent over-fitting and we will choose the model checkpoint that has the minimum validation loss to prevent overfitting. For every model, we will use the same train-validation-test split in order to ensure reasonable comparison. And to avoid data leakage, the model is only able to predict after 14 days after the beginning of the input in each data split time; for example, the first prediction sample for validation set should start with input data 14 days after the validation split beginning so that not to include training sample as input data.

## 5. Proposed Model

In this paper, we present a hybrid deep learning architecture combined with a Crypto mamba, a shallow self-attention mechanism and a MLP prediction head to predict future cryptocurrency time-series. The model aims to exploit both long-range temporal dependency and dynamically varying feature dependencies in financial time-series data. Instead of using conventional RNN models, the proposed model adopts a novel state space formulation in order to linearly model the sequence dependencies, a shallow attention module is then added to reinforce the representation with the most useful features in time-series and finally a fully connected MLP layer is employed to predict the future value.

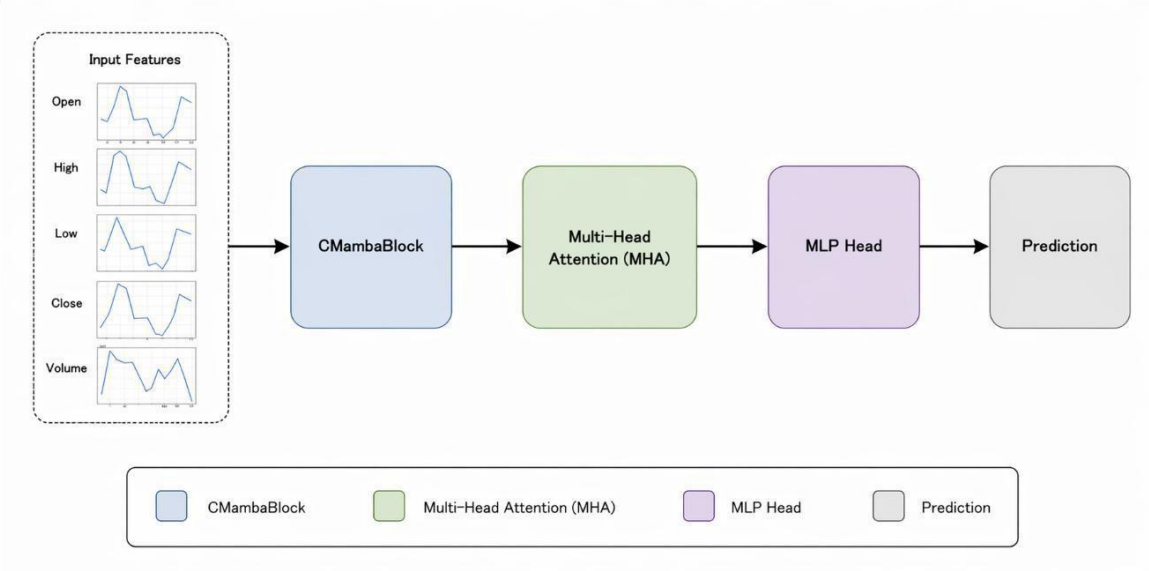



**Figure 6: Proposed Model**

### *5.1* Architecture Description

The overall architecture of the proposed model consists of four main components:

1. **Input Layer:**

The model takes multivariate time-series data $X = \{x1, x2, \ldots, xT\}$, where each $xt \in \mathbb{R}d$represents features such as open, high, low, close, and volume (OHLCV).

2. **Mamba SSM Block:**

The input sequence is processed using a State Space Model to capture temporal dependencies across long horizons.

3. **Self-Attention Module:**

A lightweight attention mechanism is applied to refine feature representations by assigning adaptive importance weights.

4. **MLP Prediction Head:**

The final representation is passed through a fully connected layer to produce the output prediction.

### *5.2* Mamba Block

Mamba [34] adds a data-dependent mechanism to select the state transition for S4, and utilizes hardware-aware parallel algorithms for the loop mode. This mechanism is capable of retrieving context information from long sequences, while keep computationally feasible. Since Mamba has about linear complexity on perplexity series, it performs favorably on long sequence problems than transformers on efficiency and effectiveness. The detailed implementation is presented in the algorithm regarding mamba layer in Alg.1, which contains the entire process flow of data treatment, as well as in Figure 7, which demonstrates the construction of the output at the sequence position.

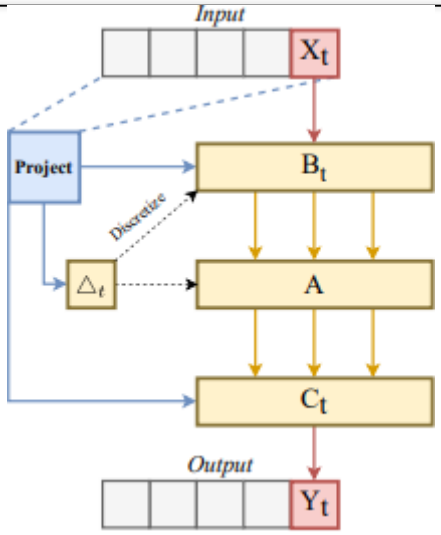



Figure 7: Structure of SSM

**Algorithm 1:** The process of Mamba Block

**Input:** $\mathbf{X} : (B, V, D)$ {Input sequence}
**Output:** $\mathbf{Y} : (B, V, D)$ {Output sequence}

$\mathbf{X}_p, \mathbf{Z} : (B, V, ED) \leftarrow \text{Linear}(\mathbf{X})$ ▷ Expansion}
$\tilde{\mathbf{X}} : (B, V, ED) \leftarrow \text{SiLU}(\text{Conv1D}(\mathbf{X}_p))$ ▷ Local features}
$\mathbf{B}_t, \mathbf{C}_t : (B, V, N) \leftarrow \text{Linear}(\tilde{\mathbf{X}})$ ▷ Input-dependent params}
$\Delta_t : (B, V, ED) \leftarrow \text{Softplus}(\text{Linear}(\tilde{\mathbf{X}}))$ ▷ Step size}
$\mathbf{A}_d, \mathbf{B}_d : (B, V, ED) \leftarrow \text{Discretize}(\mathbf{A}, B_t, \Delta_t)$ ▷ Dynamic discretization}
$\mathbf{Y} : (B, V, ED) \leftarrow \text{SelectiveSSM}(\mathbf{A}_d, \mathbf{B}_d, \mathbf{C}_t)(\tilde{\mathbf{X}})$ {Sequence modeling}
$\mathbf{Y} : (B, V, ED) \leftarrow \mathbf{Y} \odot \text{SiLU}(\mathbf{Z})$ ▷ Gating}
$\mathbf{Y} : (B, V, D) \leftarrow \text{Linear}(\mathbf{Y})$ {Projection}
**return** $\mathbf{Y}$

### 5.3 Self-Attention Mechanism

The attention mechanism is based on human visualization which scans the entire image rapidly to extract target area on which focus needed, namely focal point of attention and gives greater attention on that region to extract fine-grained information and attenuates non-target information. Though, attention mechanism can lead to over-focus on one or two parts of input, failing to extract adequate comprehensive useful information. To tackle this limitation, multi-head attention mechanism represented on right side where the left side demonstrates scaled dot product attention for single attention mechanism and right shows the overall multi-head attention mechanism which essentially contains several scaled dot product attentions in parallel. Multiple attention heads allow model to learn more diversified features by paying attention to different sections of the input information, simultaneously, multiple heads make model steadier and proficient in learning and capturing long-range dependencies and complex structure information. In a multi-head attention inputs were linearly transformed to generate V, Q and K vector of each head, which calculates weighted

output and attention score separately and then each head's output is concatenated and linearly transformed to get the final output. Each head output and that of multi-heads are illustrated in Eq. (1) and Eq. (2).

$$head = Attention(Q, K, V) = V * softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (1)$$

$$\text{MultiHead}\,(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \text{Concat}\,(\text{head}_1, \ldots, \text{head}_\mathrm{h})\,\mathrm{W}^\mathrm{O} \quad (2)$$

In light of the detailed analysis above on these algorithms, this paper employs the advantage of multi-attention mechanism to the features that have been encoded through the convolutional layer in mamba. In this way, the input sequences dependency is captured more effectively and the feature representation and model expressive ability can be further improved to cope with more complicated tasks.

### *5.4* MLP Head:

To further exploit the nonlinear dependencies between features, the MLP Head is applied. In a CMamba block, the temporal dependency among sequences can be effectively modeling through time attention; through global attention layer, the relation between each element in the sequence and others can be modeled; MLP head then processes the sequence-wise features to model complex nonlinear relations.

$$H_{out} = \text{LayerNorm}\Big(H_{in} + \phi(H_{in}W_1 + b_1)W_2 + b_2\Big)$$

Forecasting procedure of our proposed model

**Algorithm 2:** The Forecasting Procedure of CMamba

**Input:** $Batch(U_{in}) = [u_1, u_2, \ldots u_L] : (B, L, V)$
**Output:** $Batch(U_{out}) = [u_{L+1}, u_{L+2}, \ldots u_{L+T}] : (B, T, V)$

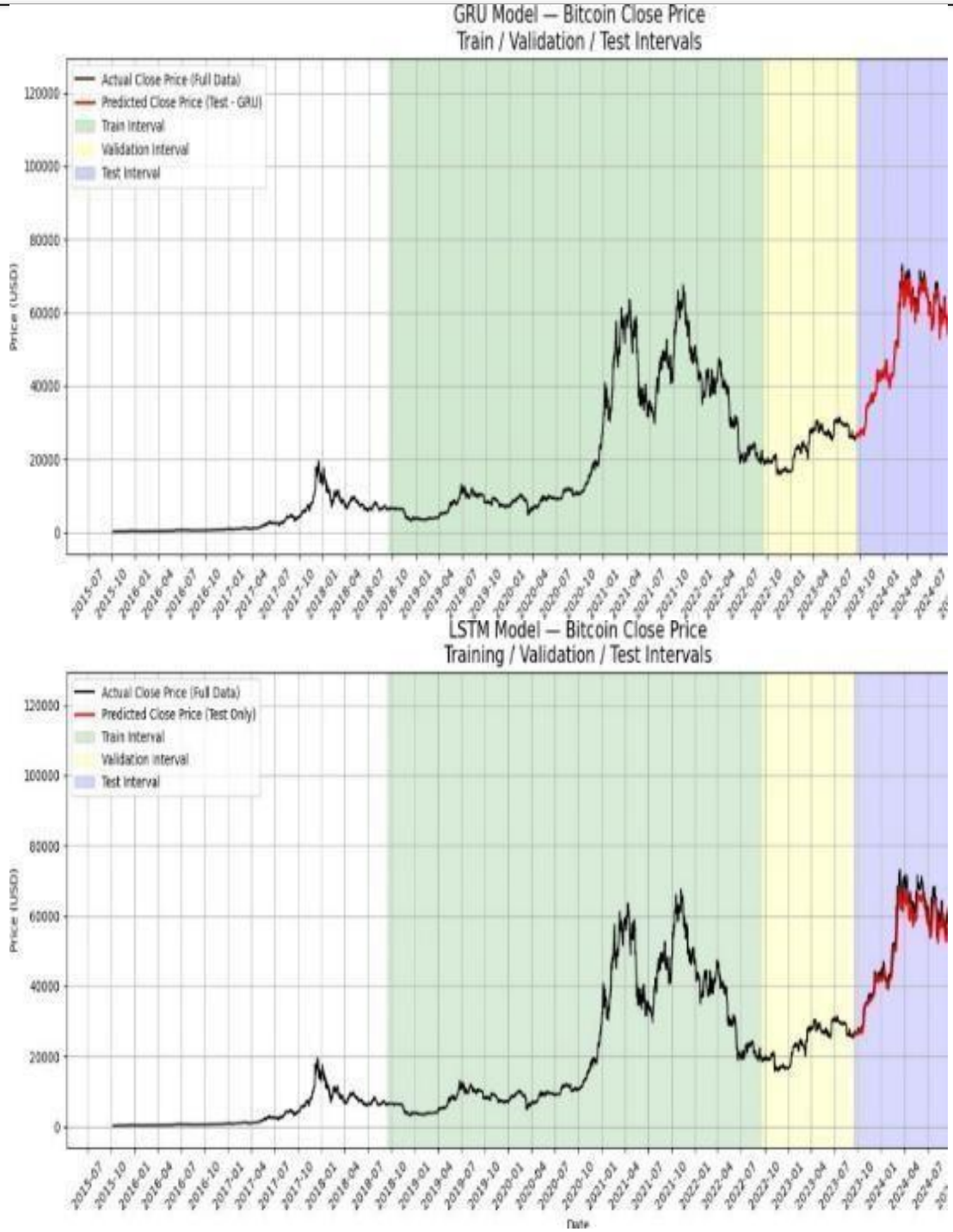
For a batch of data $Batch(U_{in})$...
**Linear Tokenization & Latent Space:**
$X \to H = \text{Linear}(X)$ ▷ Project input features into latent space
**For** $l$ in $l$ CMamba Blocks**:**
    **CMamba Block:**
    $H \to (X_p, Z) = \text{Linear}(H)$ ▷ Feature expansion + gating branch separation
    $X_p \to \tilde{X} = \text{SiLU}(\text{Conv1D}(X_p))$ ▷ Extract local temporal patterns
    $\tilde{X} \to (B_t, C_t) = \text{Linear}(\tilde{X})$ ▷ Generate input-dependent state parameters
    $\tilde{X} \to \Delta_t = \text{Softplus}(\text{Linear}(\tilde{X}))$ ▷ Compute adaptive step size
    $(A, B_t, \Delta_t) \to (A_d, B_d) = \text{Discretize}(A, B_t, \Delta_t)$ ▷ Convert continuous SSM into discrete form
    $(\tilde{X}, A_d, B_d, C_t) \to H_m = \text{SelectiveSSM}(...)$ ▷ Perform sequence modeling via selective scan
    $(H_m, Z) \to H = \text{Linear}(H_m \odot \text{SiLU}(Z))$ Apply gating and project features
    **Multi-Head Attention Layer:**
    $H \to (Q, K, V) = \text{Linear}(H)$ ▷ Generate query, key, value representations
    $(Q, K, V) \to A = \text{Softmax}(\frac{QK^T}{\sqrt{d}})V$ ▷ Compute attention scores and weighted aggregation
    $(H, A) \to H = \text{LayerNorm}(H + A)$ ▷ Residual connection + normalization
    **MLP Head & Final Stabilization:**
    $H \to H_{ff} = \text{GELU}(HW_1 + b_1)W_2 + b_2$ ▷ Nonlinear feature transformation
    $(H, H_{ff}) \to H = \text{LayerNorm}(H + H_{ff})$ ▷ Residual connection + stabilization
    **Prediction:**
    $H \to Y = \text{Linear}(H)$ ▷ Map latent features to forecast output
**End for**
**Output Preparation:**
$Batch(U_{out}) \leftarrow \text{Transpose}(Y)$
**return** $Y$

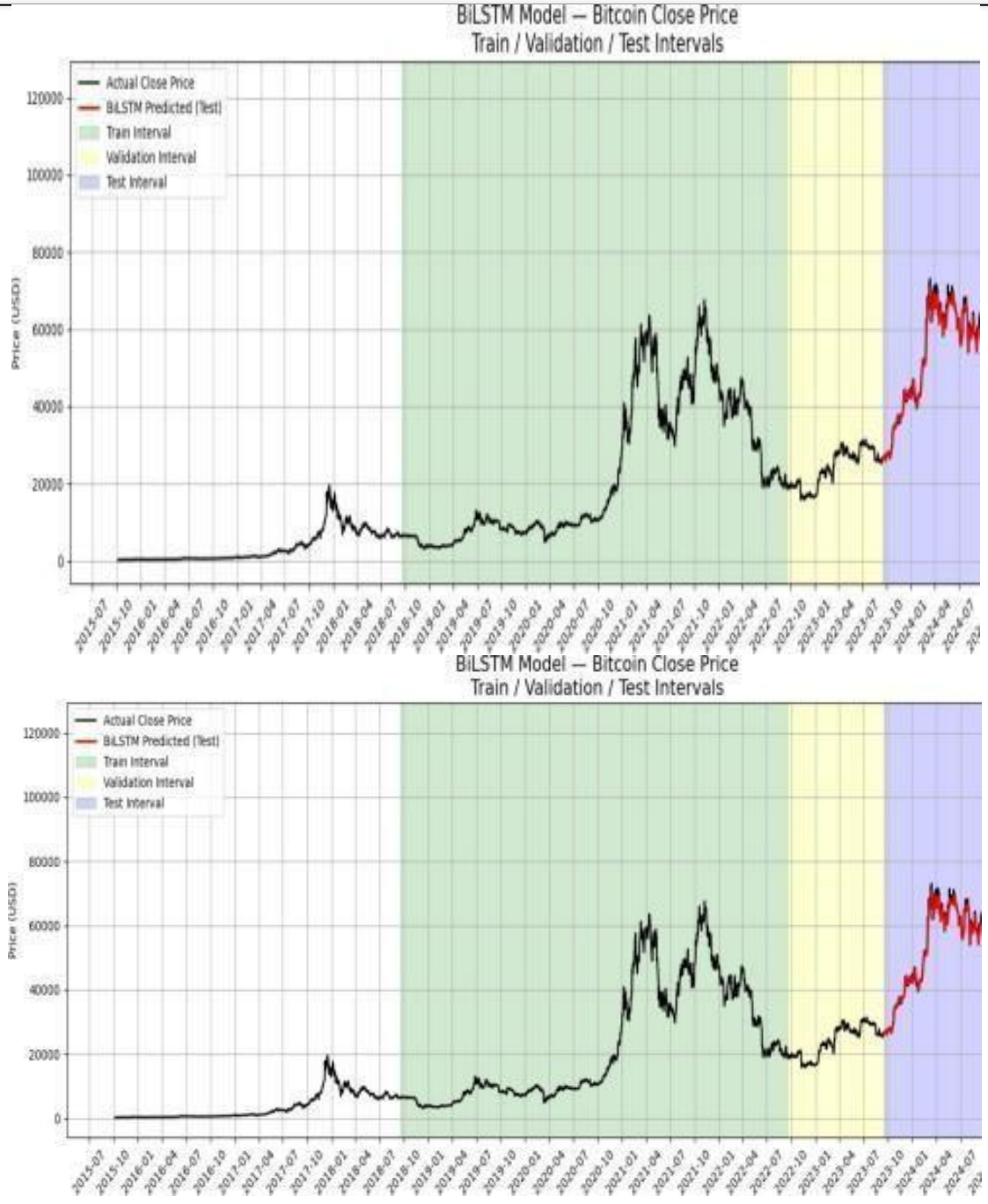
## 6. Results s Discussion

Table 1 lists a comparison between various deep learning models like LSTM, GRU, Bi-LSTM, CryptoMamba-v, and the developed model CryptoMamba+AttentionMLP-v and compares their performance across RMSE, MAPE, MAE and 2. It can be observed that the performance of the base model LSTM performs the weakest among all other deep learning models yielding the highest RMSE (2618.06), MAE (1929.68) and a comparatively low 2 (0.9645) to display its weakness in modeling long-range sequential dependencies. GRU improved upon the LSTM in reducing the error values along with an increased 2 of 0.9716. Bi-LSTM performed even better using Bidirectional sequence learning with lower RMSE (1866.78) and MAE (1327.97) values and 2 of 0.9819. The results clearly shows how capturing both future and past contexts significantly aids in time-series forecasting. Finally the CryptoMamba-v model yields better performance than all the recurrent architectures with reduced RMSE (1594.38) and improved 2 of 0.9868 demonstrating the robustness of Mamba in modeling long-range dependencies and complex temporal features.

**Table 1: Comparative performance proposed model with different deep learning models**

| Method | RMSE | MAPE | MAE | $R^2$ |
|---|---|---|---|---|
| LSTM | 2618.06 | 3.42 | 1929.68 | 0.9645 |
| GRU | 2341.89 | 3.18 | 1813.11 | 0.9716 |
| Bi-LSTM | 1866.78 | 2.41 | 1327.97 | 0.9819 |
| CryptoMamba-v | 1594.38 | 2.14 | 1142.3 | 0.9868 |
| **CryptoMamba+AttentionMLP-v** | **1491.02** | **1.9** | **1029.74** | **0.9885** |


GRU Model — Bitcoin Close Price
Train / Validation / Test Intervals
Actual Close Price (Full Data)
Predicted Close Price (Test - GRU)
Train Interval
Validation Interval
Test Interval
Price (USD)
LSTM Model — Bitcoin Close Price
Training / Validation / Test Intervals
Actual Close Price (Full Data)
Predicted Close Price (Test Only)
Train Interval
Validation Interval
Test Interval
Price (USD)
Date


BiLSTM Model — Bitcoin Close Price
Train / Validation / Test Intervals
Actual Close Price
BiLSTM Predicted (Test)
Train Interval
Validation Interval
Test Interval
Price (USD)
BiLSTM Model — Bitcoin Close Price
Train / Validation / Test Intervals
Actual Close Price
BiLSTM Predicted (Test)
Train Interval
Validation Interval
Test Interval
Price (USD)

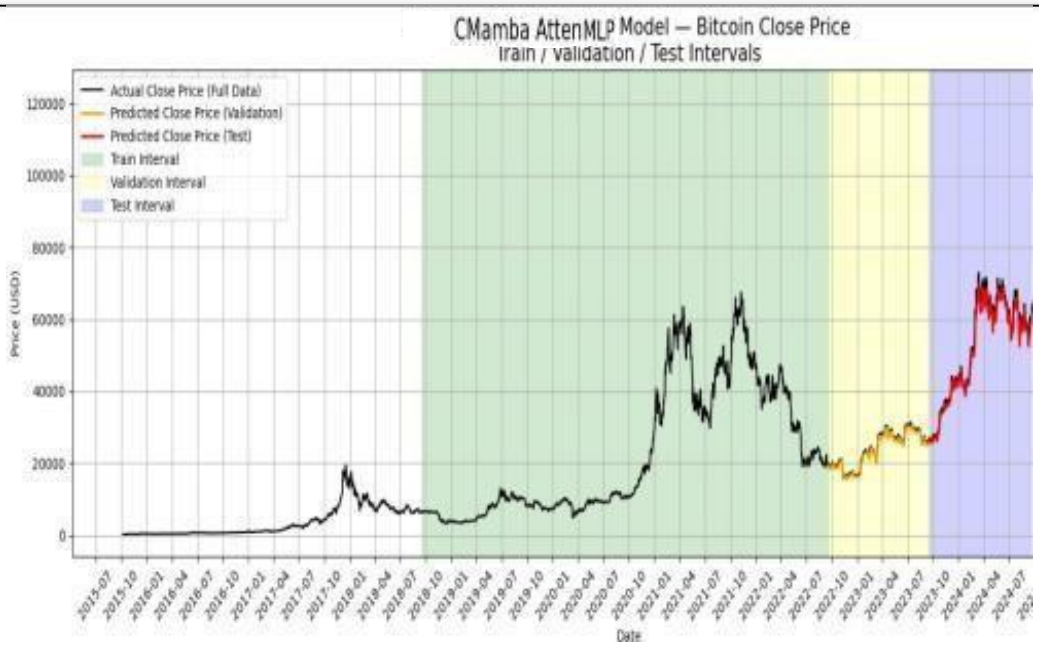



**Figure 8: Predictions of all previous DL models and proposed model**

Finally, the suggested CryptoMamba+AttentionMLP-v obtains optimal results for all evaluation indicators, reporting the lowest value for RMSE (1491.02), MAPE (1.9) and MAE (1029.74), as well as the largest 2value (0.9885). The integration of the AttentionMLP module helps the model weigh the features and temporal relationships the most pertinent for a prediction with the highest precision and stability.

verall, the experiments show that the proposed hybrid model significantly outperforms the conventional RNN based methods and the stand-alone Crypto Mambamodel.

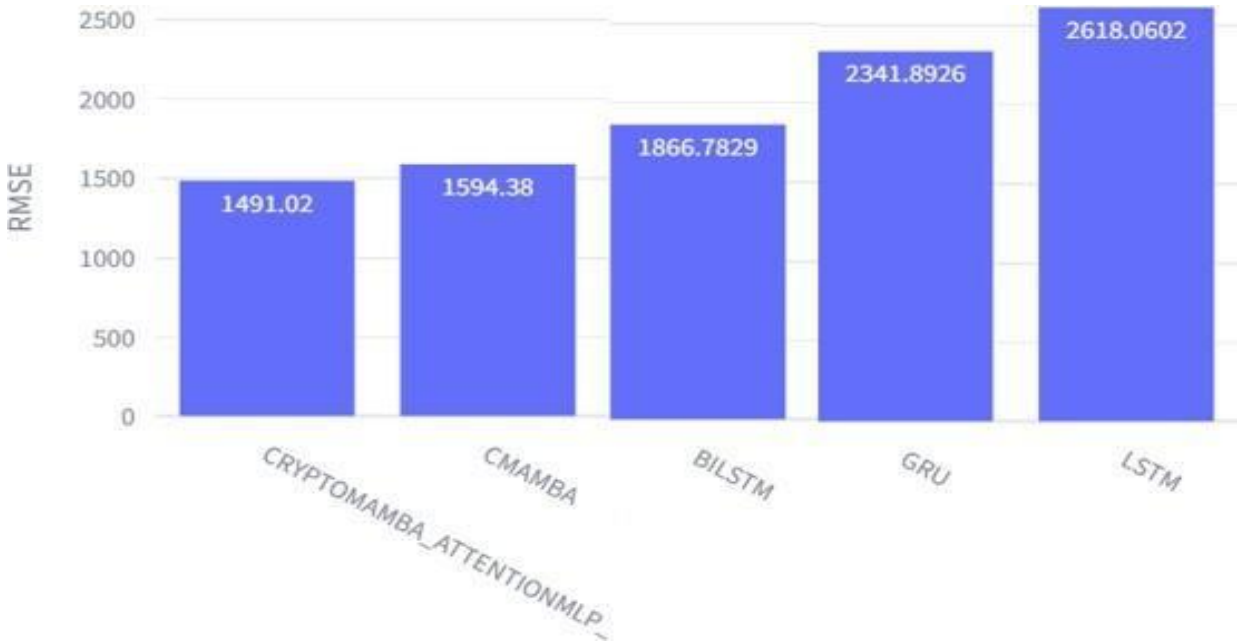



**Figure 9: RMSE values of all previous DL models and Proposed model**

## 7. Conclusion

In this work, a new hybrid model is proposed, named CryptoMamba+AttentionMLP-v, for time series forecasting. It combine with the Mamba architecture and an Attention-based Multi-Layer Perceptron to effectively capture long-range dependencies and salient temporal features. The proposed model have been rigorously tested and compared with widely used deep learning models such as LSTM, GRU, Bi-LSTM, and the baseline model CryptoMamba-v. Experimental results have shown that the proposed model outperforms all comparative methods over a series of prediction metrics by achieving the lowest errors and the highest coefficient of determination (2). This performance is the result of its ability to focus dynamically on the most informative patterns using the attention mechanism, and to efficiently modeling complex sequential dependencies with the Mamba framework. Such combined characteristics make the proposed model well- suited to practical time-series forecasting tasks, especially under high volatilities in the context of financial market time-series forecasting. In conclusion, the CryptoMamba+AttentionMLP-v model achieves a breakthrough beyond existing time-series models, and future research directions include extending it to multi-step prediction, considering external influential variables, and real-time application.

## 8. REFERENCES

Jesse, A. Algorithmic Trading: Leveraging AI and ML in Finance. RapidInnovation. Available online: https://www.rapidinnovation.io/post/algorithmic-trading-leveraging- ai-and-ml-in-finance (accessed on 28 September 2024).

Shah, D.; Isah, H.; Zulkernine, F. Stock Market Analysis: A Review and Taxonomy of Prediction Techniques. Int. J. Financ. Stud. 2019, 7, 26. [CrossRef]

T. A. Muniye, M. Rout, and S. Satapathy, "Bitcoin Price Prediction and Analysis Using Deep Learning Models," in Proceedings of the International Conference on Computational Intelligence and Data Science, Springer, Singapore, Oct. 2020, pp. 635–641, doi: 10.1007/978-981-15-5397-4_63.

V. Chang, Q. A. Xu, A. Chidozie, and H. Wang, "Predicting Economic Trends and Stock Market Prices with Deep Learning and Advanced Machine Learning Techniques,"Electronics, vol. 13, no. 17, p. 3396, Aug. 2024,doi: 10.3390/electronics13173396.

P. Han, H. Chen, A. Rasool, Q. Jiang, and M. Yang,"MFB: A Generalized Multimodal Fusion Approach for Bitcoin Price Prediction Using Time-Lagged Sentiment and Indicator Features,"Expert Systems with Applications, vol. 261, p. 125515, 2025,doi: 10.1016/j.eswa.2024.125515.

J. Xie, Y. Cui, F. Huang, C. Liu, and K. Zheng, "MARINA: An MLP-Attention Model for Multivariate Time-Series Analysis," Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM), Atlanta, GA, USA, Oct. 2022, pp. 2229–2238, doi: 10.1145/3511808.3557386.

M. S. Sepehri, A. Mehradfar, M. Soltanolkotabi, and S. Avestimehr, "CryptoMamba: Leveraging State Space Models for Accurate Bitcoin Price Prediction," IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 2025.

S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," arXiv preprint arXiv:1803.01271, 2018.

D. M. Teixeira and R. S. Barbosa, "Stock Price Prediction in the Financial Market Using Machine Learning Models," Computation, vol. 13, no. 1, p. 3, 2025.

G. Kim, D.-H. Shin, J. G. Choi, and S. Lim, "A Deep Learning-Based Cryptocurrency Price Prediction Model That Uses On-Chain Data," IEEE Access, vol. 10, pp. 56232–56247, 2022, doi: 10.1109/ACCESS.2022.3177888.

R. Mousa, M. Afrookhteh, H. Khaloo, A. A. Bengari, and G. Heidary, "Forecasting of Bitcoin Prices Using Hashrate Features: Wavelet and Deep Stacking Approach," arXiv preprint arXiv:2501.13136, 2025.

H. Liu, Z. Dai, D. R. So, and Q. V. Le, "Pay Attention to MLPs," arXiv preprint arXiv:2105.08050, 2021.

C. C. Ukwuoma et al., "Hydrogen production prediction from co-gasification of biomass and plastics using attention-gated MLP model," Renewable Energy, vol. 249, p. 123076, 2025.

Y. Li et al., "A TCN-Based Hybrid Forecasting Framework for Hours-Ahead Utility- Scale PV Forecasting," IEEE Transactions on Smart Grid, vol. 14, no. 5, pp. 4073–4084, 2023.

Yahoo Finance. Available online: https://finance.yahoo.com/

Cort, J. Willmott and Kenji Matsuura. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Clim. Res. 2005, 30, 79–82.

Ken, S. Mean Squared Error. Encyclopedia Britannica, 2024. Available online: https://www.britannica.com/science/meansquared-error

Scott, N. Coefficient of Determination: How to Calculate It and Interpret the Result. Investopedia. 2024. Available online: https://www.investopedia.com/terms/c/coefficient-of-determination.asp

Scikit-Learn. Available online: https://scikit-learn.org (accessed on 5 October 2024)

Pandas. Available online: https://pandas.pydata.org/ (accessed on 28 September 2024).

Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780.

Ayoobi, N.; Sharifrazi, D.; Alizadehsani, R.; Shoeibi, A.; Gorriz, J.M.; Moosaei, H.; Khosravi, A.; Nahavandi, S.; Chofreh, A.G.; Goni, F.A.; et al. Time series forecasting of new cases and new deaths rate for COVID-19 using deep learning methods. *Results Phys.* **2021**, *27*, 104495.

Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.

Yang, S.; Yu, X.; Zhou, Y. Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example. In Proceedings of the 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI), Shanghai, China, 12–14 June 2020; pp. 98–101.

Wang, X.; Jiang, W.; Luo, Z. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In Proceedings of the Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–16 December 2016; pp. 2428–2437.

Dey, R.; Salem, F.M. Gate-variants of gated recurrent unit (GRU) neural networks. In Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 6–9 August 2017; pp. 1597–1600.

Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681.

Lai, S.; Ye, C.; Zhou, H.J.H. Chinese stock trend prediction based on multi-feature learning and model fusion. In Proceedings of the 2021 IEEE International Conference on Smart Data Services (SMDS), Chicago, IL, USA, 5–10 September 2021; pp. 18–23.

Singh, A.; Kumar, A.; Akhtar, Z. Bitcoin Price Prediction: A Deep Learning Approach. In Proceedings of the 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 26–27 August 2021; pp. 1053–1058.

Althelaya, K.A.; El-Alfy, E.S.M.; Mohammed, S. Stock market forecast using multivariate analysis with bidirectional and stacked (LSTM, GRU). In Proceedings of the 2018 21st Saudi Computer Society National Computer Conference (NCC), Riyadh, Saudi Arabia, 25–26 April 2018; pp. 1–7.

Sepehri, Mohammad Shahab, et al. "CRYPTOMAMBA: Leveraging State Space Models for Accurate Bitcoin Price Prediction." University of Southern California, Department of Electrical and Computer Engineering.

Gu, A., Dao, T., 2023. Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752.

Elfwing, S., Uchibe, E., Doya, K., 2017. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural networks : the official journal of the International Neural Network Society 107, 3–11.

Gu, A., Dao, T., 2023. Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 .