

Hybrid LSTM–HMM Framework for Regime-Aware Cryptocurrency Price Forecasting

Tran Kim Toai¹ and Nguyen Phan Nhat Minh²

¹ Ho Chi Minh City University of Technology and Engineering (HCM-UTE)
No 1 Vo Van Ngan Street, Thu Duc Ward, Ho Chi Minh City, Vietnam
toaitk@hcmute.edu.vn (corresponding author)

² Nguyen Phan Nhat Minh
phannhatminh288@gmail.com

Abstract. The rapid evolution of digital finance and blockchain technology has positioned cryptocurrencies as significant yet highly volatile financial assets. Their nonlinear and regime-shifting price behavior challenges traditional econometric and deep learning models, creating a need for more adaptive forecasting frameworks. Addressing this research gap, the present study proposes a **hybrid Hidden Markov Model–Long Short-Term Memory (HMM–LSTM)** architecture that jointly captures **latent market regimes** and **temporal dependencies** in cryptocurrency price dynamics. The HMM identifies latent market regimes, while the LSTM captures temporal dependencies. Technical indicators, including Bollinger Bands, EMA, and MACD, support feature benchmarking. Empirical results show that hybrid gains are asset-dependent and feature-dependent: performance is strongest for Ethereum and BNB, comparable for SOL, and weaker than the standalone LSTM for Bitcoin, indicating that regime information is conditionally beneficial for financial forecasting. Key findings highlight that the hybrid design effectively mitigates overfitting, improves generalization across assets, and captures regime-dependent price movements and robust regime-sensitive forecasting. The implications of this research extend to **AI-driven financial forecasting**, trading automation, and contributing to the advancement of intelligent, data-centric financial analytics.

Keywords: Hybrid Deep Learning, Long Short-Term Memory (LSTM), Hidden Markov Model (HMM), Cryptocurrency Price Forecasting, Regime Detection.

1. Introduction

Bitcoin, introduced by Nakamoto [25], established the concept of blockchain, a decentralized, publicly distributed ledger designed to record transactions securely and transparently without reliance on third-party intermediaries. This innovation enabled a peer-to-peer (P2P) electronic payment system that operates independently of centralized authorities. Since its inception, the underlying blockchain framework has inspired extensive research and numerous applications across finance, supply chain management, healthcare, and other data-intensive domains.

Within this complex and heterogeneous environment, cryptographic protocols have been increasingly adopted to address key data and transaction challenges such

as data integrity, decentralized storage, enhanced security, and trust management [15,6,12]. Decentralized consensus mechanism eliminates single points of failure and enhances transparency across distributed financial networks. Notably, it serves as a foundational component for cryptocurrency platforms, enabling autonomous, smart-contract-based financial exchanges without reliance on trusted third-party intermediaries [39]. Consequently, financial system integrated with cryptocurrency are gaining significant traction as enablers of secure, scalable, and autonomous digital economies.

Therefore, this study examines the dynamics of cryptocurrency prices, with a focus on Bitcoin as a representative asset. We propose a hybrid predictive framework that integrates the Hidden Markov Model (HMM) for regime detection with the Long Short-Term Memory (LSTM) network for sequential forecasting. While HMMs [4] and LSTMs [22] have independently demonstrated potential in modeling temporal dependencies, their standalone performance often remains limited in volatile and non-stationary environments such as cryptocurrency markets. By experimentally validating a combined HMM–LSTM architecture [1], this work aims to enhance predictive accuracy and provide deeper insight into the structural and behavioral regimes governing cryptocurrency price fluctuations.

The remainder of this paper is structured as follows. Section 2 provides a comprehensive review of the relevant literature, followed by Section 3, which presents the theoretical foundations and research methodology. Section 4.1 describes the dataset employed in this study, and the partitioned data is handled in Section 4.4, while Section 4.2 outlines the process of feature extraction and selection. Correlation analysis between features is evaluated in Section 4.3 while the implementation details of the proposed hybrid modeling framework are discussed in Section 4.5. Section 5 reports and analyzes the empirical findings, including model evaluation and performance comparison. Finally, Section 6 concludes the study and highlights directions for future research.

2. Literature Review

This section illustrates the progression from standalone Long Short-Term Memory (LSTM) networks to advanced hybrid architectures in financial time series forecasting, particularly for stock and cryptocurrency markets. To address the non-stationary and volatile behavior of cryptocurrencies, recent studies have employed regime-switching mechanisms and Hidden Markov Model (HMM)–based graphical frameworks to model dynamic interdependencies and volatility clustering among assets. Hybrid architectures that integrate LSTM with classical time series models, such as ARIMA and GARCH, have gained prominence due to their ability to decouple linear and nonlinear components, frequently achieving lower root mean square error (RMSE) compared to single-model baselines. Contemporary research further enhances predictive robustness by incorporating heterogeneous data sources, including market prices, sentiment indicators, on-chain metrics, and web search trends, and by leveraging advanced architectures such as AdaBoost-LSTM ensembles, transformer-based attention mechanisms, and stacked deep learning networks.

While early financial forecasting studies relied on linear statistical methods (e.g., ARIMA, VAR), modern deep learning approaches, including LSTM and Transformer models, have substantially improved temporal representation learning. This evolution has enabled superior performance in modeling complex, nonlinear market dynamics. Originally introduced by Hochreiter and Schmidhuber [20], the LSTM architecture demon-

strated an exceptional ability to address the long-term dependency and vanishing gradient problems inherent in traditional recurrent neural networks.

Subsequent empirical studies have successfully adapted this architecture for modern financial forecasting tasks. For instance, Tran et al. [34] reported that LSTM models achieved an accuracy of approximately 88% in predicting Bitcoin prices and trends, outperforming conventional architectures such as standard Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN). In contrast, traditional statistical and regression-based approaches exhibit only moderate success in financial time series modeling. They are constrained by their linear assumptions and reliance on normally distributed variables—conditions that are rarely met in cryptocurrency markets, which are characterized by high volatility, heavy-tailed distributions, and structural non-stationarity. As Gurgul et al. [17] highlight, such methods fail to capture complex nonlinear dependencies and dynamic volatility structures, leading to reduced forecasting accuracy, particularly over long horizons.

Iman Abu Hashish.[1] proposed a hybrid predictive framework for Bitcoin price forecasting that integrates HMM with a Genetic Algorithm-optimized Long Short-Term Memory (GA-LSTM) network. The HMM component captures latent market regimes and underlying structural dynamics, while the GA is employed to fine-tune LSTM hyperparameters, therefore enhancing predictive accuracy and model generalization. This approach effectively combines probabilistic regime detection with deep learning-based temporal modeling to uncover non-observable patterns in cryptocurrency price movements. Ahmed Bouteska [8] conducted comprehensive experiments on historical time series data from four major cryptocurrencies. The study evaluated multiple machine learning paradigms, including boosting-based ensemble algorithms, recurrent deep neural networks, and hybrid two-stage models combining ARIMA with recurrent architectures. Empirical results demonstrate that LightGBM outperforms alternative approaches, yielding highly profitable trading strategies for investors in Bitcoin, Ethereum, and Litecoin. Hosun Ryou [31] introduced a momentum-based investment framework that employs Hidden Markov Models (HMMs) to capture regime-dependent market dynamics. The study compared HMM-driven momentum portfolios with conventional momentum strategies, demonstrating superior performance, particularly under a one-month formation and one-week holding period. Results indicate that the HMM-based approach effectively enhances return predictability and portfolio profitability. However, the findings are constrained by the study's focus on long-only portfolios within the Korean equity market, limiting broader generalization.

The study employs a diverse set of evaluation metrics that reflect both statistical accuracy and statistical error. Benchmarking studies consistently demonstrate that hybrid LSTM approaches outperform standalone models. Notably, LSTM-ARIMA combinations have shown significant reductions in RMSE compared to non-hybrid baselines in both daily and hourly forecasting scenarios [26]. Hybrid systems incorporating denoising techniques, CNN feature extraction, and GAN augmentation have reported higher backtest returns and elevated predictive accuracy compared to traditional ARIMA and standalone LSTM approaches. Although hybrid approaches generally report superior predictive performance, considerable variability persists across studies owing to inconsistencies in data preprocessing protocols, evaluation horizons, and the treatment of transaction costs [32].

Although hybrid forecasting models have achieved notable predictive success, the field still lacks rigorous frameworks for uncertainty quantification, which is particularly critical given the high volatility and stochastic nature of cryptocurrency markets. Probabilistic and Bayesian forecasting techniques remain underexplored in this domain, and most existing studies continue to employ sequential or pipeline-based frameworks that require separate optimization stages [2]. Moreover, many studies overlook essential real-world considerations, such as detailed transaction-cost modeling, market-impact and slippage effects, latency constraints, risk-management mechanisms, and regulatory compliance. Current research provides limited investigation into the dynamic interdependencies among cryptocurrencies and their linkages with conventional financial assets within hybrid modeling frameworks. These omissions hinder the development of operationally viable hybrid models and limit research addressing the computational and practical challenges of real-time deployment in production trading environments [13].

The present study extends our previous work, which examined the predictive performance and stability of HMM and LSTM networks in modeling stock and cryptocurrency time series. Building upon that foundation, this research focuses specifically on forecasting Bitcoin closing prices and introduces a systematic feature selection framework to enhance predictive accuracy. The study evaluates multiple subsets of technical indicators and market-derived variables to assess their marginal contribution to LSTM performance when combined with latent regime information extracted through HMM. This hybrid approach enables a more robust benchmarking and optimization process, improving both model interpretability and forecasting reliability.

3. Methodologies

This section focuses on exploring existing methodologies and identifying the optimal hybrid architecture. The evolution of hybrid LSTM-HMM frameworks can be broadly characterized by three developmental stages: (i) early cascade models with minimal interaction between components, (ii) feature-level fusion architectures enhancing regime transition modeling, and (iii) advanced end-to-end frameworks with fully integrated graphical structures and joint optimization mechanisms [24]. Contemporary forecasting frameworks increasingly segment financial time series into distinct regimes using HMM or Markov chain techniques, followed by regime-specific deep architectures such as LSTM networks or Transformers [14]. Hybrid approaches that combine LSTM with statistical models, such as ARIMA or GARCH, or integrate methods like AdaBoost and stacking have demonstrated superior capability in capturing both long-term trends and abrupt regime shifts [10]. Numerous studies employ limited evaluation periods or insufficient data partitioning between training and testing sets, which may result in overfitting and inflated performance estimates [27]. Furthermore, the lack of standardized benchmark datasets hampers reproducibility and the objective comparison of model performance across research groups [35]. To address these limitations, this study proposes an advanced hybrid forecasting framework that integrates LSTM networks with HMM and introduces a data-driven benchmarking methodology designed to enhance model generalization and facilitate comparative evaluation.

3.1. Hidden Markov Model

A Hidden Markov Model **HMM** is a generative probabilistic framework in which a sequence of observations $\mathbf{Y}_{1:T} = (Y_1, \dots, Y_T)$ is produced by a sequence of hidden states $\mathbf{S}_{1:T} = (S_1, \dots, S_T)$. Hidden Markov Models are valuable for time-series modeling in various fields, including speech recognition, computational biology, computer vision, and financial forecasting.

An HMM rests on two first-order assumptions. **First**, each observable value recorded at time t is generated by a process that occupies a hidden state S_t ; this state is not directly visible to the observer. **Second**, the hidden states themselves follow the first-order Markov property: once the previous state S_{t-1} is known, the probability of the current state S_t is independent of any states prior to $t - 1$.

Joint probability of a sequence Likewise, the output of an HMM obeys the Markov property. Consequently, the joint probability of a sequence of hidden states and observations factorises as described in the equation 1.

$$P(\mathbf{S}_{1:T}, \mathbf{Y}_{1:T}) = P(S_1) P(Y_1 | S_1) \prod_{t=2}^T P(S_t | S_{t-1}) P(Y_t | S_t). \quad (1)$$

Here, $P(S_1)$ is the initial-state distribution, $P(S_t | S_{t-1})$ are the state-transition probabilities, and $P(Y_t | S_t)$ are the emission probabilities. To simplify, an HMM is fully specified by $\mathbf{A}$, $\mathbf{B}$, and $\boldsymbol{\pi}$, together with the implicit by the number of hidden states M and the number of distinct observations N :

- $\mathbf{A} \in \mathbb{R}^{M \times M}$: the state–transition probability matrix, where $a_{ij} = P(S_t = j | S_{t-1} = i)$;
- $\mathbf{B} \in \mathbb{R}^{M \times N}$: the emission (observation) probability matrix, where $b_{ik} = P(Y_t = k | S_t = i)$;
- $\boldsymbol{\pi} \in \mathbb{R}^M$: the initial-state distribution, where $\pi_i = P(S_1 = i)$.

Using this notation, an HMM can be compactly written in the detailed form of the equation 2.

$$\boldsymbol{\lambda} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}). \quad (2)$$

HMM gives rise to three fundamental computational problems:

- **Problem 1 – Likelihood (evaluation).** Given a model $\boldsymbol{\lambda} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ and an observation sequence $\mathbf{Y}$, compute the probability that the model generated the observation data $\Pr(\mathbf{Y} | \boldsymbol{\lambda})$. How do we compute the probability of $\mathbf{X}$? The solution to this problem allows us to select the competing model that best matches the observation sequence.
- **Problem 2 – Decoding.** Given a model $\boldsymbol{\lambda} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ and observation sequence, determine the optimal sequence of hidden states in a Markov process. How do we find the state sequence that best explains the observations? The solution to this problem attempts to uncover the hidden part of the stochastic mode

- **Problem 3 – Learning (parameter estimation).** Given only the observation sequence, infer the model parameters $\mathbf{A}$, $\mathbf{B}$, and π that maximize the likelihood of the data to describe the observation sequence best. The Baum–Welch algorithm (an EM procedure) is typically employed.

In the present study, we begin with a sequence of cryptocurrency returns, construct an initial HMM, determine the range of states to compute on each component for the evaluation of the data likelihood through metrics BIC for optimization, and then decode the optimal hidden-state path (lowest BIC score) as part of the data-modeling stage in Section 4.5. Then, the Baum–Welch algorithm is used to locally maximize the likelihood by adjusting the model parameters to maximize the probability of an observation sequence for a given model. These complex techniques are described in Algorithm 1 specifically.

3.2. Long Short-Term Memory Network

Long Short-Term Memory (LSTM) networks are a specialized form of RNN introduced to overcome the vanishing and exploding gradient problems. Long-term dependencies, such as those found in LSTM, can retain relevant information over hundreds or even thousands of time-steps, making it especially effective for tasks that require long-range temporal context [1].

Cell state layer. At the heart of the LSTM lies the cell state, which provides a direct route for information propagation over time. This “information highway,” illustrated as the horizontal connection across the top of the model, preserves long-term dependencies and mitigates gradient decay.

Forget gate layer. The first step inside an LSTM cell is deciding which information to discard from the previous cell state. This decision is made by the sigmoid-activated *forget gate*. It receives the previous hidden state $\mathbf{h}_{t-1}$ and the current input $\mathbf{x}_t$ and returns a vector $\mathbf{f}_t$ whose elements lie in the interval $[0, 1]$. For each component of the previous cell state $\mathbf{c}_{t-1}$, a value of 1 in $\mathbf{f}_t$ means that information is kept entirely, whereas a value of 0 means it is forgotten completely, as described in the equation 3.

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \begin{bmatrix} \mathbf{h}_{t-1} \\ \mathbf{x}_t \end{bmatrix} + \mathbf{b}_f). \quad (3)$$

Enter gate layer. The next step is to decide what new information should be added to the cell state. This involves two sub-steps. First, a sigmoid-activated layer called the *input gate layer* selects which elements to update in the formula (see Equation 4). Continue it is a *tanh* layer that generates a vector for the new value $\tilde{\mathbf{C}}_t$ added to the cell state in the given equation 5. In the following step, we will combine 2 values to generate the updated cell state

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \begin{bmatrix} \mathbf{h}_{t-1} \\ \mathbf{x}_t \end{bmatrix} + \mathbf{b}_i). \quad (4)$$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_c \begin{bmatrix} \mathbf{h}_{t-1} \\ \mathbf{x}_t \end{bmatrix} + \mathbf{b}_c). \quad (5)$$

Cell-state update. After that, it's time to update the old cell state to become the new cell state. With the forget vector $\mathbf{f}_t$ and the two input components $\mathbf{i}_t$ and $\tilde{\mathbf{C}}_t$ in place, we can process this step. First, the old state $\mathbf{C}_{t-1}$ is element-wise multiplied by $\mathbf{f}_t$, discarding the information we chose to forget in the previous layer, and plus with $\mathbf{i}_t \odot \tilde{\mathbf{C}}_t$. Then the new state from the result depends on how we decide to update the value for each state as mentioned in the equation 6.

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t. \quad (6)$$

Output gate. The last step is to decide what the LSTM will output at the current time step. Although the hidden output is based on the updated cell state $\mathbf{c}_t$, it passes through an additional filtering stage. First, a sigmoid-activated function determines which parts of the cell state to output in the equation 7. Then the updated cell state is passed through a tanh activation to a value in range $[-1, 1]$, and multiply it with the output sigmoid gate to gain the new hidden output in the formula 8.

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \begin{bmatrix} \mathbf{h}_{t-1} \\ \mathbf{x}_t \end{bmatrix} + \mathbf{b}_o). \quad (7)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t). \quad (8)$$

The vector $\mathbf{h}_t$ serves both as the output of the LSTM cell for the current time step and as the recurrent input to the next time step.

3.3. The proposed hybrid HMM-LSTM model

This section introduces a hybrid modeling framework designed to improve the accuracy of cryptocurrency price forecasting. Conventional LSTM models typically rely on historical closing prices to predict subsequent values. However, as demonstrated by [22], this univariate approach exhibits limited predictive power and is often inadequate for real-world financial environments characterized by volatility and non-stationarity. To enhance predictive performance, the proposed model extends the LSTM input space by incorporating additional explanatory variables that are empirically known to influence price dynamics, including the previous day's opening, high, and low prices, intraday volatility, and traded volume. In parallel, HMM is employed to identify latent market regimes and extract sequential state indicators that characterize structural shifts in market behavior. These HMM-derived state features are subsequently integrated into the LSTM network to enable regime-aware forecasting.

Algorithm 1: Regime-Aware HMM–LSTM Forecasting

Input: $\mathcal{D}$ – Time-ordered predictor and target observations
 $\mathcal{K}$ – Candidate set of HMM hidden-state numbers
 h – Forecast horizon
 w – Input sequence length
Output: $\mathcal{M}$ – Trained HMM–LSTM forecasting framework
 s^* – Selected number of hidden states

// Phase 1: Temporal partitioning
1 Partition $\mathcal{D}$ chronologically into training, validation, and test sets

// Phase 2: Hidden regime identification
2 Extract the predictor representation X_{HMM} from the training observations
3 $s^* \leftarrow \emptyset$
4 $BIC_{\min} \leftarrow +\infty$
5 **foreach** $k \in \mathcal{K}$ **do**
6 Fit an HMM with k hidden states using X_{HMM}
7 **if** the HMM converges **then**
8 Compute BIC_k
9 **if** $BIC_k < BIC_{\min}$ **then**
10 $BIC_{\min} \leftarrow BIC_k$
11 $s^* \leftarrow k$
12 **end if**
13 **end if**
14 **end foreach**
15 Fit the final HMM with s^* hidden states using the training observations
16 Infer the regime representation using the fitted HMM

// Phase 3: Regime-aware sequence construction
17 Augment the predictive representation with the inferred regime information
18 Estimate preprocessing parameters using the training observations only
19 Transform the training, validation, and test predictors using the estimated preprocessing parameters
20 Construct input sequences of length w and corresponding h -step-ahead targets

// Phase 4: Sequence forecasting
21 Initialize the LSTM forecasting model according to the predefined architecture
22 Train the LSTM using the training sequences and validation data

// Phase 5: Forecast generation and evaluation
23 Generate forecasts for the test observations
24 Compute the predefined forecasting-error metrics
25 $\mathcal{M} \leftarrow (\text{HMM, LSTM, preprocessing procedure})$
26 **return** $\langle \mathcal{M}, s^* \rangle$

The conceptual motivation for this hybrid approach stems from prior studies, such as [1], which demonstrate the effectiveness of combining HMM-driven representations with deep recurrent neural networks. Furthermore, to capture complex temporal and technical dependencies, the model incorporates a diverse set of technical indicators reflecting trend strength, momentum, and volatility. The complete feature engineering and data preprocessing procedures are detailed in Section 4.2.

The proposed framework, outlined in Algorithm 1, integrates traditional financial mathematics with advanced machine learning techniques, specifically stacked Long Short-Term Memory (LSTM) neural networks and Hidden Markov Models (HMMs). By leveraging the LSTM’s ability to process sequential dependencies alongside the HMM’s capacity for regime identification, the model effectively captures the non-linear dynamics inherent in financial time series.

In phase 1: Latent Market Regime Identification via HMM, we utilize: **GaussianHMM** initializes the Markov model with continuous Gaussian emissions; **CalculateBIC** computes the Bayesian Information Criterion score for model optimization; and **CreateMultiStepSequences** applies a sliding window technique to transform the continuous time series into supervised input-output sequences consisting of 60 historical time steps to predict h future steps. To find the optimal number of hidden states, s^* , we perform a search evaluation over the range of $k \in [2, 20]$. The case of $k = 1$ usually returns a high BIC score, as a single-state model assumes market stationarity, directly contradicting the fundamental hypothesis of regime-switching in cryptocurrency markets and the volatility of the environment. The maximum range at 20 states was established to prevent overfitting, as higher states will make model learning more complex and capture noise. In phase 2: Sequence Modeling and Training, the LSTM architectural parameters (100 and 50 hidden units, with a dropout rate of 0.2) were adopted based on established conventions in time-series forecasting. This standard configuration provides an effective balance between capturing complex temporal dependencies and mitigating the risk of overfitting.

Beyond predictive accuracy, the model emphasizes interpretability and reliability by producing simulation outputs accompanied by evaluating performance across multiple independent training runs with statistically rigorous confidence intervals. This hybrid configuration enables the dynamic estimation of drift and volatility parameters, thereby improving forecast precision and facilitating robust uncertainty quantification. Overall, the approach contributes to the development of resilient forecasting architectures that enable the digital transformation of economic and financial systems.

3.4. Performance Evaluation Metrics

Table 1 employs a diverse set of evaluation metrics. These metrics evaluate the accuracy, robustness, and proportional error characteristics of the hybrid HMM–LSTM models for predicting cryptocurrency prices.

Here N_{sample} denotes the total number of test samples. C_t and $\hat{C}_t$ represent the actual and predicted closing prices at time t , respectively. Lower values of MAPE, sMAPE, MPE, RMSE, RMSPE, MAE and Theil’s Inequality Coefficient (U2) indicate better forecasting accuracy.

3.5. Naive baseline

To analyze the model’s effectiveness in adapting to the market, we will employ a naive baseline using a persistence model. A persistence model is a basic forecasting approach that predicts future values by assuming they will remain identical to the previous day’s observation:

$$\hat{P}_{t+k|t} = P_t$$

Table 1. Summary of performance evaluation metrics used for hybrid HMM–LSTM forecasting.

Metric	Mathematical Definition
Mean Absolute Error (MAE)	$\text{MAE} = \frac{1}{N_{\text{sample}}} \sum_{t=1}^{N_{\text{sample}}} C_t - \hat{C}_t $
Mean Absolute Percentage Error (MAPE)	$\text{MAPE} = \frac{1}{N_{\text{sample}}} \sum_{t=1}^{N_{\text{sample}}} \left \frac{C_t - \hat{C}_t}{C_t} \right $
Symmetric Mean Absolute Percentage Error (sMAPE)	$\text{sMAPE} = \frac{1}{N_{\text{sample}}} \sum_{t=1}^{N_{\text{sample}}} \frac{2 C_t - \hat{C}_t }{ C_t + \hat{C}_t }$
Mean Percentage Error (MPE)	$\text{MPE} = \frac{1}{N_{\text{sample}}} \sum_{t=1}^{N_{\text{sample}}} \frac{C_t - \hat{C}_t}{C_t}$
Root Mean Squared Error (RMSE)	$\text{RMSE} = \sqrt{\frac{1}{N_{\text{sample}}} \sum_{t=1}^{N_{\text{sample}}} (C_t - \hat{C}_t)^2}$
Root Mean Squared Percentage Error (RMSPE)	$\text{RMSPE} = \sqrt{\frac{1}{N_{\text{sample}}} \sum_{t=1}^{N_{\text{sample}}} \left(\frac{C_t - \hat{C}_t}{C_t} \right)^2}$
Theil's Inequality Coefficient (U_2)	$U_2 = \frac{\sqrt{\frac{1}{N_{\text{sample}}-1} \sum_{t=2}^{N_{\text{sample}}} \left(\frac{C_t - \hat{C}_t}{C_{t-1}} \right)^2}}{\sqrt{\frac{1}{N_{\text{sample}}-1} \sum_{t=2}^{N_{\text{sample}}} \left(\frac{C_t - C_{t-1}}{C_{t-1}} \right)^2}}$

where $\hat{P}_{t+k|t}$ is the forecast at time t for the look-ahead time k , and P_t is the measurement at time t . This model performs well for short forecasting horizons, often yielding low prediction errors. As our study focuses on learning daily market changes, benchmarking our results against this persistence model will help us analyze how well our proposed model adapts to environmental volatility and whether it actually learns meaningful patterns.

4. Design and Evaluation of Benchmarking Strategies

4.1. The proposed workflow of the study

This section presents the implementation workflow of the proposed hybrid forecasting framework, as illustrated in Figure 1. The methodological pipeline consists of four main stages: data collection, feature extraction, model construction, and performance evaluation. In the first stage, cryptocurrency price series are retrieved from reliable financial data sources (Section 4.1). Next, these raw data are transformed into predictive feature sets comprising both market-derived and technical indicators (Section 4.2). In the modeling phase, the Hidden Markov Model (HMM) is employed to identify latent market regimes and generate state-dependent representations, which are subsequently utilized by a Long Short-Term Memory (LSTM) network to forecast price movements (Section 4.5). Finally, model performance is quantitatively assessed using multiple error-based metrics, includ-

ing Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Root Mean Square Percentage Error (RMSPE) (Section 5).

The proposed hybrid pipeline demonstrates the potential of combining probabilistic and deep learning methods to improve the predictive accuracy and robustness of cryptocurrency value forecasting. Furthermore, selecting an optimal model configuration and integrating complementary techniques enhances overall precision, making this framework a promising approach for data-driven financial prediction tasks.

The proposed hybrid HMM–LSTM framework is implemented through a three-stage modeling process. In the first stage, a Gaussian Hidden Markov Model (HMM) is trained on the selected feature set, comprising open, close, high, low prices, and derived technical indicators. The optimal number of hidden states, n , is determined by minimizing the Bayesian Information Criterion (BIC). The inferred hidden-state sequence, which captures latent market regimes, is subsequently appended as an additional explanatory feature.

In the second stage, all features, including the HMM-inferred states, are normalized using a Standard Scaler scaling technique and transformed into multi-step temporal sequences of length n input to enable the forecasting of h -step-ahead values. In the third stage, a multi-layer LSTM architecture, comprising stacked LSTM and Dropout layers, is trained on these sequences to exploit its capacity to model the nonlinear dependencies and long-term temporal correlations inherent in financial time series. In the multi-step forecasting mechanism, let $X_t = [x_{t-w+1}, \dots, x_t]$ represent the multivariate input sequence ending at time t , where w is the window size. Each vector x_i within this sequence contains the concatenated market features, technical indicators, and the HMM-derived state variable. The hybrid HMM-LSTM network, parameterized by its learned weights Θ , maps this input sequence to an h -step ahead prediction vector. The output layer of the network utilizes a dense, fully connected configuration to project the final hidden state into an h -dimensional space:

$$\hat{\mathbf{C}}_{t+1:t+h} = f_{LSTM}(X_t; \Theta)$$

where h denotes the defined forecast horizon, and $\hat{\mathbf{C}}_{t+1:t+h}$ is the predicted vector containing the sequential closing prices:

$$\hat{\mathbf{C}}_{t+1:t+h} = [\hat{C}_{t+1}, \hat{C}_{t+2}, \dots, \hat{C}_{t+h}]^T$$

This equation demonstrates how the final dense layer transforms the temporal representations extracted by the LSTM into a simultaneous, multi-step output of length h .

Following the typology proposed by Alexei Botchkarev [7], model performance is assessed using six widely adopted evaluation metrics: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Symmetric Mean Absolute Percentage Error (sMAPE), Mean Percentage Error (MPE), Root Mean Square Error (RMSE), and Root Mean Square Percentage Error (RMSPE). These indicators encompass both scale-dependent (MAE, RMSE) and scale-independent, percentage-based measures (MAPE, sMAPE, MPE, RMSPE), thereby providing a comprehensive and unbiased evaluation of predictive accuracy and error symmetry. The comparative results across all metrics are summarized in Table 1.

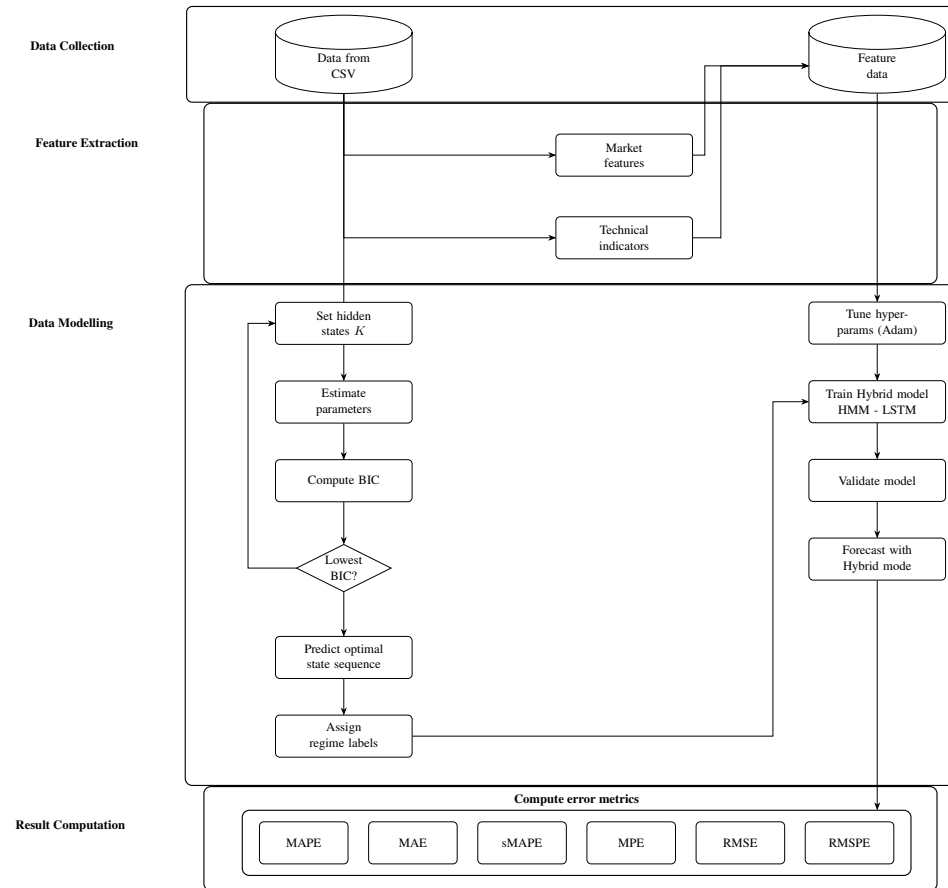


Fig. 1. Workflow of the hybrid HMM–LSTM forecasting

4.2. Collect Data and Extract Features

For dataset preparation, daily closing price series for Bitcoin (BTC–USD) and Ethereum (ETH–USD) were collected from multiple reputable financial data providers, including Bybit, Exness, and Financial Modeling Prep. These datasets encompass a continuous historical period from 17 January 2021 to 1 July 2025. All datasets were retrieved in comma-separated values (CSV) format to facilitate preprocessing and integration.

The Bitcoin dataset serves as the primary source for model development and training, while the Ethereum dataset is employed for out-of-sample validation to assess the generalization capability and robustness of the proposed hybrid model. This two-stage evaluation strategy ensures that model performance is not confined to a single cryptocurrency, thereby enhancing the reliability and external validity of the forecasting framework.

Table 2. Training, validation, and test periods for Bitcoin and Ethereum.

Asset	Train		Validation		Test	
	Start	End	Start	End	Start	End
Bitcoin	17/01/2021	28/02/2024	29/02/2024	29/10/2024	30/10/2024	01/07/2025
	1138 days (70 %)		244 days (15 %)		245 days (15 %)	
	Total: 1627 days					
Ethereum	17/01/2021	28/02/2024	29/02/2024	29/10/2024	30/10/2024	01/07/2025
	1138 days (70 %)		244 days (15 %)		245 days (15 %)	
	Total: 1627 days					

As summarized in Table 2, the dataset is partitioned into three distinct subsets: training, validation, and testing. The training subset is employed to fit the model parameters, the validation subset is used for hyperparameter tuning and model selection, and the testing subset serves to evaluate the final predictive performance. This systematic partitioning strategy minimizes overfitting and ensures an unbiased assessment of generalization capability.

The raw dataset comprises fundamental market attributes derived from the following daily fields: date, open, high, low, close, adjusted close, and trading volume. These variables collectively describe the primary market dynamics of each cryptocurrency. To enrich the feature space and capture additional market behavior not directly observable from raw prices, supplementary variables are extracted or computed from these base attributes.

All resulting features are categorized into two primary groups: (i) **market features**, representing directly observable price and volume information; and (ii) **technical indicators**, representing higher-order statistical and temporal transformations known to influence cryptocurrency price movements.

Market Features: The market features are either directly extracted from or derived using the daily price and trading volume series. In addition to the fundamental variables — open, high, low, close prices, and trading volume — two supplementary variables are computed to form a comprehensive market feature vector used in the subsequent modeling

stage. These extended features aim to capture nuanced market behaviors and improve the descriptive representation of short-term price dynamics.

Market spread. The *market spread* measures the intraday amplitude between the highest and the lowest traded prices, and is calculated in the function 9.

$$\text{Spread}_t = \text{High price}_t - \text{Low price}_t. \quad (9)$$

Daily change. The *daily change* captures the percentage movement of the closing price relative to the previous day as mentioned in the equation 10.

$$\text{Change}_t = \frac{\text{Close price}_t - \text{Close price}_{t-1}}{\text{Close price}_{t-1}}. \quad (10)$$

Technical Indicators represent quantitative transformations of historical price data designed to capture underlying market dynamics such as momentum, trend strength, and volatility behavior. In this study, each indicator is systematically derived from the closing price series and incorporated as an additional feature within the modeling framework. The selected indicators are as follows:

- **Relative Strength Index (RSI)** – A momentum oscillator used in technical analysis. RSI measures the speed and magnitude of recent price changes to identify overbought or oversold conditions in the price of a financial asset. The RSI is calculated using the following formula 11.

$$\text{RSI} = 100 - \left(\frac{100}{1 + \frac{\text{Average Gain over } n \text{ day}}{\text{Average Loss over } n \text{ day}}} \right) \quad (11)$$

RSI values range between 0 and 100. Typically, an RSI above 70 indicates that the asset is overbought, while an RSI below 30 suggests that it is oversold. The most common period used for calculation is 14 days.

- **Moving Average (MA)** – arithmetic mean of the last n closing prices, often used to smooth out price fluctuations; Given our daily sampling frequency, we compute six moving averages: MA₅, MA₂₀, MA₅₀, MA₈₈, MA₁₀₀, and MA₂₀₀, corresponding to time periods of 5, 20, 50, 88, 100, and 200 trading days, respectively. For an n -day window, the SMA is defined as mentioned in the equation 12.

$$\text{MA}_n(t) = \frac{1}{n} \sum_{i=1}^n \text{Close}_{t-i+1}. \quad (12)$$

- **Exponential Moving Average (EMA)** – an exponentially weighted moving average of the asset price, by placing exponentially decreasing weights on the price. This weighting scheme enables the EMA to respond more quickly to the most recent prices, thereby contributing the most to the average, and is therefore considered more informative for forecasting future movements. With $\mathbf{n}$ with a sequence of time, EMA can be calculated in the function 13.

$$\text{EMA}_t = \alpha \sum_{i=1}^n \text{Close}_{t-i+1} (1 - \alpha)^{i-1}, \quad (13)$$

where α is the smoothing factor that controls the rate of weight decay at each point. In line with our daily sampling frequency, we compute four exponential moving averages: EMA_5 , EMA_{20} , EMA_{50} , and EMA_{100} , corresponding to time periods of 5, 20, 50, and 100 trading days, respectively.

- **Standard Deviation (STD)** — a descriptive statistical quantity that is not too unfamiliar to us, often used to measure the dispersion level of a data set that has been tabulated into a frequency table, corresponding to time periods of 5, 20, 50, 88, 100, and 200 trading days. For a window of n trading days, the rolling STD is computed in the equation 14.

$$\text{STD}_n = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\text{Close}_i - \overline{\text{Close}})^2}. \quad (14)$$

- **Moving-Average Convergence–Divergence (MACD)**. The MACD represents the relationship between two exponential moving averages (EMAs) and is calculated by subtracting the longer-period EMA from the shorter-period EMA. This indicator helps to understand price movements in a downtrend (where the price is expected to decline) and an uptrend (where the price is expected to rise), as mentioned in the function 15.

$$\text{MACD}(t) = \text{EMA}_a(t) - \text{EMA}_b(t), \quad a < b. \quad (15)$$

Several $\langle a, b \rangle$ pairs were evaluated. Empirically, the combination $\langle a, b \rangle = \langle 5, 20 \rangle$ provided the most informative feature for the Ethereum data set, while $\langle a, b \rangle = \langle 5, 50 \rangle$ performed best for the Bitcoin data set. Therefore, both MACD series are included in the final feature matrix.

- **Bollinger Bands**. Bollinger Bands consist of two lines positioned at a specified multiple of the rolling standard deviation (STD) above and below a moving average(MA) of the price. Empirically, roughly 90% of price observations fall between the two bands; a breakout beyond either band therefore signals an unusually large move. In this study, we use the 20-day simple moving average and a two-standard-deviation band as described in the equations 16 and 17 below.

$$\text{UpperBand}_{20}(t) = \text{MA}_{20}(t) + (2 \times \text{STD}_{20}(t)), \quad (16)$$

$$\text{LowerBand}_{20}(t) = \text{MA}_{20}(t) - (2 \times \text{STD}_{20}(t)). \quad (17)$$

Closing price above the upper Bollinger band typically indicates strong upward momentum, while prices below the lower band suggest significant downward pressure.

4.3. Dataset analysis and evaluation

Summary statistics. Following feature extraction, descriptive statistics were computed for each variable, including the minimum, maximum, mean, and standard deviation (rounded to two decimal places). Table 3 presents the statistical summary of all **29 extracted features** — comprising **seven market features** based on price and volume data, and **twenty-two technical indicators** derived from moving averages, volatility measures, and momentum oscillators. These statistics provide an initial overview of the data distribution and variability within the feature set.

Correlation Analysis. Figure 2 presents the heatmap of pairwise correlation coefficients among numerical features, highlighting the strength and direction of their linear relationships. Warm tones denote strong positive correlations, while cool tones indicate strong negative correlations, offering insight into feature interdependence and potential multicollinearity within the dataset.

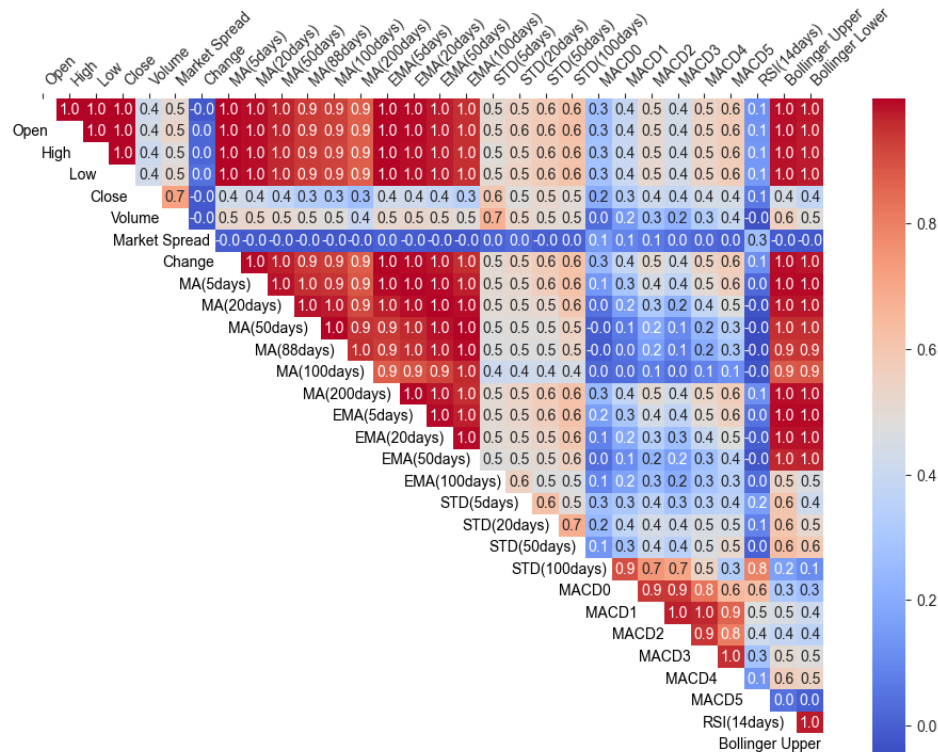


Fig. 2. Correlation Matrix of Key Predictive Variables.

Correlation between features. Understanding the interdependence among features is essential for analyzing factors that influence cryptocurrency price dynamics. Figure 2 presents

the correlation heatmap, where coefficients range from -1 to 1, indicating the strength and direction of linear relationships. The results reveal weak correlations between *close_price* and *change* ($r = 0.0$), highlighting the inherent randomness of cryptocurrency movements. The 14-day relative strength index (*rsi_14days*) exhibits a mild positive correlation with the closing price (0.1), suggesting limited momentum effects at extreme prices. Trading volume (*volume*) shows a moderate correlation with the closing price (0.4), reflecting that volume generally rises with market volatility. Meanwhile, market spread (*market_spread*) demonstrates a medium correlation (0.5), implying that widening spreads accompany price increases and signal higher transaction risk. The strong correlations observed among several other feature groups indicate potential multicollinearity, which should be carefully addressed in predictive modeling.

Forecast-model configurations. After feature extraction, each observation at time t is represented by a feature vector $\mathbf{x}_t$, as defined in Section 4.2. The LSTM model is trained to forecast the closing price at $t+1$ based on $\mathbf{x}_t$. The dataset comprises **1,776** trading days, divided into training, validation, and testing subsets to ensure robust model evaluation.

- **MA200.** Clare et al. [11] demonstrated that the 200-day moving average (MA200) can outperform buy-and-hold strategies for the S&P 500 by effectively capturing long-term market trends and mitigating short-term volatility through trend-following and stop-loss mechanisms. Conversely, Zakamulin [37] argued that many market-timing studies, including those based on moving averages, may suffer from data-mining bias and overlook transaction costs, resulting in inflated performance estimates.
- **MA88.** The 88-day moving average (MA88) serves as a mid-term technical indicator that effectively balances sensitivity to market fluctuations with noise reduction. Its ability to smooth short-term volatility while remaining responsive to emerging trends makes it well-suited for medium-horizon trading strategies. Compared to shorter averages, such as the 50-day MA, the MA88 captures broader market dynamics and provides improved stability in crossover strategies when combined with longer-term indicators, like the 200-day MA. Recent studies have highlighted the utility of moving averages in market trend analysis, underscoring their role in smoothing price data to reveal underlying trends [38].
- **Principal component analysis (PCA)** is a dimensionality reduction technique that transforms correlated variables into a set of uncorrelated principal components, thereby simplifying data complexity while retaining essential information. In financial applications, PCA helps identify key factors influencing asset volatility, thereby informing investment decisions. For instance, PCA has been applied to analyze the financial statements of over 3,000 medium and large enterprises, revealing significant factors such as central bank policy changes and short-term interest rate fluctuations. Additionally, PCA addresses multicollinearity issues in financial data, where high correlations among variables like opening, closing, high, and low prices can lead to model overfitting. Recent studies have demonstrated the efficacy of PCA in enhancing forecasting models by mitigating such challenges.
- **The Hidden Markov Model (HMM)** is a generative probabilistic model that assumes independence between observations, which can lead to redundancy in financial data. To address this, Principal Component Analysis (PCA) is employed to reduce

multicollinearity by transforming correlated features into a smaller set of orthogonal components. Eugene W. Park [28] demonstrated that combining PCA with HMM provides an intuitive understanding of market regimes through estimated emission probability distributions, yielding trading strategies with forecast accuracy slightly above 50

For cryptocurrency time series forecasting, a 60-day input window is often optimal. Qilan Lin [23] demonstrated that this window length achieved the lowest average RMSE compared to 14, 30, and 60-day windows, effectively balancing trend capture and noise reduction. This insight informs the proposed approach for selecting appropriate input days for prediction. In financial and cryptocurrency forecasting, the length of the input window has a significant impact on model prediction accuracy. Short windows may overlook crucial dependencies, while excessively long windows risk incorporating outdated patterns and overfitting. To optimize the window size for our LSTM model, we evaluated lengths of 10, 20, 30, and 90.

4.4. Data split technical for model training

This section assesses the performance of machine learning models on unseen test data. Both standard train-test splits and time series cross-validation are examined to identify the optimal dataset construction strategy. We implement K-fold time series cross-validation and adopt a 60-day input window, as recommended in recent studies for effective cryptocurrency forecasting [23].

- **Tradition train-test split (7:15:15):** The dataset was partitioned into training (75%), validation (15%), and testing (15%) subsets (Table 2). Model performance was assessed on the test set, following standard practice in time series forecasting.
- **Time Series Cross Validation:** Cross-validation is a key technique in machine learning for evaluating model performance on unseen data. K-fold cross-validation partitions the dataset into K equal folds, iteratively training on K–1 folds and testing on the remaining fold, with results averaged across iterations. In cryptocurrency forecasting, K-fold cross-validation helps optimize input window sizes, prevent overfitting, reduce data leakage, and assess model stability in recurrent neural networks (RNNs). For time series data, the standard K-Fold Cross-Validation is unsuitable because it ignores temporal dependencies. Instead, **Time Series Cross Validation (TSCV)** preserves the sequential order by ensuring the validation set always follows the training split, providing a more reliable evaluation of predictive performance [21]. Tandon [33] proposed a model for Bitcoin price prediction utilizing various neural network approaches, including Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks, in conjunction with 10-fold cross-validation. The LSTM model demonstrated superior performance, achieving a Mean Absolute Error (MAE) of 0.0043, outperforming both Random Forest and Linear Regression models. This study employs a progressive training approach, expanding the training window while shifting the validation and test windows forward in time. This method preserves temporal order and prevents data leakage, making it well-suited for financial and sequential forecasting tasks [5].

Table 3. Descriptive statistics of extracted features for Bitcoin (rounded to two decimals).

Feature	Min	Max	Mean	Median	Std. dev.
Open price	15758.96	111722.54	63740.75	48631.52	24664.64
High price	16279.99	112000.00	64140.00	49699.21	25095.06
Low price	15460.00	109217.23	62338.61	47499.06	24220.66
Close price	15760.14	111722.53	63741.33	48674.43	24703.11
Volume	5.33×10^9	3.51×10^{11}	1.78×10^{11}	3.49×10^{10}	2.22×10^{10}
RSI	6.66	99.45	53.06	52.30	17.35
Market spread	74.18	13591.70	6832.94	2200.15	1723.67
Daily change	-0.15	0.19	0.02	0.00	0.03
SMA ₅	16277.40	109118.87	62698.14	48588.56	24584.31
SMA ₂₀	16533.86	106809.28	61671.57	48268.18	24172.62
SMA ₅₀	16776.59	106075.78	61426.19	47534.73	23339.16
SMA ₈₈	17691.69	99467.69	58579.69	46582.05	22354.71
SMA ₁₀₀	17894.03	98026.07	57960.05	46298.61	22105.32
SMA ₂₀₀	15650.73	96281.71	55966.22	43876.27	20101.15
EMA ₅	16265.17	108808.95	62537.06	48588.50	24566.79
EMA ₂₀	16,758.31	106,275.72	61,517.02	48,254.28	24,092.30
EMA ₅₀	16758.31	106275.72	61517.02	48254.28	24092.30
EMA ₁₀₀	18,032.20	100,453.66	59,242.93	46,313.36	21,832.21
STD ₅	17106.07	104208.19	60657.13	47529.47	23165.82
STD ₂₀	340.43	12693.23	6516.83	3852.54	2438.16
STD ₅₀	18032.20	100453.66	59242.93	46313.36	21832.21
STD ₁₀₀	1275.30	15886.36	8580.83	5806.85	3392.42
MACD ₀	-7938.26	9311.94	686.84	334.21	2286.30
MACD ₁	-12655.56	18531.46	2937.95	1059.02	4796.91
MACD ₂	-12902.77	24828.57	5962.90	2275.14	7271.67
MACD ₃	-5910.48	9842.26	1965.89	724.81	2897.76
MACD ₄	-9016.15	17515.53	4249.69	1940.93	5780.17
MACD ₅	-4887.68	9210.09	2161.20	1216.12	3191.15
Bollinger upper	16979.62	112472.41	64726.01	52957.29	25913.75
Bollinger lower	13827.85	102111.37	57969.61	43579.06	22719.91

- **Evaluation and strategy selection.** To validate the proposed evaluation strategy, we train an LSTM model for 30 epochs and conduct 20 experimental runs on a representative market-indicator dataset (**Algorithm 2**). Two data splitting approaches are compared:
 1. **Tradition:** the dataset was divided into training, validation, and test sets at a 70:15:15 ratio.
 2. **Time Series Cross-Validation:** Experiments were conducted using K folds, where $K \in [5, 10, 15, 20, 25, 30]$. For each K , performance metrics include the mean and standard deviation across folds, as well as the minimum and maximum results to capture best- and worst-case scenarios [5].

Algorithm 2: Time-Series Cross-Validation Evaluation

Input: $\mathcal{D}$ – Time-ordered forecasting dataset
 $\mathcal{K}$ – Candidate numbers of temporal folds
 w – Input sequence length
 h – Forecast horizon
 F – Forecasting model
 $\mathcal{M}$ – Evaluation metrics
Output: Aggregated validation results for each $K \in \mathcal{K}$

```

1 foreach  $K \in \mathcal{K}$  do
2   Construct  $K$  temporally ordered training–validation folds from  $\mathcal{D}$ 
3    $\mathcal{S}_K \leftarrow \emptyset$ 
4   foreach fold  $(\mathcal{D}_k^{train}, \mathcal{D}_k^{val})$  do
5     Estimate preprocessing parameters using  $\mathcal{D}_k^{train}$  only
6     Transform  $\mathcal{D}_k^{train}$  and  $\mathcal{D}_k^{val}$  using the estimated parameters
7     Construct input sequences of length  $w$  and corresponding  $h$ -step-ahead targets
8     Train forecasting model  $F$  using the training sequences
9     Generate forecasts for the validation fold
10     $\epsilon_k \leftarrow \text{Evaluate}(\mathcal{M})$ 
11     $\mathcal{S}_K \leftarrow \mathcal{S}_K \cup \{\epsilon_k\}$ 
12  end foreach
13  Compute the minimum, maximum, mean, standard deviation, and median of the
    fold-level results in  $\mathcal{S}_K$ 
14 end foreach
15 return aggregated validation results for all  $K \in \mathcal{K}$ 

```

Table 4 summarizes the performance of each fold in the TSCV across 20 repeated experiments. For each experiment, the model predicts the validation fold, and the best and worst results correspond to the minimum and maximum error metrics across folds. As K increases from 5 to 25, the overall average error systematically decreases, with $K = 25$ achieving a mean MAE of $9,707.00 \pm 6,057.34$ and a mean RMSE of $12,590.53 \pm 7,533.39$. While splits ($K = 30$) produce slightly lower mean RMSE (12,491.56), they also exhibit higher variance (RMSE Std = 7,882.64) and severe worst-case degradation (RMSE Max = 43,045.78, $R_{\text{Min}}^2 = -1.1196$) due to brief, highly volatile market sub-periods. Therefore, $K = 25$ [or $K = 30$, reported via overall means] is selected as the

baseline configuration, offering the optimal balance between average predictive accuracy and cross-regime stability.

To evaluate the stability of the proposed framework, we compare two primary data-splitting strategies: experiments were conducted using the optimal K (selected based on the $\text{RMSE}_{\min}$) and a traditional method scheme. In the **traditional** (70:15:15), the LSTM model was trained on the training set and evaluated on the test set (indices 1356–1596), with the best model selected according to the lowest error. This direct comparison highlights the robustness and reliability of TSCV in forecasting sequential data [5].

Table 4. Summary of TimeSeriesSplit results (number of experiments = 20). An asterisk (*) marks the best K by lowest RMSE (Min).

Metric	5 Folds	10 Folds	15 Folds	20 Folds	25 Folds	30* Folds
RMSE Mean	14,901.73	13,091.60	12,873.68	12,678.43	12,590.53	12,491.56
RMSE Std	8,729.14	7,608.84	7,757.10	7,834.12	7,533.39	7,882.64
RMSE Min (Best)	4,729.44	3,764.78	3,604.73	3,430.65	2,843.04	2,803.62
RMSE Max (Worst)	29,964.81	30,778.23	35,766.56	33,773.27	41,382.97	43,045.78
R^2 Mean	0.5345	0.6188	0.5953	0.6056	0.6161	0.6096
R^2 Std	0.4191	0.3777	0.4175	0.3908	0.3914	0.4232
R^2 Max (Best)	0.9523	0.9660	0.9717	0.9754	0.9832	0.9842
R^2 Min (Worst)	-0.1761	-0.2543	-0.8847	-0.4133	-1.2467	-1.1196
MAE Mean	11,535.45	10,090.70	9,960.50	9,798.32	9,707.00	9,802.47
MAE Std	6,802.21	6,135.38	6,333.19	6,235.35	6,057.34	6,499.66
MAE Min	3,634.54	2,744.91	2,815.99	2,654.98	2,377.52	2,222.14
MAE Max	22,808.69	23,545.80	27,681.60	24,182.04	32,522.16	30,586.37

4.5. Data Modeling

As illustrated in the workflow (Figure 1), the modeling phase comprises two complementary components, each contributing to the overall predictive framework and performance evaluation.

Hidden Markov Model (HMM) Analysis. HMM captures historical price dynamics to forecast closing prices, interpreting each hidden state as a latent market regime. We model the observation distribution $P(\mathbf{y}_t \mid s_t = k)$ using a multivariate Gaussian HMM. The optimal number of hidden states is then used as input features for an LSTM network, which exploits sequential dependencies to enhance prediction accuracy and generate robust market indicators [28].

Feature-selection strategy. Leveraging HMM to capture latent market regimes, we implement a feature-grouping approach combining core market indicators and technical analysis features. Low-correlation features—such as closing price (*close_price*), trading volume (*volume*), 14-day RSI (*rsi_14days*), price change (*change*), and market spread

(*market spread*) — are selected to enhance stability and forecast accuracy (Table 5). To ensure the integrity of the feature set, we employed the Variance Inflation Factor (VIF) for multicollinearity analysis. As shown in Table 5, all selected features exhibit VIF values significantly below the common threshold of 5, with the Market Spread feature yielding the highest score at 2.34. These low values indicate minimal redundancy among the predictors, ensuring that each feature contributes unique information to the hybrid model without inflating the coefficient variance. High-correlation features are reduced via Principal Component Analysis (PCA), projecting top components to account for inter-feature dependencies, including price-volume interactions. These features, reflecting price dynamics, momentum, and volatility, are integrated with HMM-derived hidden states in a hybrid HMM-LSTM framework, evaluated using walk-forward validation to ensure robust forecasting performance [28,40].

Table 5. Variance Inflation Factor (VIF)

Feature	VIF
Close (price)	1.412897
Volume	2.093257
RSI(14days)	1.126025
Change	1.076766
Market Spread	2.343610

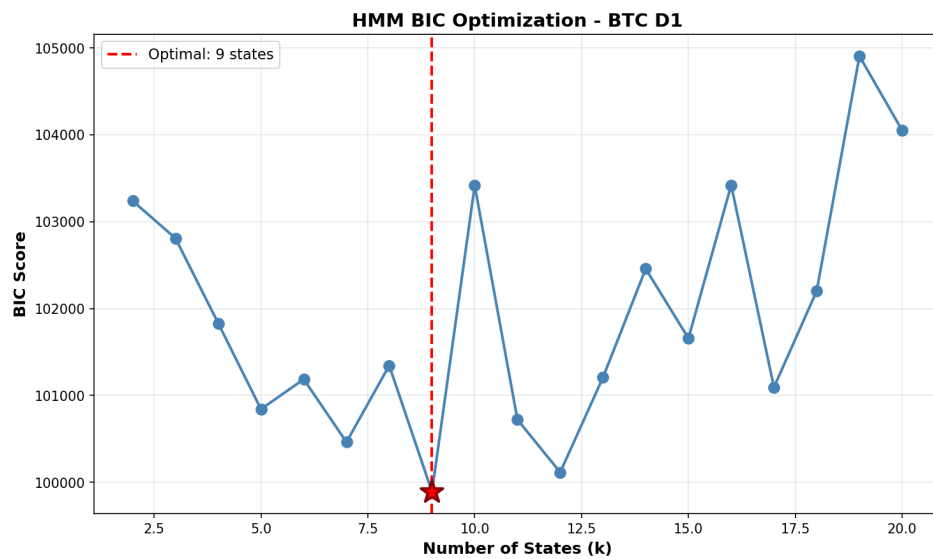


Fig. 3. Bayesian Information Criterion (BIC) for Different Hidden State Counts in the HMM.

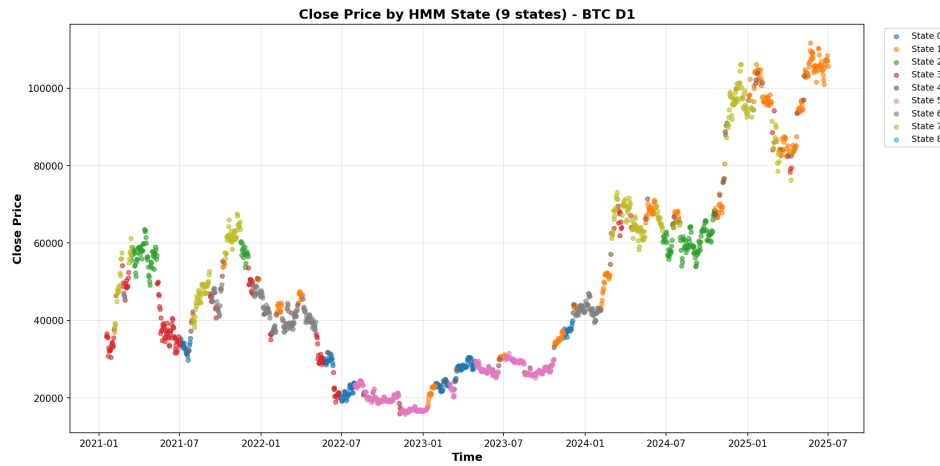


Fig. 4. HMM State Estimation for Bitcoin Price Data.

For each K , HMM parameters are estimated via maximum likelihood, and the Bayesian Information Criterion (BIC) is computed to select the optimal model. Figure 3 shows BIC values versus the number of hidden states, with the minimum indicating the optimal K . For the Bitcoin dataset, the lowest BIC occurs at **9 hidden states**, guiding the HMM configuration. This approach strikes a balance between model fit and complexity, enabling the accurate labeling of latent market regimes [28], [41]. Figure 4 shows that Bitcoin closing prices within specific ranges are typically assigned to the same hidden state, indicating a shared latent market regime. For instance, prices between \$20 000 – \$30 000 are predominantly cyan, reflecting a single regime. Increasing the number of hidden states divides the price axis into more detailed bands, creating categorical features for the forecasting model. This richer state space offers a more granular view of price movements, capturing different market behaviors such as bullish, bearish, or neutral trends. [28], [41].

LSTM Training and Feature Selection Strategy: As shown in Figure 1, once the dataset — including market variables, technical indicators, and HMM-derived state features — is prepared, an LSTM network is trained to predict next-day closing prices. The network comprises four stacked LSTM layers with a ($p = 0.2$) dropout to prevent overfitting, uses mean absolute error (MAE) as the loss function, and is optimized with Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate 10^{-3}). Training uses a batch size of 100 for 30 epochs, with early stopping set to 10 epochs on validation loss. The same setup is used on the Ethereum dataset for out-of-sample testing, showing the model’s ability to generalize. [20], [23].

Feature-selection strategy: In time series financial forecasting, datasets often include technical indicators, lag features, sentiment, and volume signals. While these features enable LSTM models to capture complex patterns, irrelevant or highly correlated features can introduce noise, increase complexity, and destabilize predictions, leading to overfitting. To mitigate these issues, we employ a Random Forest-based feature-selection strategy to identify the most informative technical features, reduce noise, and enhance

model robustness and generalization in cryptocurrency forecasting [40], [18]. In feature importance analysis for stock selection, Wang demonstrates that assessing the statistical relevance of features supports effective factor selection and improves predictive accuracy across sectors with distinct characteristics. Using a Random Forest (RF), feature importance is computed by averaging each feature's contribution across all trees, ranking them according to their influence on the target variable [9]. The dataset is split into training and test sets (80:20), with the model trained on the training set and evaluated on the test set. Features with higher importance scores are more influential and help mitigate overfitting in forecasts, as detailed in **Algorithm 3**, and are evaluated on both the training set and the test set[36], [18].

Algorithm 3: Time-Aware Random-Forest Feature Ranking

Input: X – Candidate predictor matrix
 y – Forecast target series
Output: R – Ranked feature set

```

// Step 1: Temporal alignment
1 Lag predictor variables by one observation so that only information available before the
  forecast origin is used
2 Align the lagged predictors with the target series chronologically

// Step 2: Temporal data partitioning
3 Partition the observations chronologically into training and evaluation sets without random
  shuffling

// Step 3: Random-Forest model fitting
4 Fit a Random Forest regression model using the training observations
5 Evaluate the model on the training and evaluation sets

// Step 4: Time-aware stability evaluation
6 Construct temporally ordered validation folds from the training observations
7 Evaluate the fitted forecasting relationship across the validation folds
8 Compute the corresponding validation performance measures

// Step 5: Permutation-based feature ranking
9 Estimate the importance of each predictor by measuring the change in predictive
  performance after permutation
10 Rank the predictors in descending order of their estimated importance
11  $R \leftarrow$  ranked predictor set
12 return  $R$ 

```

Figure 5 presents the Random Forest feature importance results. The 200-day moving average (MA200) is the most influential feature, while the 5-day moving average (MA5) has the lowest importance. This indicates that short-term variables, although highly correlated with the closing price, may increase the risk of overfitting and multicollinearity. To address these issues, features with importance scores below 0.04 are removed, focusing on stable variables to lower noise and improve the LSTM model's forecasting robustness. Higher importance scores indicate features with greater predictive power, aiding interpretability and feature selection. [36], [18].

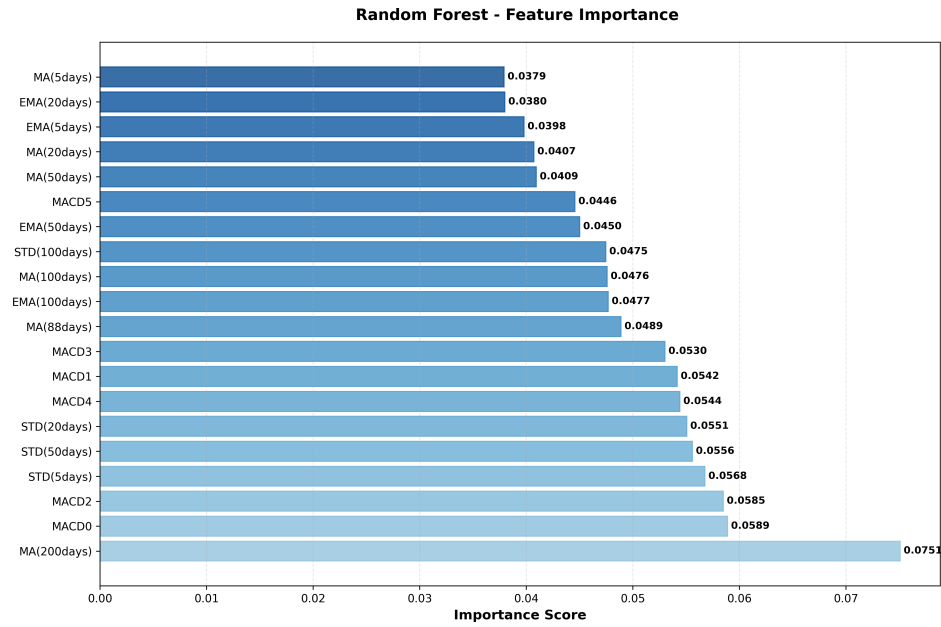


Fig. 5. Feature ranking based on importance scores obtained from the **Random Forest** model.

5. Experiment Results And Discussion

The experimental results demonstrate that the performance of the proposed hybrid HMM–LSTM framework varies across feature configurations and cryptocurrency assets. As demonstrated in Tables 6 and 8, selected hybrid configurations improve forecasting accuracy relative to standalone LSTM and HMM models in the BTC and ETH feature-specific experiments. This improvement results from the model’s ability to combine latent regime information extracted by the HMM with nonlinear temporal dependencies captured by the LSTM. The HMM identifies distinct volatility states, offering a structural understanding of market behavior, while the LSTM dynamically learns sequential patterns within those regimes. Notably, using a single volatility-sensitive indicator, especially Bollinger Bands (BB), provides better forecasting accuracy. In contrast, combining multiple indicators such as EMA, MACD, and MA causes multicollinearity and redundant information, which increases prediction variance and slightly reduces generalization. This highlights the importance of feature simplicity in volatile financial areas. Additionally, the model maintains stable inference latency (around 1.1 seconds per prediction), confirming its suitability for real-time use.

The proposed Hybrid HMM–LSTM framework provides practical value for real-time cryptocurrency trading and risk management systems. By combining regime detection with temporal forecasting, it allows traders and automated agents to predict volatility shifts and improve their entry and exit strategy decisions. The model’s low latency and

reliable performance during market turbulence make it suitable for algorithmic trading, portfolio rebalancing, and machine-to-machine (M2M) payment prediction within decentralized finance ecosystems. Additionally, its streamlined feature structure improves interpretability, supporting transparent decision-making and compliance in AI-driven financial systems, where explainability and dependability are essential for deployment in live blockchain environments.

Table 6. Comparison of single fixed holdout-test performance using the selected model instance on the Bitcoin dataset.

Model	MAE	MAPE (%)	sMAPE (%)	MPE (%)	RMSE	RMSPE (%)	Theil's U
Naive Persistence	1664.38	1.78	1.79	0.12	2324.56	2.53	1.000
LSTM	5554.00	5.65	5.85	5.49	6282.53	6.26	3.167
Hidden Markov Model (HMM)							
5 states	7127.55	7.59	7.85	2.93	9505.67	10.19	4.254
(optimal) 9 states	5227.46	5.67	5.83	1.51	7498.76	8.22	3.178
15 states	5824.37	6.23	6.45	2.21	7857.02	8.45	3.492
20 states	5732.53	6.14	6.39	2.02	8014.53	8.60	3.441
Hybrid (HMM + LSTM)							
Bollinger Bands (BB)	3972.57	4.03	4.14	3.68	4564.83	4.57	2.261
EMA	5381.80	5.46	5.64	5.27	5994.09	5.97	3.059
MA	6557.66	6.67	6.93	6.57	7171.01	7.17	3.736
MACD	5050.52	5.13	5.30	4.96	5641.94	5.65	2.877
STD	4255.31	4.32	4.44	4.08	4898.98	4.89	2.420
STD + EMA + MACD + BB	7041.09	7.13	7.44	7.10	7732.31	7.69	3.997
STD + EMA + MACD + BB + MA	9465.13	9.70	10.24	9.70	9975.43	10.08	5.437
MA + STD + EMA + MACD	9627.13	9.85	10.40	9.85	10 148.06	10.22	5.521

All models are evaluated on the train/validation/test split detailed in Table 2, with predictive performance measured using MAE, MAPE, sMAPE, MPR, MSE, RMSE, and RMSPE (Section 3.4). Comparative results in Table 6 highlight (i) the incremental value of incorporating regime information and (ii) which technical indicators provide the greatest marginal improvement when combined with market variables.

Table 6 summarizes the predictive performance of all models on the Bitcoin dataset using the evaluation metrics discussed earlier. The baseline LSTM models, trained solely on closing prices and extended feature groups, deliver moderate predictive accuracy, with the Bollinger Bands (BB) feature achieving the lowest RMSE of 6282.53. In contrast, the stand-alone HMM models demonstrate underperform performance underperform the baseline , reduces RMSE to 7498.76 with an optimal configuration of 9 hidden states. This finding confirms that capturing latent market regimes significantly enhances forecast precision [28].

The hybrid HMM–LSTM models exhibit the best overall performance when incorporating selective technical indicators. Specifically, the BB-based hybrid achieves the lowest RMSE (4,564.83), outperforming variants that utilize MA, EMA, STD, or MACD. Conversely, combining multiple indicators tends to degrade accuracy — for instance, the STD + EMA + MACD + MA combination yields an RMSE of 10,148.06 — **suggesting noise** and feature redundancy . These results indicate that combining regime-aware features

from HMM with a simple set of technical indicators improves both accuracy and stability. [28],

As detailed in Table 6, we utilize Heil’s U to compare our proposed models against the Naive baseline, which serves as a benchmark with a standard Theil’s U of 1.00. During our evaluation of Bitcoin (BTC), we observed that all proposed models yielded a Theil’s U greater than 1.00. The weaker BTC result suggests that incorporating HMM-derived states does not necessarily improve forecasting when the latent-state representation provides limited incremental information beyond the temporal patterns already learned by the LSTM. Additional analysis would be required to determine whether this result arises from regime instability, feature sensitivity, non-stationarity, or model-estimation variability. Nevertheless, among the evaluated complex models, the hybrid configuration incorporating Bollinger Bands (BB) achieves the lowest Theil’s U of 2.261.

Consequently, the findings underscore that while HMMs improve the stability of regime-sensitive forecasting, excessive feature inclusion can impair generalization due to multicollinearity and noise. The hybrid HMM–LSTM configuration incorporating Bollinger Bands emerges as the most reliable and robust model under volatile market conditions.

Table 7. Distribution of performance over 50 independent training runs under the BTC feature-ablation experimentl instance on the Bitcoin dataset

Model	RMSE (mean $\pm$ std)	RMSE 95% CI	MAE (mean $\pm$ std)	MAE 95% CI
BB	6,496.15 $\pm$ 881.98	[6,251.68, 6,740.62]	5,827.27 $\pm$ 850.96	[5,591.40, 6,063.15]
STD	6,772.23 $\pm$ 997.80	[6,495.66, 7,048.81]	6,102.07 $\pm$ 981.36	[5,830.05, 6,374.09]
MACD	7,226.34 $\pm$ 808.15	[7,002.33, 7,450.35]	6,580.18 $\pm$ 788.80	[6,361.54, 6,798.83]
EMA	7,727.18 $\pm$ 842.96	[7,493.53, 7,960.84]	7,087.23 $\pm$ 827.35	[6,857.90, 7,316.56]
MA	9,209.66 $\pm$ 1,011.77	[8,929.22, 9,490.11]	8,585.46 $\pm$ 1,002.92	[8,307.46, 8,863.46]
STD_EMA_MACD_BB	10,827.84 $\pm$ 1,635.29	[10,374.56, 11,281.12]	10,265.05 $\pm$ 1,660.64	[9,804.75, 10,725.36]
MA_STD_EMA_MACD	13,837.69 $\pm$ 1,992.61	[13,285.37, 14,390.02]	13,371.93 $\pm$ 1,992.09	[12,819.75, 13,924.11]
STD_EMA_MACD_BB_MA	13,868.85 $\pm$ 2,246.54	[13,246.14, 14,491.56]	13,393.85 $\pm$ 2,259.29	[12,767.60, 14,020.09]

Table 7 reports results from 50 repeated runs of the hybrid HMM–LSTM model across several feature configurations. Performance is summarized by the mean RMSE, standard deviation, minimum and maximum errors, the best experiment index, and average inference latency (in milliseconds). Single-indicator configurations consistently outperform multi-indicator combinations. In particular, the Bollinger Bands (BB) feature attains the lowest mean RMSE (6,496.15 $\pm$ 881.98) and the tightest 95% confidence interval [6,251.68, 6,740.62], indicating superior accuracy and stability. By contrast, full combinations (e.g., STD + EMA + MACD + BB + MA) yield substantially higher errors (mean RMSE = 13,868.85) and a larger variance ($\pm$ 2,246.54), indicating feature redundancy and an increased risk of instability. These findings support a simple feature strategy — combining HMM-derived regime information with a small set of high-value indicators (notably BB) — to optimize predictive performance while managing complexity and latency [29].

Figure 6 compares actual and predicted closing prices for the BTC test dataset using LSTM, HMM, and hybrid HMM–LSTM models with features (STD + EMA + MACD + MA, Bollinger Bands). The hybrid model exhibits the closest tracking of market movements, capturing short-term volatility while maintaining stability during sharp fluctua-

tions, thereby demonstrating improved robustness and predictive consistency over standalone models [28]. Moreover, as depicted in Figure 6, the hybrid HMM–LSTM model achieves superior alignment with actual market movements compared to the standalone LSTM and HMM architectures. This finding is consistent with the quantitative results in **Table 6 and Table 7**, where the hybrid configuration incorporating Bollinger Bands (BB) attains the lowest RMSE and MAE values, indicating enhanced predictive accuracy. The hybrid model effectively integrates **the regime identification capability** of HMM with **the temporal pattern extraction** of LSTM, allowing it to capture short-term volatility while maintaining trend stability. In contrast, the LSTM tends to overreact to local fluctuations, and the HMM underestimates sudden regime shifts. Overall, these results confirm that the hybrid approach provides a more robust and adaptive framework for forecasting in volatile cryptocurrency markets.

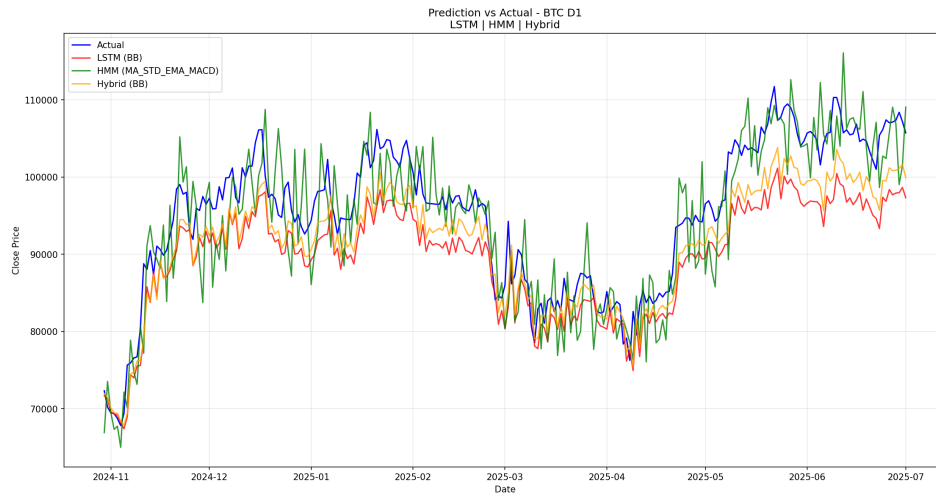


Fig. 6. Comparing test dataset BTC between features (STD+EMA+MACD+MA, Bollinger Bands) in LSTM, HMM, and Hybrid based on RMSE.

The Hidden Markov Model (HMM) effectively identifies latent market regimes, while the Long Short-Term Memory (LSTM) network captures nonlinear temporal dependencies. The hybrid integration layer merges both representations to produce robust, regime-sensitive forecasts for Ethereum price movements. To ensure methodological consistency and generalization, the experimental framework was replicated on the Ethereum dataset spanning 1,627 days, using the same training, validation, and testing configurations as described in Table 2. The HMM determined an optimal configuration with 9 hidden states, which was subsequently integrated with the LSTM for hybrid prediction.

As summarized in Table 8, the hybrid *HMM–LSTM* model combined with a single Bollinger Bands (BB) indicator achieved the lowest prediction error (MAE = 66.64, RMSE = 96.29), outperforming both the standalone LSTM and HMM models. A cross-

dataset comparison between Bitcoin and Ethereum (Tables 6 and 8) reveals consistent predictive behavior across assets. In both cases, the hybrid HMM–LSTM model, combined with a single Bollinger Bands (BB) indicator, achieved the lowest RMSE (BTC = 4564.84; ETH = 96.29), outperforming standalone LSTM and HMM baselines. The BB feature proved most effective in capturing short-term volatility and regime shifts. Conversely, multi-indicator configurations degraded accuracy, likely due to correlated or redundant inputs that introduced noise and overfitting. These findings confirm the robustness and generalizability of regime-aware hybrid frameworks across distinct cryptocurrencies, supporting prior studies emphasizing parsimonious deep-learning architectures for volatile financial markets [3], [19].

Conversely, our analysis of the Ethereum (ETH) dataset yielded highly favourable results. For ETH, the majority of our proposed models successfully outperformed the Naive baseline. Specifically, the BB-featured hybrid model achieved a Theil's U of 0.969 (< 1.00), demonstrating the model's capability to effectively learn market dynamics and surpass simple persistence forecasting under standard volatility conditions.

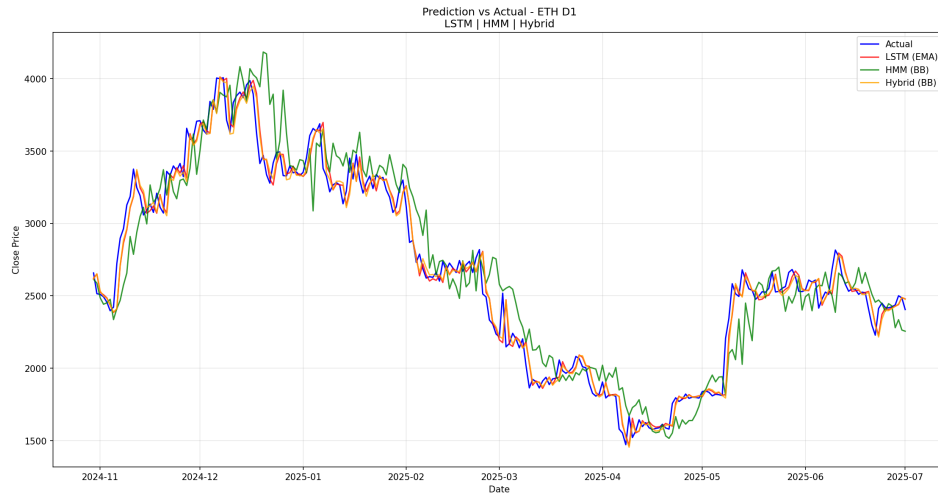
Table 8. Comparison of model performance on the Ethereum dataset.

Model	MAE	MAPE (%)	sMAPE (%)	MPE (%)	RMSE	RMSPE (%)	Theil's U
Naive Persistence Baseline	75.35	2.85	2.85	−0.12	106.72	4.09	1.000
LSTM (best feature set)	74.42	2.80	2.81	0.05	106.01	4.03	0.985
Hidden Markov Model (HMM)							
5 states	325.22	12.56	12.22	−3.33	404.14	15.94	4.411
9 states	343.87	13.08	12.88	−2.57	420.13	16.05	4.594
15 states	320.97	12.15	12.13	−1.31	400.33	15.04	4.267
(optimal) 20 states	230.89	8.79	8.86	−0.05	287.62	10.69	3.087
Hybrid (HMM + LSTM)							
Bollinger Bands (BB)	66.64	2.76	2.76	−0.03	96.29	4.06	0.969
EMA	67.20	2.77	2.78	0.02	97.17	4.08	0.973
MA	68.62	2.81	2.82	0.20	97.93	4.09	0.988
MACD	68.21	2.81	2.83	0.18	96.93	4.05	0.988
STD	70.08	2.90	2.89	−0.67	98.68	4.14	1.019
STD + EMA + MACD + BB	68.24	2.81	2.82	0.14	96.37	4.04	0.99
STD + EMA + MACD + BB + MA	70.35	2.94	2.93	−0.71	98.28	4.17	1.03
MA + STD + EMA + MACD	71.99	3.04	3.02	−0.82	97.02	4.15	1.07

Table 9 summarizes the performance of various feature configurations for the hybrid HMM–LSTM model on the Ethereum dataset, averaged over 20 repeated runs. Performance is summarized by the mean RMSE, standard deviation, minimum and maximum errors, the best experiment index, and average inference latency (in milliseconds). Single-indicator configurations consistently outperform multi-indicator combinations. In particular, the MACD feature attains the lowest mean RMSE (98.15 ± 0.53), narrowly outperforming the BB model (98.55 ± 1.04). By contrast, full combinations (e.g., STD + EMA + MACD + BB + MA) yield substantially higher errors (mean RMSE = 13,440.72) and a larger variance ($\pm 2,246.54$), indicating feature redundancy and an increased risk of instability. However, the BB configuration maintained the lowest Mean Absolute Error (68.74 ± 1.19) compared to MACD (70.09 ± 1.39) [29].

Table 9. Distribution of performance over 20 independent training runs under the ETH feature-ablation experiment.

Model	RMSE (mean $\pm$ std)	RMSE 95% CI	MAE (mean $\pm$ std)	MAE 95% CI
MACD	98.15 $\pm$ 0.53	[97.92, 98.38]	70.09 $\pm$ 1.39	[69.48, 70.71]
BB	98.55 $\pm$ 1.04	[98.09, 99.00]	68.74 $\pm$ 1.19	[68.22, 69.27]
EMA	98.73 $\pm$ 0.74	[98.41, 99.05]	69.13 $\pm$ 0.94	[68.72, 69.54]
STD_EMA_MACD_BB	98.81 $\pm$ 1.66	[98.08, 99.54]	71.99 $\pm$ 2.70	[70.81, 73.18]
MA	99.82 $\pm$ 1.32	[99.24, 100.40]	70.39 $\pm$ 1.28	[69.83, 70.95]
STD_EMA_MACD_BB_MA	101.27 $\pm$ 2.18	[100.31, 102.22]	75.38 $\pm$ 3.25	[73.96, 76.81]
MA_STD_EMA_MACD	101.75 $\pm$ 2.00	[100.87, 102.62]	75.81 $\pm$ 2.39	[74.77, 76.86]
STD	102.42 $\pm$ 2.49	[101.33, 103.52]	74.45 $\pm$ 2.51	[73.35, 75.55]

**Fig. 7.** Comparing test dataset ETH between features (EMA, Bollinger Bands) in LSTM, HMM, and Hybrid based on RMSE.

Tables 7 and 9 present feature configuration performance for Bitcoin and Ethereum using the hybrid HMM–LSTM model. For Bitcoin, the Bollinger Bands (BB) indicator achieves the lowest mean error ($6,496.15 \pm 881.98$) with minimal deviation. For Ethereum, single-indicator features remain the most effective configurations, with MACD achieving the lowest mean error (98.15 ± 0.92), closely followed by BB (98.55 ± 1.04), and minimal deviation, confirming its superior predictive stability. In contrast, multi-indicator combinations, such as STD + EMA + MACD + MA, yield higher errors and variance, indicating overfitting and redundant feature interactions. The average inference latency remains stable at approximately 1.10 seconds per prediction, demonstrating computational efficiency. The consistent dominance of BB across assets highlights its sensitiv-

ity to volatility and short-term market shifts, supporting prior findings in volatility-driven forecasting models [29].

Figure 7 further demonstrates that the proposed hybrid model closely tracks actual market trends, effectively capturing short-term fluctuations and maintaining predictive stability during volatile market phases. Figures 6 and 7 illustrate the comparative forecasting performance of the proposed hybrid HMM–LSTM model against standalone LSTM and HMM baselines for Bitcoin and Ethereum, respectively. In both datasets, the hybrid configuration employing Bollinger Bands (BB) most closely follows actual market movements, effectively capturing rapid fluctuations and trend reversals. The model demonstrates strong adaptability to dynamic volatility regimes, maintaining predictive stability across distinct market phases. The similar tracking accuracy observed in both cryptocurrencies highlights the hybrid model’s robustness and transferability across heterogeneous financial assets, reinforcing its regime-aware learning capability, as supported by prior empirical studies [19], [3].

Figures 6, 7, and 8 collectively evaluate the predictive behavior and training stability of the proposed hybrid HMM–LSTM model on Bitcoin (BTC) and Ethereum (ETH) datasets. Figures 6 and 7 show that the hybrid configuration, particularly when combined with Bollinger Bands (BB), most accurately tracks actual market prices, effectively capturing high-frequency volatility while minimizing forecast deviation compared to standalone LSTM and HMM models. This consistency across BTC and ETH underscores the model’s generalization capacity and adaptability to heterogeneous market dynamics. Figure 8 further validates model convergence, with both training and validation losses exhibiting rapid stabilization after approximately 50 epochs, indicating robust optimization without overfitting. Notably, ETH demonstrates marginally lower validation loss variability, suggesting smoother learning dynamics. These findings confirm the reliability, predictive robustness, and computational efficiency of the regime-aware hybrid model for forecasting volatile cryptocurrencies [19], [3].

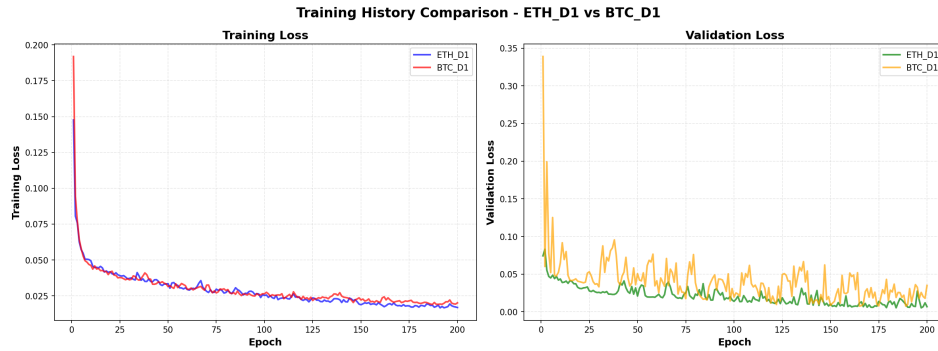


Fig. 8. Hybrid HMM–LSTM model trained on BTC and ETH using Bollinger Bands.

Figure 9 presents the runtime efficiency and hardware resource utilization of the proposed hybrid HMM–LSTM model trained on BTC and ETH datasets. The CPU utilization

indicates that BTC training exhibits greater fluctuation ($\approx 25\text{--}55\%$) compared to ETH ($\approx 10\text{--}25\%$), reflecting the higher computational demand caused by increased volatility and longer temporal dependencies. Memory consumption remains stable for both datasets, with BTC slightly exceeding ETH toward the final epochs due to more complex state transitions. Latency remains consistent at approximately 1.1 s per inference, with minor spikes observed near training completion, likely due to checkpoint synchronization. The performance distribution histogram demonstrates balanced resource usage and stable throughput, confirming the model's scalability and efficiency. Overall, these findings affirm that the hybrid HMM–LSTM framework delivers accurate and regime-sensitive forecasts while maintaining computational stability and efficient system utilization during both training and inference phases [19], [3].

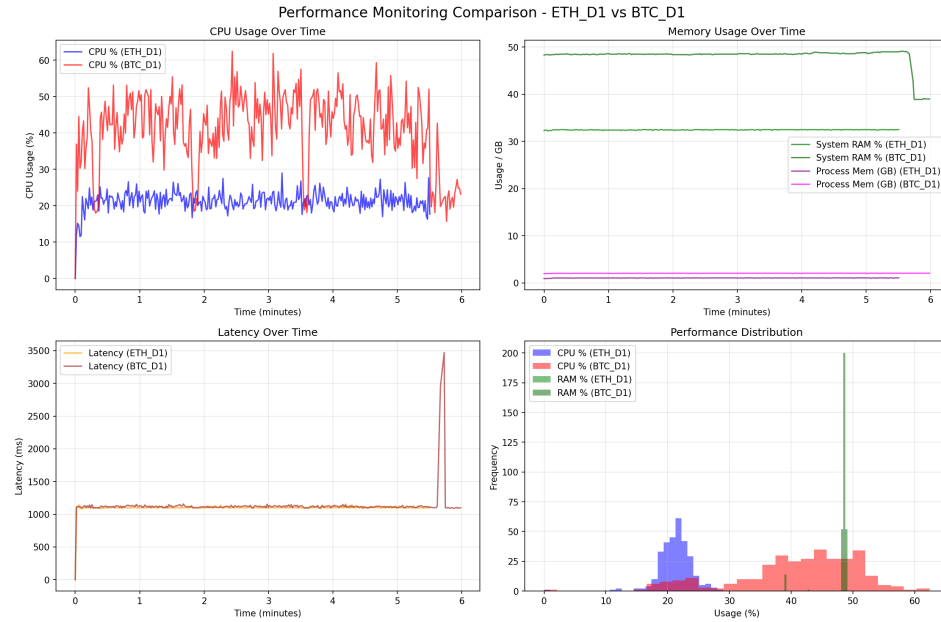


Fig. 9. Performance monitoring of the best hybrid HMM–LSTM model for BTC and ETH using the Bollinger Bands feature.

For further analysis, we have compared the hybrid model with other models like DeepAR [16] and ARIMA [30] across four different cryptocurrency types: Bitcoin(BTC), Ethereum(ETH), Binance Coin(BNB) and Solana(SOL). Table 10 shows the comprehensive predictive performance of the evaluated models across four cryptocurrency assets (BTC, ETH, BNB, and SOL). The evaluation framework compares our proposed hybrid LSTM-HMM architecture against traditional statistical methods (ARIMA), state-of-the-art probabilistic deep learning (DeepAR), and the standalone components of our architecture (LSTM and HMM) using RMSE, expressed as the mean $\pm$ standard deviation and a

95% confidence interval. For BNB and SOL, which have smaller volumes and volatility, the model performance may be weaker compared to the others. As observed in the results, we recommend the HMM-LSTM Hybrid as the primary model for deployment, with the standalone LSTM as a strong fallback baseline where lower complexity is preferred.

Table 10. Cross-asset benchmark under a common model-comparison protocol across ETH,BTC,SOL,and BNB using RMSE(mean $\pm$ std) and 95%CI.

Asset Metric		LSTM	HMM	ARIMA	DeepAR	HMM + LSTM Hybrid
ETH	Mean $\pm$ Std	107.67 $\pm$ 1.58	406.09 $\pm$ 58.35	746.24 $\pm$ 65.39	101.92 $\pm$ 1.15	99.94 $\pm$ 2.23
	95% CI	[106.57, 108.76]	[387.03, 425.15]	[700.93, 791.56]	[101.12, 102.72]	[99.59, 100.28]
BTC	Mean $\pm$ Std	8,153.96 $\pm$ 1,696.28	9,459.47 $\pm$ 1,468.25	20,454.46 $\pm$ 3,720.74	17,711.28 $\pm$ 2,248.11	9,495.74 $\pm$ 3,167.45
	95% CI	[6,978.49, 9,329.42]	[8,979.84, 9,939.10]	[17,876.12, 23,032.80]	[16,153.41, 19,269.14]	[9,185.33, 9,806.15]
SOL	Mean $\pm$ Std	8.56 $\pm$ 0.10	33.25 $\pm$ 4.26	39.97 $\pm$ 0.70	9.24 $\pm$ 0.33	8.56 $\pm$ 0.31
	95% CI	[8.49, 8.63]	[31.86, 34.64]	[39.49, 40.45]	[9.01, 9.47]	[8.51, 8.61]
BNB	Mean $\pm$ Std	16.98 $\pm$ 0.97	52.74 $\pm$ 10.71	52.60 $\pm$ 3.24	23.13 $\pm$ 2.73	15.87 $\pm$ 1.34
	95% CI	[16.31, 17.65]	[49.24, 56.24]	[50.36, 54.85]	[21.23, 25.02]	[15.66, 16.07]

The results indicate that the value of HMM-derived regime information is conditional on the asset, feature specification, and evaluation setting. Under the BTC and ETH feature-ablation experiments, selected single-indicator hybrid configurations often improve upon standalone HMM and LSTM models. However, the broader cross-asset comparison in Table 10 reveals heterogeneous performance. The hybrid model outperforms standalone LSTM for ETH and BNB, performs comparably for SOL, but underperforms LSTM for BTC. Accordingly, the empirical evidence supports an asset-dependent benefit of regime-aware fusion rather than universal superiority of the hybrid architecture.

Overall, these findings indicate that the regime-aware hybrid architecture provides conditional utility balancing model complexity and interpretability where market regimes are distinctly separated, rather than functioning as an unconditionally superior approach across all digital assets. These findings align with recent studies advocating for hybrid deep learning and probabilistic integration to improve financial time series modeling [28],[19].

6. Conclusion

This study addressed the critical challenge of forecasting cryptocurrency price behavior in highly volatile markets, emphasizing the influence of financial, technical, and macroeconomic factors on predictive performance. The primary objective was to develop a **hybrid HMM–LSTM framework** capable of modeling latent market regimes and temporal dependencies in cryptocurrency price movements, with a specific focus on Bitcoin—the most actively traded and valuable blockchain asset.

A hybrid HMM–LSTM framework is introduced for predicting cryptocurrency prices. The model identifies hidden market regimes and nonlinear temporal relationships. Bollinger Bands (BB) enhance prediction accuracy during volatile market periods. Comparative results demonstrate better performance than standalone LSTM and HMM models. The framework provides scalable, regime-aware predictions for blockchain-based finance.

The empirical results demonstrate that integrating HMM-derived regime information with LSTM forecasting can improve predictive accuracy, but the magnitude and direction of the improvement depend on the cryptocurrency and feature configuration. The hybrid model performs favourably for several asset–model combinations, particularly ETH and BNB, while standalone LSTM remains superior for BTC in the broader cross-asset benchmark. The results therefore support HMM–LSTM integration as a potentially useful regime-aware forecasting strategy rather than a universally superior alternative to standalone deep-learning models.

Moreover, feature selection plays a crucial role in predictive stability. The results reveal that using a single Bollinger Bands (BB) indicator, combined with HMM regime states and core market variables, yields the most consistent and accurate outcomes. Conversely, including multiple correlated indicators (e.g., EMA, MACD, MA, STD) leads to diminishing returns due to multicollinearity and feature redundancy. These findings align with recent research emphasizing the benefits of regime-aware hybrid frameworks that balance model complexity and generalization capability [19], [3].

Performance evaluations across multiple cryptocurrency assets indicate that the framework can generalize across markets, although its relative forecasting advantage varies by asset and model configuration. Training histories exhibit smooth optimization curves and stable validation losses after approximately 50 epochs, while performance monitoring reveals efficient memory usage and moderate CPU load, supporting the model’s scalability for real-time financial forecasting applications.

Nevertheless, certain limitations remain. The model primarily relies on historical market data and does not incorporate real-time sentiment, blockchain transaction data, or macroeconomic indicators, which can significantly influence price dynamics. In addition, because the HMM is trained on a feature set that includes non-stationary price-level information (e.g., closing price and level-based technical indicators), the inferred hidden states may partially reflect price-range clustering rather. Consequently, the interpretation of these latent states as “market regimes” should be made with caution.

Future research should integrate these external data sources—such as news sentiment, on-chain analytics, and social media trends—to enhance model responsiveness to sudden market events and re-estimate the HMM using stationary regime descriptors (e.g., bull/bear conditions or high/low volatility) that are invariant to the absolute price scale. Additionally, expanding this framework to other emerging cryptocurrencies, including IoT-oriented assets like IOTA, may further validate its generalization capability.

In summary, the proposed **HMM–LSTM hybrid model** offers a robust and interpretable approach for cryptocurrency price forecasting under volatile market conditions. By effectively combining regime detection with deep sequence modeling, it advances the predictive precision and practical relevance of AI-driven financial analytics, paving the way for more adaptive and intelligent trading decision-support systems.

Acknowledgment. This research is partly funded by Ho Chi Minh City University of Technology and Engineering (HCM-UTE), Vietnam.

Conflict of Interest. The authors declare that there is no conflict of interest.

References

1. Abu Hashish, I., Forni, F., Andreotti, G., Facchinetti, T.: A hybrid model for bitcoin price prediction using hidden markov models and optimised lstm networks. In: 2019 International Conference on ... (2019)
2. Avinash, G., Ramasubramanian, V., Ray, M., Paul, R.K., Godara, S., Nayak, G.H., Kumar, R.R., Manjunatha, B., Dahiya, S., Iquebal, M.A.: Hidden markov guided deep learning models for forecasting highly volatile agricultural commodity prices. *Applied Soft Computing* 158, 111557 (2024)
3. Avinash, G., Ramasubramanian, V., Ray, M., Paul, R.K., Godara, S., Nayak, G.H., Kumar, R.R., Manjunatha, B., Dahiya, S., Iquebal, M.A.: Hidden markov guided deep learning models for forecasting highly volatile agricultural commodity prices. *Applied soft computing* 158, 111557 (2024)
4. Awad, M., Khanna, R.: Hidden Markov Model, pp. 81–104. Apress, Berkeley, CA (2015), accessed from ResearchGate
5. Bergmeir, C., Benítez, J.M.: On the use of cross-validation for time series predictor evaluation. *Information Sciences* 191, 192–213 (2012)
6. Biswas, K., Muthukkumarasamy, V.: Securing smart cities using blockchain technology. In: 18th IEEE Int'l Conf. on High Performance Computing and Communications (HPCC) / SmartCity / DSS. pp. 1392–1393. Sydney, Australia (2016)
7. Botchkarev, A.: A new typology design of performance metrics to measure errors in machine learning regression algorithms. *Interdisciplinary Journal of Information, Knowledge, and Management* 14, 45–79 (2019)
8. Bouteska, A.: Comparative analysis of boosting ensembles, recurrent neural networks, and hybrid arima–rnn methods for cryptocurrency price prediction. Unpublished manuscript (2024), empirical evaluation on Bitcoin, Ethereum, Litecoin and more
9. Breiman, L.: Random forests. *Machine Learning* 45(1), 5–32 (2001)
10. Busari, G.A., Lim, D.H.: Crude oil price prediction: A comparison between adaboost-lstm and adaboost-gru for improving forecasting performance. *Computers and Chemical Engineering* 155, 107513 (2021)
11. Clare, A., Seaton, J., Smith, P.N., Thomas, S.: Breaking into the blackbox: Trend following, stop losses and the frequency of trading – the case of the S&P500. *Journal of Asset Management* 14(3), 182–194 (2013)
12. Conoscenti, M., Vetro, A., De Martin, J.C.: Blockchain for the internet of things: A systematic literature review. In: 13th IEEE/ACS Int'l Conf. on Computer Systems and Applications (AICCSA). pp. 1–6. Agadir, Morocco (2016)
13. Eggen, S., Espe, T.J., Grude, K., Ristad, M., Sandberg, R.: Financial time series uncertainty: A review of probabilistic ai applications. *Journal of Economic Surveys* (2025)
14. Erlwein-Sayer, C., Grimm, S., Pieper, A., Alsac, R.: Forecasting corporate credit spreads: Regime-switching in lstm. *SSRN Electronic Journal* (2021), available at SSRN: <http://dx.doi.org/10.2139/ssrn.4003338>
15. Fakhri, D., Mutijarsa, K.: Secure iot communication using blockchain technology. In: 2018 International Symposium on Electronics and Smart Devices (ISESD). pp. 1–6. Bandung, Indonesia (2018)
16. Ferenczi, A., Bădică, C.: Prediction of ethereum gas prices using deepar and probabilistic forecasting. *Journal of Information and Telecommunication* 7(4), 449–467 (2023)
17. Gurgul, V., Lessmann, S., Härdle, W.K.: Deep learning and nlp in cryptocurrency forecasting: Integrating financial, blockchain, and social media data. *International Journal of Forecasting* (2025)
18. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of machine learning research* 3(Mar), 1157–1182 (2003)

19. Henrique, B.M., Sobreiro, V.A., Kimura, H.: Practical machine learning: Forecasting daily financial markets directions. *Expert Systems with Applications* 233, 120840 (2023)
20. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Computation* 9(8), 1735–1780 (Nov 1997), introduces LSTM with constant-error carousel (CEC) units; empirical evaluation mentions cryptocurrency time-series like Bitcoin, Ethereum, Litecoin (see metrics on IEEE Xplore)
21. Hyndman, R.J., Athanasopoulos, G.: *Forecasting: principles and practice*. OTexts (2018)
22. Lima, A.L.: Bitcoin price prediction using recurrent neural networks and lstm. Project report (2021)
23. Lin, Q.: Price prediction of bitcoin using lstm neural network. *BCP Business & Management EMFRM* 38(4183), 2738–2744 (2023)
24. Murúa, C., Marín, M., Cofré, A., Wuth, J., Pino, O.V., Yoma, N.B.: An end-to-end dnn-hmm based system with duration modeling for robust earthquake detection. *Computers and Geosciences* 179, 105434 (2023)
25. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008), <https://bitcoin.org/bitcoin.pdf>, white paper
26. Neagoe, A., Tică, E.I., Vuță, L.I., Nedelcu, O., Dumitran, G.E., Popa, B.: Hybrid lstm-arima model for improving multi-step inflow forecasting in a reservoir. *Water* 17(21) (2025)
27. Olorunnimbe, K., Viktor, H.: Deep learning in the stock market—a systematic survey of practice, backtesting, and applications. *Artificial Intelligence Review* 56(3), 2057–2109 (2023)
28. Park, E.: Principal component analysis and hidden markov model for forecasting stock returns. arXiv preprint arXiv:2307.00459 (2023)
29. Petrozziello, A., Troiano, L., Serra, A., Jordanov, I., Storti, G., Tagliaferri, R., La Rocca, M.: Deep learning for volatility forecasting in asset management. *Soft Computing* 26(17), 8553–8574 (2022)
30. Rizvi, M.F., Sahu, S., Rana, S.: Arima model time series forecasting. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)* 12(5), 3782–3785 (2024), <https://www.ijraset.com/best-journal/arima-model-time-series-forecasting>
31. Ryou, H., Bae, H.H., Lee, H.S., Oh, K.J.: Momentum investment strategy using a hidden markov model. *Sustainability* (2020), publisher Copyright: © 2020 by the authors
32. Shah, J., Vaidya, D., Shah, M.: A comprehensive review on multiple hybrid deep learning approaches for stock prediction. *Intelligent Systems with Applications* 16, 200111 (2022)
33. Tandon, S., Tripathi, S., Saraswat, P., Dabas, C.: Bitcoin price forecasting using lstm and 10-fold cross validation. In: 2019 International Conference on Signal Processing and Communication (ICSC). pp. 323–328 (2019)
34. Tran, T.K., Le, T.T.T., Bui, T.T., Dang, V.Q., Šenkeřík, R.: Constructing a cryptocurrency-price prediction model using deep learning. In: *Proceedings of the 8th International Conference on Engineering and Emerging Technologies (ICEET 2022)*. pp. 1–6. IEEE, Kuala Lumpur, Malaysia (2022), presents a deep learning-based approach for cryptocurrency price prediction; evaluated on multiple crypto datasets
35. Vancsura, L., Tatay, T., Bareith, T.: Navigating ai-driven financial forecasting: A systematic review of current status and critical research gaps. *Forecasting* 7(3) (2025)
36. Wang, Y.: Analysis of feature importance based on random forest for stock selection. In: *Proceedings of the 1st International Conference on E-commerce and Artificial Intelligence (ECAI 2024)*. pp. 374–379. SCITEPRESS, London, U.K. (2024)
37. Zakamulin, V.: The real-life performance of market timing with moving average and time-series momentum rules. *Journal of Asset Management* 15(4), 261–278 (2014)
38. Zakamulin, V., Giner, J.: Optimal trend-following with transaction costs. *International Review of Financial Analysis* 90, 102928 (2023)
39. Zhang, Y., Wen, J.: The iot electric business model: Using blockchain technology for the internet of things. *Peer-to-Peer Networking and Applications* 10(4), 983–994 (2017)

40. Zhu, Z., Wang, S., Wang, F., Tu, Z.: Recommendation networks of homogeneous products on an e-commerce platform: Measurement and competition effects. *Expert Systems with Applications* 201, 117128 (2022)
41. Zucchini, W., MacDonald, I.L.: *Hidden Markov models for time series: an introduction using R*. Chapman and Hall/CRC (2009)

Tran Kim Toai earned a PhD from the Department of Computer Science at VSB– Technical University of Ostrava, Czech Republic. He works as a lecturer at the Ho Chi Minh City University of Technology and Engineering (HCM-UTE). His current academic research interests are emerging technologies such as machine learning, artificial intelligence, data mining, neural networks, data analysis, ESG, and finance.

Phan Nhat Minh is a researcher working in the fields of data science and artificial intelligence. His research focuses on applying data-driven and intelligent methods to real world problems, particularly in prediction, pattern recognition, and decision-making. His current research interests include machine learning, deep learning, time-series forecasting, data mining, and financial data analysis.

Received: November 16, 2025; Accepted: August 23, 2026.

