

AI model for stock price prediction: hybrid neural network and deep learning

Abstract—This study develops a hybrid time-series prediction framework that combines Moving Average (MA) preprocessing with Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) networks. The central motivation is to improve prediction accuracy for rapidly changing sequential data by separating systematic temporal patterns from residual fluctuations and then learning both short- and long-range dependencies with recurrent deep-learning components. The workflow begins with removal of missing observations, MA-based transformation and decomposition, Random Forest-based feature selection, and data scaling. Technical indicators are subsequently added before the dataset is divided into training/testing and development subsets. The development subset is used for hyperparameter tuning under a mean squared error (MSE) acceptance criterion, while the final GRU–LSTM hybrid produces the prediction output. The empirical dataset contains approximately 8,600 hourly observations collected during working hours from a main company and three sub-companies in Hanoi, Vietnam, over 3.5 years from 2022 to the first half of 2025. Reported results show that the proposed model achieves an MSE of 1,265 and a mean absolute percentage error (MAPE) of 4.37%, outperforming the reproduced GRU and LSTM baselines and substantially improving over the MA benchmark. These findings indicate that combining decomposition-oriented preprocessing with complementary recurrent architectures can provide a practical and computationally feasible approach to nonlinear time-series prediction.

Keywords—time-series prediction; Moving Average; GRU; LSTM; hybrid neural network; technical indicators; MSE; MAPE.

1. Introduction

Time-series prediction is a central problem in data-driven decision making because many economic and operational variables evolve rapidly, exhibit nonlinear behavior, and respond to both short-term shocks and longer-term patterns. In stock-market applications, these properties become especially important: price movements can be abrupt and difficult to model, yet they also contain recurring structures that are useful for investors, traders, firms, and policy makers. The supplied experimental material frames stock-market dynamics as an important indicator of economic strength and development and argues that more accurate modeling of price fluctuations can support decisions under uncertainty by revealing meaningful patterns and trends.

Traditional forecasting approaches are often effective when the data-generating process is comparatively stable and linear, but their performance can deteriorate when the series contains nonlinear interactions and long temporal dependencies. Recurrent Neural Networks (RNNs) and their gated variants are therefore widely used for sequential prediction. In particular, LSTM networks are designed to preserve relevant information across long sequences through a structured memory mechanism, whereas GRU networks use a more compact gating structure that can reduce computational complexity while maintaining competitive predictive capability. The supplied study adopts these two architectures as complementary components of a single hybrid model rather than treating them as isolated alternatives.

The main research gap addressed here is the limited use of a complete preprocessing and technical-analysis pipeline together with recurrent learning mechanisms that can represent both short- and long-term dependencies. The proposed framework seeks to improve predictive efficiency and accuracy by combining four ideas: data cleaning, MA-based transformation/decomposition, feature selection and scaling, and a sequential GRU–LSTM architecture. The method is evaluated using MSE and MAPE and is compared with MA, GRU, and LSTM benchmarks under the same experimental setting.

Accordingly, this paper has three objectives. First, it formalizes the preprocessing and model-building workflow contained in the supplied experimental design. Second, it reports and interprets the performance of the proposed hybrid architecture using the available dataset and benchmark results. Third, it identifies practical limitations and future extensions, including richer multi-input data, stronger neural architectures, optimization procedures, and more capable hardware. The remainder of the paper is organized as follows: Section 2 presents the methodology, Section 3 describes the data and empirical results, and Section 4 concludes with directions for future work.

2. Methodology

2.1 Overall research workflow

The proposed methodology follows a staged pipeline that converts historical observations into a preprocessed time series, enriches the resulting data with technical indicators, and then learns the predictive mapping through a GRU–LSTM hybrid network. Figure 1 summarizes the preprocessing flow. Missing observations are removed first. The remaining data are transformed into a time-series representation using MA-based processing, relevant features are selected through Random Forest, and the retained variables are scaled before model training. This sequence is intended to reduce noise, standardize the learning inputs, and prevent less informative variables from dominating the recurrent model.

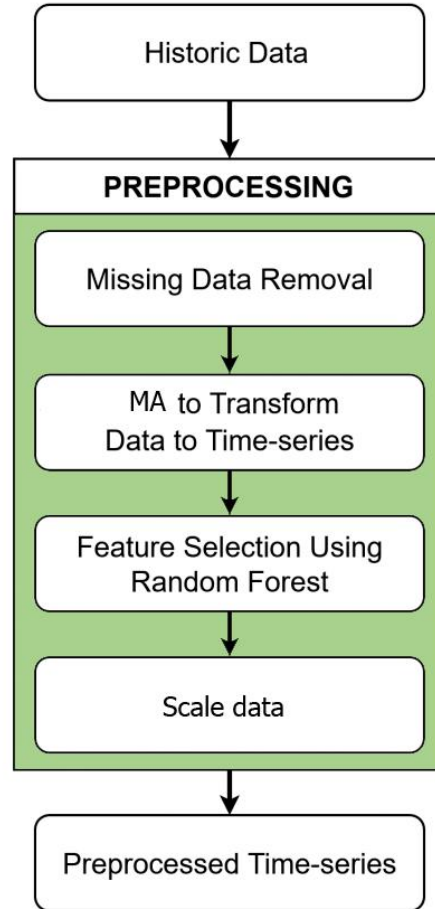


Figure 1. Data preprocessing procedure used in the proposed framework.

After preprocessing, technical indicators are added and the dataset is split into two functional parts. The train/test subset is used to fit and evaluate the hybrid model, whereas a development subset is reserved for hyperparameter tuning. Tuning is repeated until the resulting MSE is considered acceptable; once this condition is satisfied, the selected configuration is passed to the GRU–LSTM model and the final output is generated. The supplied material does not enumerate the specific technical indicators, so the present paper preserves this stage at the same level of detail rather than introducing additional indicators not documented in the experiment.

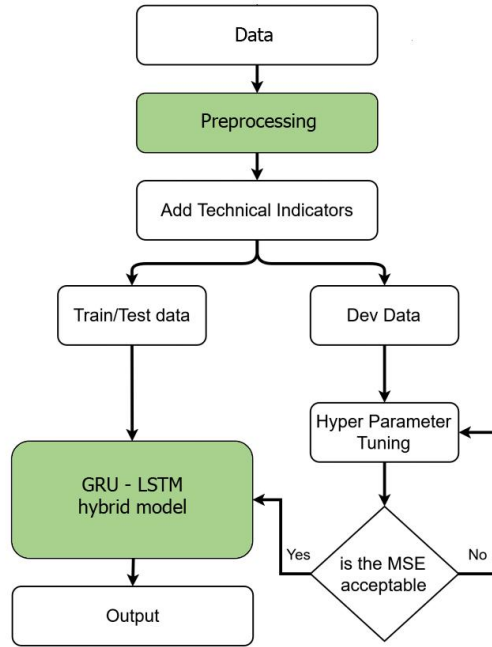


Figure 2. Proposed modeling pipeline with development-data tuning and MSE-based acceptance.

2.2 Moving Average transformation and decomposition

The MA stage is used as the first structured transformation of the raw series. In the experimental design, MA is employed to divide or smooth the data into components described as Trend, Seasonal, and Residual. Trend and seasonal behavior are treated as comparatively linear components, while the residual term captures irregular and typically nonlinear variation. This decomposition-oriented view is useful because it reduces the burden placed on a single predictor: recurrent networks can focus on learning the remaining temporal structure after the series has been organized into components with different statistical behavior.

Two MA parameters are reported in the experimental setup: $d1 = 10$ and $d2 = 50$, described respectively as 10 times per day and 50 times per week. These settings provide two temporal windows, allowing the preprocessing stage to reflect shorter and longer local behavior. The source material emphasizes that MA is computationally inexpensive, making it suitable as a front-end transformation before deeper neural processing. This is especially relevant because the reported experiments were conducted on modest hardware with a 4×2 GHz chipset and 8 GB of RAM.

2.3 GRU and LSTM components

The recurrent portion of the model combines GRU and LSTM layers. GRU employs reset and update gates to control how much previous information should be forgotten or retained. Its relatively compact structure can offer fast computation and strong predictive performance. LSTM, in contrast, uses a memory-cell mechanism and several gates to regulate information flow, enabling the model to preserve useful signals over longer sequences. The hybrid design therefore aims to exploit the computational efficiency of GRU together with the stronger long-range memory behavior associated with LSTM.

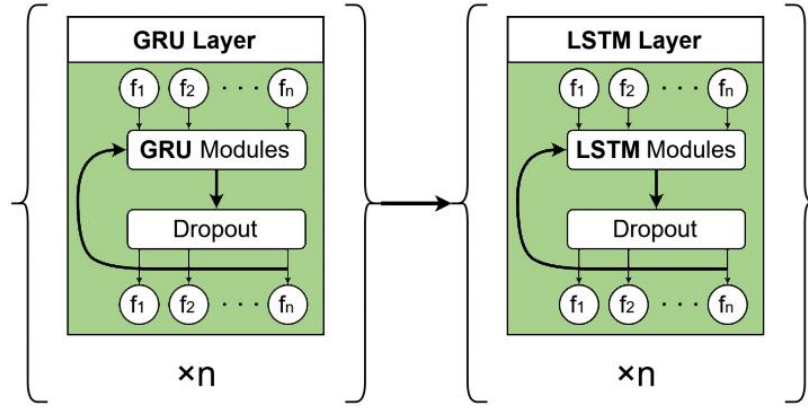


Figure 3. Sequential GRU-LSTM hybrid architecture with dropout between recurrent modules.

The supplied architecture places a GRU block before an LSTM block and uses dropout within the recurrent stacks to reduce overfitting. Feature vectors $f_1, f_2, \dots, f_n$ are fed to the GRU modules, transformed through recurrent learning and dropout, and then passed to the LSTM stage. In the experimental configuration, both the GRU and LSTM models are reported as using 128 layers with 128 neurons per layer, 10 epochs, and a batch size of 10. Because these values are taken directly from the supplied experimental record, they are reported here without reinterpretation. Hyperparameters are tuned using the development subset and accepted according to the resulting MSE.

Table 1. Experimental model settings

Component	Reported setting	Role
MA	$d1 = 10; d2 = 50$	Shorter- and longer-window transformation
LSTM	128 layers; 128 neurons/layer; epoch = 10; batch size = 10	Long-term sequential learning
GRU	128 layers; 128 neurons/layer; epoch = 10; batch size = 10	Efficient gated sequential learning
Hardware	4×2 GHz chipset; 8 GB RAM	Experimental computing environment

3. Data and Result

3.1 Dataset and experimental setting

The empirical dataset consists of observations from a main company and three sub-companies located in Hanoi, Vietnam. Data were collected hourly during working time across approximately 3.5 years: 2022, 2023, 2024, and the first half of 2025. The resulting dataset contains about 8,600 values. This relatively long observation horizon enables the models to encounter multiple short-run fluctuations and longer temporal patterns, while the hourly sampling frequency provides sufficient granularity for recurrent sequence learning.

The experiment uses the preprocessing and model-selection process described in Section 2. After cleaning, MA transformation, feature selection, scaling, and the addition of technical indicators, the model is trained and evaluated. Prediction accuracy is assessed using two criteria. Mean squared error is defined as $MSE = (1/n) \sum (y_i - \hat{y}_i)^2$ and assigns greater penalty to larger deviations. Mean absolute percentage error is defined as $MAPE = (100/n) \sum |(y_i - \hat{y}_i)/y_i|$ and expresses the average prediction error in percentage terms. Lower values indicate better predictive performance for both metrics.

3.2 Comparative performance

Table 2 reproduces the reported benchmark results. The MA baseline produces an MSE of 5,985 and a MAPE of 20.63%. The rebuilt GRU model improves substantially to an MSE of 1,337 and a MAPE of 4.61%, while the

rebuild LSTM model reaches an MSE of 1,305 and a MAPE of 4.50%. The proposed hybrid performs best, with an MSE of 1,265 and a MAPE of 4.37%.

Table 2. Reported prediction performance

Model	MSE	MAPE	Note
MA	5,985	20.63%	Baseline
GRU [2]	1,337	4.61%	Rebuild
LSTM [1]	1,305	4.50%	Rebuild
Proposed	1,265	4.37%	Hybrid MA–GRU–LSTM

The relative improvement is also meaningful. Based on the reported values, the proposed model reduces MSE by approximately 5.4% compared with GRU and 3.1% compared with LSTM. Its MAPE is about 5.2% lower than GRU and 2.9% lower than LSTM. Relative to the MA benchmark, both error measures decline by nearly 79%. These calculations reinforce the conclusion that the gain does not arise only from moving to a generic recurrent model; the combined architecture provides an additional improvement over the two individual deep-learning baselines.

3.3 Interpretation of the hybrid model

The prediction plots supplied with the experiment show that both GRU and LSTM track the target series reasonably closely, but the proposed model provides the lowest aggregate errors. This result is consistent with the motivation of the architecture. GRU can react efficiently to relatively local changes through its update and reset gates, while the LSTM stage can preserve longer temporal information through its memory structure. When these capabilities are preceded by MA-based preprocessing and feature selection, the model receives a cleaner, more structured sequence from which to learn.

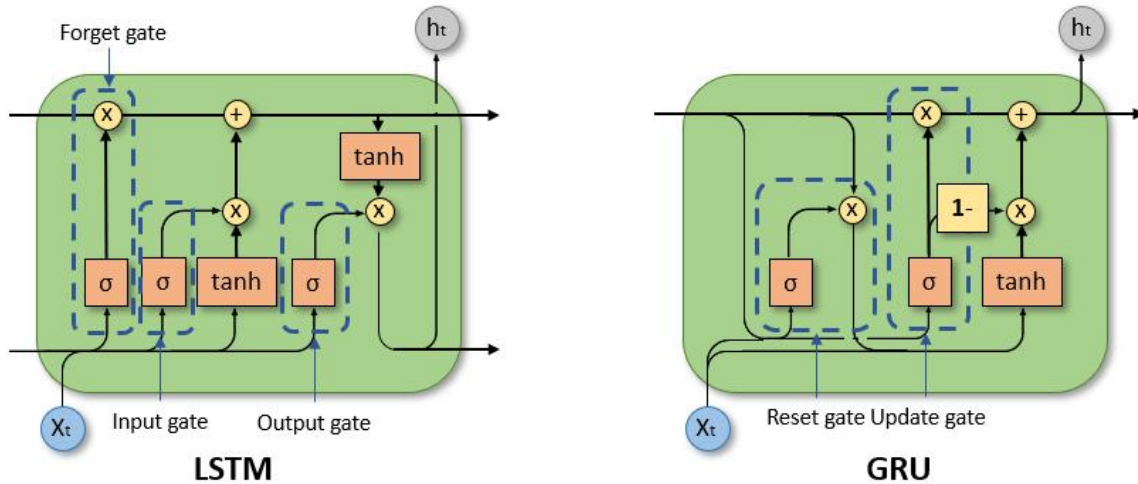


Figure 4. Internal gating mechanisms of LSTM (left) and GRU (right).

The reported novelty of the study lies in successfully developing an MA-based hybrid GRU–LSTM model, demonstrating performance gains over selected recent GRU and LSTM studies, and applying the approach to actual data collected in Hanoi. The experiment therefore contributes primarily at the level of model integration and practical implementation. At the same time, the available evidence should be interpreted within the boundaries of the supplied dataset and evaluation setup: the study reports a single dataset, a fixed collection period, and two main accuracy metrics, while detailed information about the technical indicators, data split ratios, repeated-run variability, and statistical significance tests is not provided in the source material.

4. Conclusion and Future work

This paper presented a hybrid MA–GRU–LSTM framework for sequential prediction using approximately 8,600 hourly observations collected in Hanoi over 3.5 years. The method combines a lightweight MA-based transformation with data cleaning, Random Forest feature selection, scaling, technical-indicator enrichment, and a recurrent architecture in which GRU and LSTM modules are used sequentially. Hyperparameter selection is performed on development data under an MSE-based acceptance condition, after which the final hybrid is evaluated using MSE and MAPE.

The empirical results support the central proposition of the study. The proposed model records an MSE of 1,265 and MAPE of 4.37%, compared with 1,337 and 4.61% for GRU, 1,305 and 4.50% for LSTM, and 5,985 and 20.63% for MA. Thus, the hybrid architecture achieves the best reported performance and yields additional gains beyond either recurrent model used alone. The result suggests that combining preprocessing designed to structure the time series with complementary gated recurrent mechanisms is a promising strategy for complex nonlinear prediction tasks.

Several directions can strengthen the work. First, future studies should increase the number and diversity of input variables through a multi-input design. For stock-price prediction, this can include additional technical, market, sectoral, or macroeconomic variables, provided that the data are consistently available. Second, more powerful neural architectures can be evaluated to determine whether the present accuracy can be improved without excessive computational cost. Third, stronger hardware would reduce training time and permit broader hyperparameter searches, larger networks, and repeated validation runs. Fourth, explicit optimization methods can be integrated into the tuning process to search more efficiently for effective model configurations.

Beyond the future directions stated in the source material, the empirical design would also benefit from a more detailed reporting protocol in subsequent work. In particular, documenting the exact technical indicators, train/test/development split ratios, random seeds, normalization method, forecast horizon, and repeated-run statistics would improve reproducibility. Additional evaluation measures and statistical comparisons across multiple datasets could further establish whether the observed advantage generalizes beyond the current Hanoi-based sample. These extensions would transform the present proof-of-concept into a more comprehensive forecasting framework suitable for broader operational and financial applications.

References

- [1] M. Krichen and A. Mihoub, “Long Short-Term Memory Networks: A Comprehensive Survey,” *AI*, vol. 6, no. 9, article 215, 2025.
- [2] X. Zeng and W. Chen, “A hybrid model integrating GRU with artificial rabbit optimization for stock price prediction,” *International Journal of Intelligent Computing and Cybernetics*, vol. 18, no. 2, pp. 418–443, 2025.