Ain Shams Engineering Journal 17 (2026) 104219

Contents lists available at ScienceDirect

Ain Shams Engineering Journal

journal homepage: www.sciencedirect.com

Full length article

A hybrid time series forecasting model combining recurrent neural networks and ensemble learning for furniture sales prediction

Onur Şahin, Burakhan Çubukçu*

Bilecik Şeyh Edebali University, Department of Computer Engineering, Bilecik, Turkey

ARTICLE INFO

Keywords:
Retail demand forecasting
Hybrid deep learning model
Recurrent neural networks
Ensemble regression
Multi-patch architecture

ABSTRACT

This study proposes a deep learning model, named MP-LRNet, that aims to improve the accuracy and stability of time series forecasting by combining recurrent neural networks and ensemble learning algorithms within a modular Multi-Patch structure. The model is designed to learn temporal patterns at different time intervals and to capture both short- and long-period dependencies in sequential data. Accurate time series forecasting plays a central role in supporting decisions across various practical domains such as retail, production, and energy management. To evaluate the performance of MP-LRNet, experiments were conducted using a real furniture sales dataset and a publicly available energy consumption benchmark (UCI Household Electric Power Consumption). The proposed model achieved an R^2 value of 0.9918, demonstrating reliable predictive ability and consistent results across different configurations. The Multi Patch structure enhanced temporal representation, while integrating long short-term memory and Random Forests improved predictive precision without a significant increase in computational time. The findings indicate that MP-LRNet serves as an effective approach for sales prediction and energy demand estimation, suggesting strong potential to be adapted for broader diverse applications, such as environmental analysis, in future studies.

1. Introduction

Sales forecasting plays a pivotal role in shaping both operational and strategic decisions within modern businesses [1]. Increasing competition in global markets, supply chain disruptions, and uncertainty in consumer behavior have made the need for accurate forecasting models more critical than ever [2–4]. In sectors such as retail and manufacturing, effective inventory control and resource planning are directly tied to the quality of sales forecasts [5]. Therefore, it has become essential to go beyond traditional methods to improve prediction accuracy.

Against this backdrop, data-driven approaches and advanced analytics have enabled more precise and adaptive forecasting processes. Machine learning-based models, for instance, have been shown to reduce stockouts by up to 10% and improve demand forecasting accuracy by approximately 15%, thereby enhancing operational efficiency. Deep learning techniques take it a step further by capturing seasonality and external factors more effectively, resulting in up to a 12% improvement in supply chain performance [4]. These developments highlight the potential of machine learning and deep learning techniques to overcome the limitations of traditional forecasting methods by offering a more comprehensive modeling of seasonality, trends, and external influences.

Mathematical methods form a crucial foundation in the modeling of complex and dynamic systems. Fractal differential models have provided effective tools for analyzing the behavior of systems under uncertainty [6]. Similarly, innovative numerical approaches have been applied to the solution of differential equations in engineering fields [7]. Fluctuation and noise analysis in stochastic processes contribute to the understanding of irregular patterns in time series data [8]. Furthermore, fractal stochastic differential models have offered new perspectives in the mathematical representation of complex data structures [9]. These mathematical foundations establish an essential infrastructure for the development of data-driven forecasting methods.

In this context, time series analysis has long been used as a method for identifying patterns within historical data [10]. Traditional approaches particularly AutoRegressive Integrated Moving Average (ARIMA) [11] and its seasonal extension, Seasonal ARIMA (SARIMA)—have proven effective in modeling linear relationships and stationary data distributions. However, these methods face significant limitations when applied to complex and dynamic data structures. They often struggle to capture sudden demand spikes or the interactions among multiple external factors, and may become computationally

* Corresponding author.
Email address: burakhan.cubukcu@bilecik.edu.tr (B. Çubukçu).

https://doi.org/10.1016/j.asej.2026.104219
Received 16 November 2025; Received in revised form 16 April 2026; Accepted 29 April 2026
Available online 7 May 2026

Table 1
Recent hybrid model studies on time series forecasting.

Study	Year	Model	Application Area	Results	Advantages / Limitations
Djaballah et al. [17]	2024	LSTM + RF + GWO	Bearing Fault Detection	Acc. = 98.97% Prec. = 99.28% Kappa = 99.13	**Advantage:** Feature selection via GWO, high classification accuracy. **Limitation:** Only classification; regression not tested.
Kabir et al. [23]	2025	LSTM + Transformer + MLP	Financial Time Series Forecasting	Superior MSE, RMSE, MAE vs benchmarks	**Advantage:** Self-attention captures long-term dependencies. **Limitation:** High computation cost, no multi-patch used.
Mariappan et al. [24]	2025	CNN + Stacked LSTM + Slime Mould Optimization	Solar Radiation Forecasting	MSE = 0.0359 R^2 = 0.9790 Adj R^2 = 0.9789	**Advantage:** Metaheuristic for hyperparameter tuning, RFE for feature selection. **Limitation:** Tested on single domain, no multi-patch.
Ma et al. [25]	2024	VMD + BiLSTM + GRU	Short-Term Traffic Flow	Better MAE, R^2, RMSE vs baselines	**Advantage:** Noise reduction with VMD, temporal modeling with BiLSTM and GRU. **Limitation:** No ensemble, no multi-patch used.
Shi et al. [26]	2024	LSTM + Transformer	Mine Water Flow Prediction	Higher accuracy vs LSTM, CNN, Transformer, CNN-LSTM	**Advantage:** Self-attention for long dependencies. **Limitation:** High training cost, dataset split optimization needed.
Abumohsen et al. [27]	2024	CNN + LSTM + Random Forest	Solar Energy Forecasting	R^2 = 0.92 RMSE = 0.07 kW MAE = 0.05 kW	**Advantage:** Combines spatial, temporal, and nonlinear modeling. **Limitation:** Tested on one region only.
Guo et al. [28]	2025	CNN + MHSA + LSTM + Attention Mechanisms	Power Load Forecasting	Higher accuracy vs mainstream models	**Advantage:** Multiple attentions, spatial-temporal modeling. **Limitation:** Complex architecture, high training time.
He et al. [29]	2025	Wavelet + CNN + BiGRU + BiLSTM	PM2.5 Concentration Forecasting	R = 0.9952 RMSE = 1.4935 MAE = 1.2091 MAPE = 7.3782%	**Advantage:** Wavelet for denoising, bidirectional modeling. **Limitation:** Complex and computationally intensive.
Salman et al. [30]	2024	Wavelet + LSTM + Transformer	Grid Fault Prediction	Higher accuracy than traditional methods	**Advantage:** Decomposition with Wavelet, LSTM + Transformer synergy. **Limitation:** Long training time on large datasets.
Zhang et al. [31]	2024	GRU + Attention + LSTM	Stock Price Prediction	Higher accuracy than CNN-LSTM	**Advantage:** Enhanced attention via GRU, long-term memory with LSTM. **Limitation:** No CNN, lacks spatial feature extraction.
Wen et al. [32]	2024	GRU + TCN + Attention Mechanism	Short-Term Load Forecasting	Lower error vs traditional methods	**Advantage:** Temporal modeling, multi-scale features, attention. **Limitation:** No bidirectional architectures.
Dakheel and Çevik [33]	2025	LSTM + XGBoost	Smart Grid Load Forecasting	RMSE = 106.54 MW MAPE = 1.18% R^2 = 0.994	**Advantage:** Ensemble strategy improves prediction accuracy and robustness. **Limitation:** Attention mechanisms did not enhance performance.
Quang and Thang [34]	2025	SARIMA + LSTM + RF	Global Gold Price Forecasting	Superior accuracy vs individual models	**Advantage:** Combines statistical and deep learning approaches with ensemble techniques. **Limitation:** Does not explicitly utilize a multi-patch temporal architecture.

inefficient when dealing with large datasets. These shortcomings become especially evident in high-variance, multi-factor environments such as financial market forecasting or retail sales data analysis.

The limitations of traditional time series models have led researchers to explore deep learning and hybrid modeling techniques. Architectures such as Long Short-Term Memory (LSTM) [12] and Gated Recurrent Unit (GRU) [13] have shown strong capabilities in capturing long-term dependencies and nonlinear patterns. However, deep learning alone may fall short when dealing with heterogeneous data sources. In such cases, hybrid models that combine deep learning with ensemble machine learning methods—such as Random Forest, XGBoost, Gradient Boosting Machine (GBM), and HistGradientBoosting Regressor (HGBR) [14–16] offer a more flexible and reliable solution. For instance, Djaballah et al. [17] demonstrated that a hybrid model combining LSTM and Random Forest outperformed their individual use in bearing fault detection, achieving 98.97% accuracy, compared to 93.56% with LSTM and 98.44% with Random Forest. Furthermore, such advanced machine learning and hybrid approaches have been successfully applied across various other complex domains, including quantitative finance [18], corporate financial performance analysis [19], and supply chain management [20].

Hybrid models have become a key research focus in time series forecasting due to their potential for improved accuracy. Table 1 provides a summary of recent studies in this field, comparing various hybrid architectures that combine deep learning methods such as LSTM, GRU, BiLSTM, CNN, and Transformer with ensemble techniques like Random Forest and XGBoost. The table also outlines their application domains, including energy, finance, fault detection, air pollution, and sales forecasting, along with the performance metrics reported in each study.

to hybrid approaches, the multi-patch approach has also gained attention for improving forecasting accuracy. This multi-patch approach allows the analysis of time series data at different scales, integrating both high-frequency dynamics and long-term trends into a single model. Woo et al. developed a universal prediction model, MOIRAI, using this method, and achieved forecasts competitive with fully trained models on the LOTSA dataset containing 27 billion observations. It is stated that the model provides high reliability even in complex scenarios, offering an effective solution for industrial applications [21].

The multi-patch approach is a structural design aimed at simultaneously capturing temporal dependencies across different scales. Prominent applications of this concept are predominantly found in Transformer-based models such as MOIRAI [21] and PatchTST [22]. However, these architectures typically necessitate large-scale datasets and substantial computational resources (GPUs). To the best of our knowledge, MP-LRNet is the first study to combine a multi-patch architecture with RNNs (LSTM) and ensemble learning (Random Forest). Compared to Transformer-based alternatives, this RNN-Ensemble-based multi-patch approach provides practical advantages, including the ability to operate on standard hardware, delivering effective results with less training data, and offering structural flexibility through its modular design.

While the literature demonstrates that hybrid use of deep learning and machine learning techniques is increasingly common, it appears that higher performance can be achieved with more complex structures in time series analysis. Moreover, more studies should focus on real-world problems, employing hybrid architectures and comparing their results with a wider range of other hybrid models. To the best of our knowledge, no studies to date have focused on sales forecasting prediction using a model based on LSTM, Random Forest, and the multi-patch approach.

For these reasons, this study aims to contribute to the literature by introducing the Multi-Patch LSTM–RF Network (MP-LRNet), a model that integrates LSTM, Random Forest, and the multi-patch approach to perform sales forecasting using real-world data from a furniture retailer. Furthermore, evaluating the proposed model against 200 different internal architectures, along with validating its generalizability on a

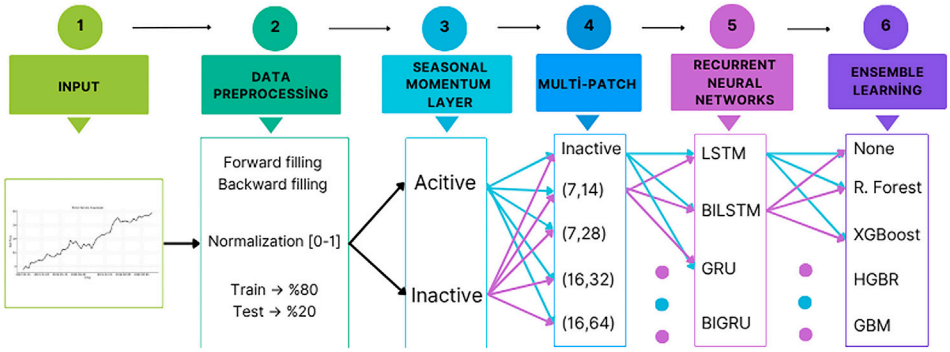

Fig. 1. An overview of the construction process of MP-LRNet and other comparative models.

widely recognized public benchmark dataset (UCI Household Electric Power Consumption), is expected to provide substantial additional contributions to the literature.

2. Method

The proposed MP-LRNet model for sales forecasting in the furniture industry is applied to a real-world dataset obtained from an international furniture manufacturer headquartered in Bursa, Turkey. Since the dataset used is unique and proprietary, direct comparison of the proposed model with studies in the existing literature may not yield significant results. Therefore, the MP-LRNet model is evaluated against 200 different models constructed using various combinations of the core components integrated within MP-LRNet itself.

The construction process of MP-LRNet and the other models is illustrated in Fig. 1. In all experiments, the same input data and identical pre-processing steps were applied to ensure consistency. Following pre-processing, models were built by systematically combining different components, including the presence or absence of a Seasonal Momentum Layer, one of the Multi-Patch configurations (None, (7,14), (7,28), (16,32), or (16,64)), a recurrent neural network variant (LSTM, BiLSTM, GRU, or BiGRU), and an ensemble learning method (None, Random Forest, XGBoost, HistGradientBoosting Regressor, or Gradient Boosting). Each of the 200 constructed combinations was executed ten times under identical conditions, and the mean performance metrics were computed for each configuration.

All experiments were conducted in Google Colab, equipped with an NVIDIA T4 GPU, 12.7 GB of system RAM, and 16 GB of GPU RAM. The software environment included Python 3.9, TensorFlow 2.10.0, Scikit-learn 1.1.3, and Pandas 1.5.1. The following subsections provide detailed information on the dataset, data pre-processing steps, the methods used, and the proposed model.

2.1. Dataset and data pre-processing

To comprehensively evaluate the predictive capabilities and generalizability of the proposed MP-LRNet architecture, experiments were conducted on two distinct datasets. The primary dataset consists of proprietary retail furniture sales data, while a secondary public benchmark dataset was utilized to validate the model's domain transferability. The characteristics of both datasets are detailed below.

2.1.1. Furniture retail sales dataset

The dataset used in this study consists of sales records from a Turkey-based furniture manufacturing and retail company, covering the period from January 1, 2022, to October 31, 2023. The dataset contains a total of 37,006 rows and includes key attributes such as order date, order quantity, unit price, and net price. Order quantities range from 1 to 750 units, while net prices vary between 390 TRY and 304,297 TRY. Table 2 presents a sample segment from the dataset, and Fig. 2 illustrates the input that was used in this study.

Table 2
A sample portion of the dataset.

Order Date	Order Quantity	Price (TRY)	Net Price (TRY)
14.04.2022	1	6,199.31	6,199.31
14.04.2022	1	2,596.41	2,596.41
14.04.2022	1	773.09	773.09
14.04.2022	2	860.61	1,721.22
14.04.2022	1	10,881.60	10,881.60
14.04.2022	2	2,698.52	5,397.04
14.04.2022	6	300.00	1,800.00
14.04.2022	1	9,845.96	9,845.96
15.04.2022	1	3,230.93	3,230.93
15.04.2022	1	3,011.64	3,011.64
15.04.2022	1	3,859.57	3,859.57

In the data pre-processing phase, missing values and outliers were identified and addressed using appropriate techniques. To preserve the temporal structure of the time series, missing values were imputed using forward and backward filling methods [35]. The data was then aggregated on a daily basis to obtain the total sales amount per day, ensuring suitability for time series analysis.

The dataset was split into 80% training and 20% testing subsets. The training set was used to learn the model parameters, while the testing set was reserved to evaluate the generalization performance of the models.

For data scaling, Min-Max normalization, as defined in Equation 1, was applied to transform all values into the [0, 1] range. Here, X_{norm} represents the normalized value, X is the original value, and X_{min} and X_{max} denote the minimum and maximum values of the feature, respectively.

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{1}$$

2.1.2. UCI household electric power consumption dataset

To validate the generalizability of the MP-LRNet model, the Individual Household Electric Power Consumption dataset obtained from the UCI Machine Learning Repository was employed as a secondary evaluation benchmark [36]. This dataset contains minute-level resolution energy consumption measurements collected from a household in Sceaux, France, spanning a period of nearly four years from December 2006 to November 2010.

The original dataset comprises 2,075,259 instances and includes seven numerical features: *Global_active_power* (total active power consumed, kW), *Global_reactive_power* (reactive power, kW), *Voltage* (V), *Global_intensity* (current intensity, A), *Sub_metering_1* (kitchen consumption, Wh), *Sub_metering_2* (laundry room consumption, Wh), and *Sub_metering_3* (electric water heater and air conditioner consumption, Wh). The *Global_active_power* (daily average active power consumption) was selected as the target variable.

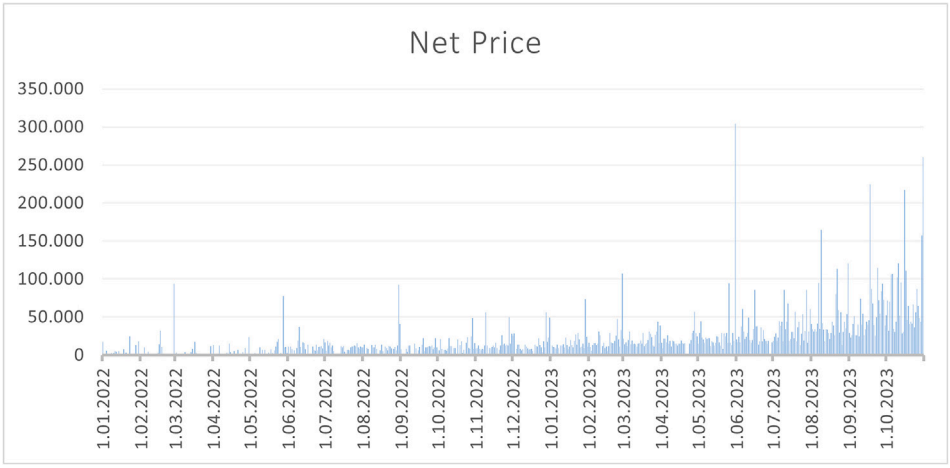

Fig. 2. Time series of net price (The study's input).

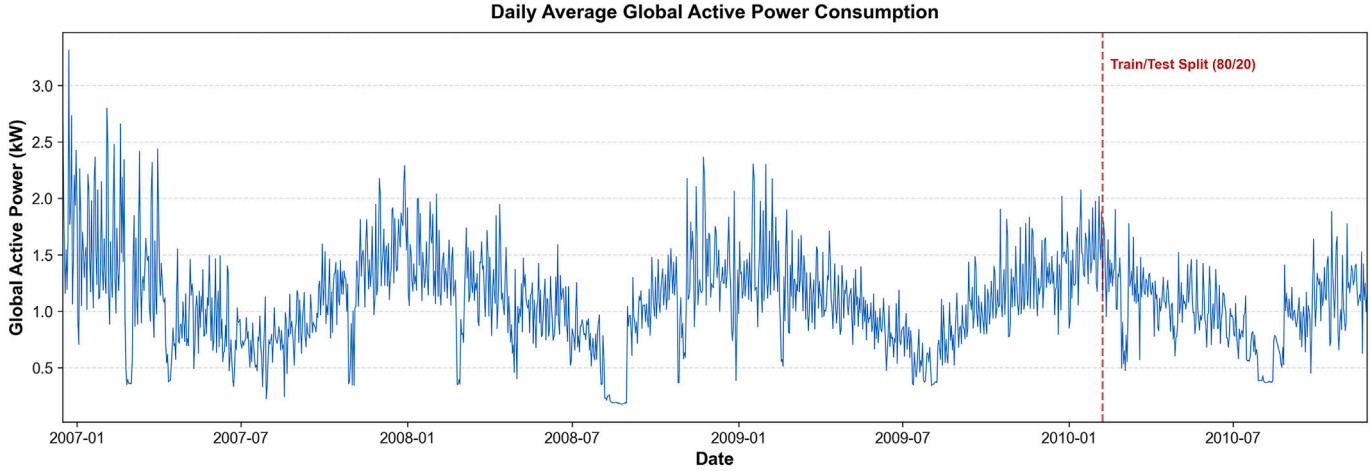

Fig. 3. Time series of the daily average active power consumption.

To ensure a consistent comparison with the furniture sales dataset, the minute-level data was aggregated to a daily average frequency. Following the removal of missing values, a total of 1433 daily observations were obtained. The time series of the daily average active power consumption is illustrated in Fig. 3.

This dataset exhibits structural characteristics highly comparable to the furniture sales data: a daily frequency, a multivariate nature, and a distinct weekly periodic cycle. The average weekend consumption (~1.23 kW) is approximately 18% higher than the weekday average (~1.04 kW). This prominent weekly cycle ensures that the multi-patch configuration of [7, 14] days utilized in our study is equally capable of capturing meaningful temporal patterns in this dataset. The identical experimental protocol from the original study was applied: a chronological split of 80% for training (1,146 days) and 20% for testing (287 days), Min-Max normalization, and 10 independent runs for each configuration.

2.2. Seasonal momentum layer

Capturing seasonal patterns in time series forecasting is a critical factor that directly affects prediction accuracy. While traditional approaches perform seasonal decomposition via fixed periodic components, deep learning-based methods have the capacity to learn these components directly from the data. In the N-BEATS model, Oreshkin et al. [37] obtained interpretable seasonal representations using Fourier-based seasonal expansion layers. Inspired by this approach, the Seasonal Momentum Layer (SML) proposed in our study is designed to capture weekly periodic patterns through learnable weights. The SML operates on an input tensor and uses an element-wise multiplication mechanism that scales each feature based on the day of the week. Unlike traditional fixed seasonal decomposition methods, the SML is a fully data-driven and end-to-end trainable layer. The mathematical formulation and pseudocode of this layer are presented below.

The SML operates on the input tensor $X \in \mathbb{R}^{B\times T\times F}$, where B is the batch size, T is the number of time steps, and F is the number of features. For each time step t, the seasonal index $s(t)$ is calculated as follows:

$$s(t) = (t + d_0) \bmod P \tag{2}$$

Here, $P = 7$ represents the weekly period, and $d_0 = 5$ denotes the starting day offset (e.g., Saturday). The learnable weight matrix $W_s \in \mathbb{R}^{P\times F}$ is initialized using the Glorot uniform distribution. The output tensor Y is then computed by scaling each feature with its corresponding learnable day-of-the-week factor:

$$Y[b,t,j] = X[b,t,j] \cdot W[s(t),j] \tag{3}$$

In vector notation for the entire batch and feature space at a given time step t, this can be expressed using element-wise (Hadamard)

multiplication (⊙):

$$Y[:,t,:] = X[:,t,:] \odot W[s(t),:] \tag{4}$$

To further clarify the internal mechanism for reproducibility, the pseudocode for the proposed SML is provided in Algorithm 1.

Algorithm 1 Seasonal Momentum Layer (SML).

Input: $X \in \mathbb{R}^{B\times T\times F}$, $P \leftarrow 7$, $d_0 \leftarrow 5$
Initialize: Learnable weights $W_s \in \mathbb{R}^{P\times F} \sim$ GlorotUniform()
for $t = 0$ to $T-1$ **do**
  $s(t) \leftarrow (t + d_0) \bmod P$
  $Y[:,t,:] \leftarrow X[:,t,:] \odot W_s[s(t),:]$
end for
Return: Y

2.3. Recurrent neural network-based models

For sales forecasting in the furniture industry, four different recurrent neural network (RNN) architectures were utilized: LSTM, GRU, BiLSTM, and BiGRU. These models are designed to capture sequential relationships and long-term dependencies in time series data.

In this context, RNN-based architectures such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) play a critical role due to their ability to effectively model long-range dependencies. LSTM processes both short- and long-term information through its forget, input, and output gates, which regulate what information should be retained, updated, or passed to the next state. Fig. 4 [38] illustrates the basic LSTM architecture, while its corresponding mathematical formulation is presented below.

Specifically, the forget gate determines which information is discarded from the previous cell state:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5}$$

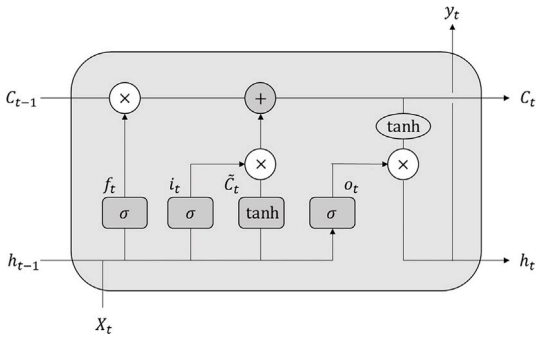

Fig. 4. LSTM architecture.

The input gate and the candidate cell state work together to enable the addition of new information:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{6}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{7}$$

The cell state is updated as follows:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{8}$$

The output gate computes the hidden state:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{9}$$

$$h_t = o_t \odot \tanh(C_t) \tag{10}$$

Here, x_t represents the input vector at time t, and h_{t-1} is the previous hidden state. The W and b terms denote the respective weight matrices and bias vectors for each gate. σ and tanh represent the sigmoid and hyperbolic tangent activation functions, respectively, while ⊙ denotes the element-wise (Hadamard) multiplication. This gate architecture effectively mitigates the vanishing gradient problem [12].

While LSTM effectively handles long-term dependencies, the Gated Recurrent Unit (GRU) operates on similar principles but utilizes a more streamlined architecture, making it computationally more efficient and faster to train [39]. Fig. 5 [40] presents the basic GRU architecture, while its corresponding mathematical formulation is detailed below.

The GRU employs a two-gate mechanism. The reset gate determines how much of the previous hidden state should be incorporated into the new candidate state:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{11}$$

The update gate controls the balance between the old and the new state:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{12}$$

Once the candidate hidden state is computed:

$$\tilde{h}_t = \tanh(W \cdot [r_t \odot h_{t-1}, x_t] + b) \tag{13}$$

the final state is updated as follows:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{14}$$

Here, x_t represents the input vector at time t, and h_{t-1} is the previous hidden state. The W and b terms denote the respective weight matrices and bias vectors for each gate. σ and tanh represent the sigmoid and hyperbolic tangent activation functions, respectively, while ⊙ denotes the element-wise (Hadamard) multiplication. Unlike the LSTM, the GRU does not maintain a separate cell state, resulting in a network with fewer parameters [13].

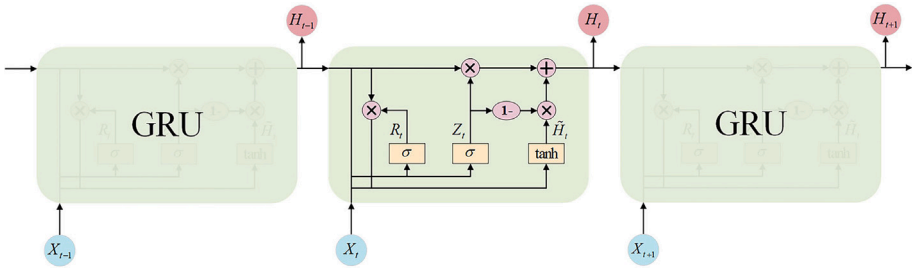

Fig. 5. GRU architecture.

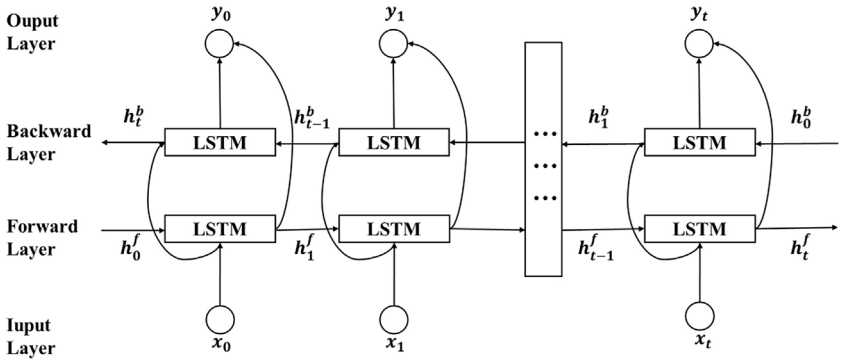

Fig. 6. BiLSTM architecture.

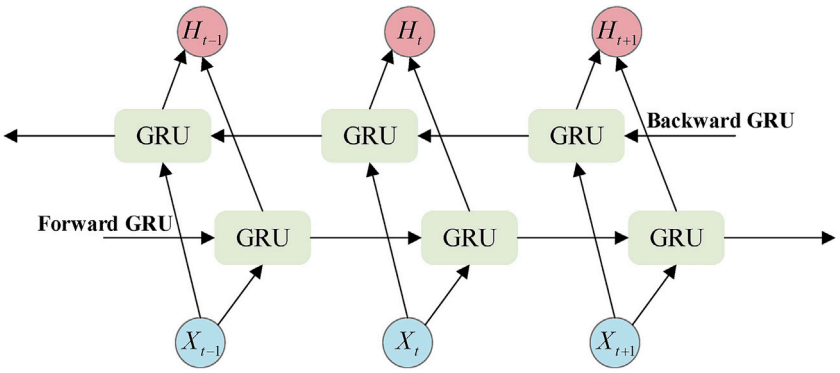

Fig. 7. BiGRU architecture.

In this study, not only were the standard LSTM and GRU models employed, but also their bidirectional variants, namely Bidirectional LSTM (BiLSTM) and Bidirectional GRU (BiGRU). These bidirectional models enable the learning of dependencies in both forward and backward directions in a time series, offering a richer temporal representation. As a result, the model becomes sensitive not only to past inputs but also to future context, potentially improving prediction accuracy [41]. Fig. 6 [38] illustrates the basic BiLSTM architecture, while Fig. 7 [40] presents the basic BiGRU architecture.

All applied models shared a common architectural setup. Each model was implemented as a single-layer structure containing 200 neurons. Neurons, the fundamental computational units in neural networks, are capable of capturing complex relationships among data points. To support the learning of non-linear and intricate patterns, ReLU (Rectified Linear Unit) activation was used. In order to enhance generalization and prevent overfitting, both dropout and recurrent dropout rates were set to 20%. Dropout functions by randomly deactivating some neurons during training, which encourages the model to learn general patterns instead of memorizing specific samples [42].

To better capture seasonal patterns inherent in time series data, a SeasonalMomentumLayer was placed before the core recurrent layer (LSTM, GRU, BiLSTM, or BiGRU) in each model [37]. This specialized layer is designed to facilitate more effective learning of seasonal dynamics. Following the recurrent component, a Dense layer with 200 neurons and ReLU activation was added to extract meaningful predictions from the model's internal representations.

All models were trained using the Adam optimization algorithm [43], which is a widely used optimizer known for its efficiency and adaptive learning rates. The learning rate was set to 0.001 for all configurations. The hyperparameters used in the deep learning models are summarized in Table 3.

Table 3
Hyperparameter settings used in the recurrent neural network-based models.

Category	Hyperparameter	Value(s)
Core Settings	Number of Epochs	20
	Batch Size	64
	Learning Rate	0.001
	Optimization Algorithm	Adam
	Loss Function	Mean Squared Error
	Dropout Rate	0.2
	Recurrent Dropout Rate	0.2
Model Architecture	Number of Neurons	200, Dense Layer: 200
	Activation Function	ReLU
Input Configuration	Patch Sizes (Time Steps)	[7, 14], [7, 28], [16, 32], [16, 64]

2.4. Ensemble learning algorithms used in the study

To further improve the accuracy and robustness of predictions generated by deep learning models (LSTM, GRU, BiLSTM, BiGRU), this study incorporates ensemble learning algorithms as an additional modeling layer. While recurrent neural networks are effective in capturing temporal dependencies, their outputs can benefit from post-processing through ensemble methods, which are known for reducing variance and enhancing generalization. Within this framework, the outputs of the deep learning models serve as the input for a second-stage prediction layer composed of ensemble regressors. These include Random Forest, XGBoost, Gradient Boosting, and HistGradientBoosting algorithms, each selected for their strong performance in regression tasks. To enable the models to handle multiple target variables simultaneously, they are wrapped using the MultiOutputRegressor interface.

Random Forest works by training multiple decision trees on different subsets of the data that are randomly selected. This method helps improve generalization and reduces the risk of overfitting [44]. XGBoost and Gradient Boosting, which rely on sequential learning, train each model to correct the mistakes made by the previous one. XGBoost includes additional features such as regularization and parallel processing, which make it more efficient on complex tasks. Gradient Boosting follows a similar structure but may require more careful parameter tuning [45,46].

HistGradientBoosting, available in the Scikit-learn library, is designed to work efficiently with large datasets. Group feature values into discrete bins, allowing faster training without significant performance loss.

To ensure consistency and comparability across all models, random state was set to 42, and n estimators was fixed at 100.

2.5. Multi-patch approach

The multi-patch approach aims to analyze time series data at multiple temporal resolutions to capture both short-term fluctuations and long-term trends more effectively. In this method, data segments (patches) extracted from different time windows are processed separately, and their outputs are combined to generate the final prediction. Fig. 8 shows the architecture of the Multi-Patch approach [21].

In this study, several patch configurations were tested, including (7,14), (7,28), (16,32), and (16,64) day intervals. The proposed model utilizes two specific patch sizes: 7-day and 14-day. The 7-day patch is designed to capture weekly patterns, while the 14-day patch targets bi-weekly dynamics. Each patch is independently used as input to the deep learning models, and the resulting outputs are then merged and passed as input to the ensemble regression layer.

2.6. MP-LRNet: multi-patch LSTM–RF network

The proposed hybrid model, called MP-LRNet (Multi-Patch LSTM–RF Network), combines multi-scale temporal representations through a multi-patch input mechanism with deep recurrent networks and ensemble-based refinements, aiming to achieve accurate sales forecasting. MP-LRNet architecture is basically demonstrated in Fig. 9.

In the MP-LRnet, raw time series data are first subjected to normalization using Min-Max scaling to standardize input values and ensure that all features fall within the same numerical range. Following normalization, the data is reshaped into multiple temporal segments, known as patches, to capture different levels of temporal granularity. In this study, two patch sizes, 7-day and 14-day, are employed to simultaneously capture short-term dynamics and medium-term seasonal trends in sales data. Each of these patches serves as an independent input sequence for the subsequent learning component.

Each patch is provided as input to an LSTM model, which processes the temporal dependencies within the sequence. The outputs of the LSTM model are referred to as feature vectors. These feature vectors are then used as input to the Random Forest Regressor. By combining the predictions from multiple decision trees trained on different subsets of the feature space, Random Forest enhances the model's generalization ability and captures relationships that may not be fully modeled by the LSTM alone.

2.7. Performance evaluation metrics

To assess the performance of the models, four different evaluation metrics were used: Coefficient of Determination (R^2), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). These metrics allow the accuracy of the models to be evaluated from different perspectives.

R^2, shown in Eq. (15), measures the proportion of variance in the target variable that is explained by the model, with values ranging between 0 and 1. Values closer to 1 indicate better model performance in explaining the data [47]. MAE, presented in Eq. (16), calculates the average of the absolute differences between predicted and actual values [48]. RMSE, defined in Eq. (17), is the square root of the average of the squared differences between predicted and actual values [49]. MAPE,

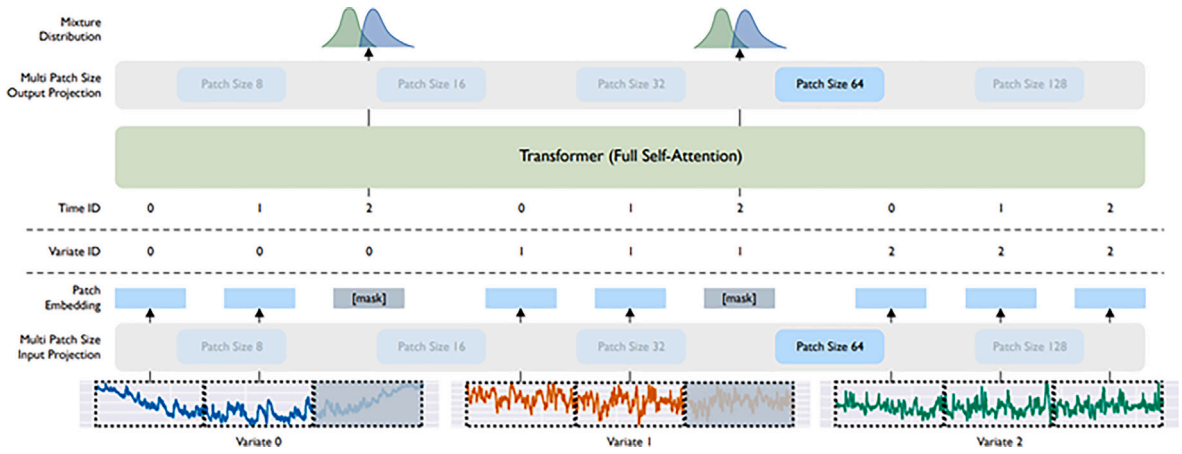

Fig. 8. Multi-Patch approach architecture.

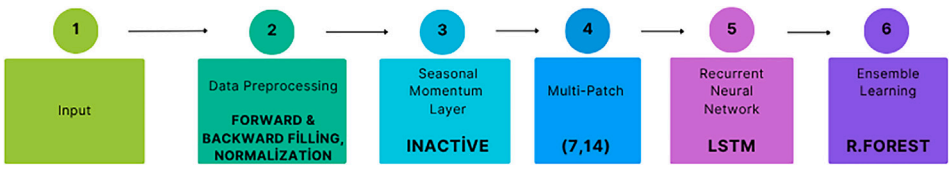

Fig. 9. Multi-Patch LSTM–RF network architecture.

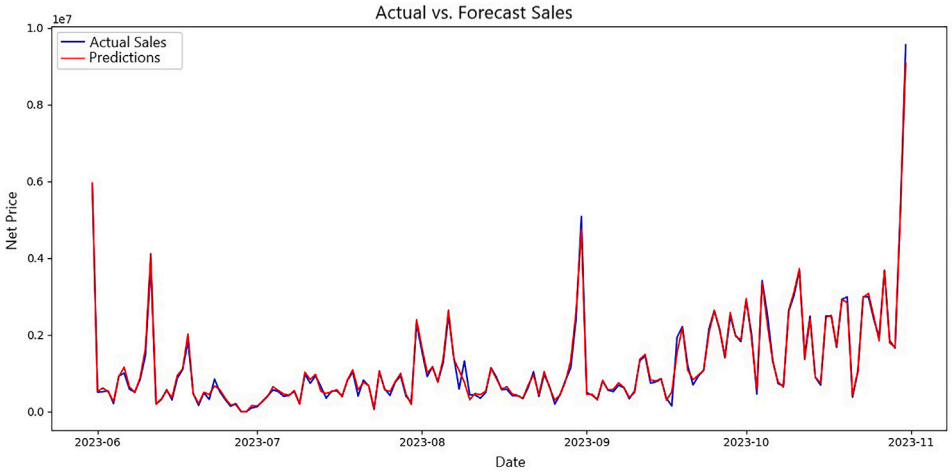

Fig. 10. MP-LRNet sale prediction results.

Table 4
Results of LSTM-based models (mean of 10 runs).

Seasonal Momentum	Multi-Patch Config	Ensemble Alg.	R^2	MAE	RMSE	MAPE	Time (min)
MP-LRNet			*0.9918*	*0.7321*	*1.1896*	*10.1562*	*16.87*
Active	7, 14	RF	0.9910	0.7518	1.2462	10.2768	17.45
Inactive	7, 28	RF	0.9871	0.9501	1.4907	12.2640	20.02
Active	7, 28	RF	0.9869	0.9533	1.5037	12.1854	19.98
Inactive	16, 32	RF	0.9858	1.0158	1.5624	13.1863	21.23
Active	16, 32	RF	0.9853	1.0330	1.5916	13.2352	22.02
Inactive	7, 14	XGB	0.9836	1.0422	1.6792	12.4183	16.80
Active	7, 14	XGB	0.9820	1.0731	1.7568	12.9476	10.47
Inactive	16, 32	XGB	0.9809	1.1501	1.8115	13.3735	27.10
Inactive	7, 28	XGB	0.9809	1.1356	1.8105	13.0542	19.57
Inactive	None	RF	0.9803	1.2739	1.8422	16.5570	27.43
Active	None	RF	0.9791	1.3008	1.8943	16.7029	17.30
Active	7, 28	XGB	0.9789	1.1506	1.9011	13.0659	13.30
Inactive	16, 64	RF	0.9785	1.3324	1.9221	16.5861	28.85
Active	16, 32	XGB	0.9782	1.1795	1.9342	13.3998	15.17
Active	16, 64	RF	0.9773	1.3432	1.9743	16.6076	28.78
Inactive	7, 14	GBM	0.9758	1.2131	2.0394	13.1448	18.48
Inactive	7, 28	GBM	0.9756	1.2162	2.0447	13.4352	22.10
Inactive	None	XGB	0.9746	1.3772	2.0891	16.4470	18.62
Inactive	16, 64	XGB	0.9737	1.3602	2.1213	15.8152	34.78
Inactive	7, 14	HGBR	0.9734	1.2700	2.1370	13.6754	17.17
Active	7, 14	HGBR	0.9722	1.2636	2.1805	13.5618	9.78
Active	7, 14	GBM	0.9721	1.2563	2.1880	13.4442	11.20
Active	None	XGB	0.9721	1.4346	2.1899	16.7620	10.08
Inactive	16, 32	GBM	0.9715	1.3039	2.2035	14.3571	29.20
Inactive	16, 32	HGBR	0.9714	1.3222	2.2107	14.5674	26.77
Inactive	7, 28	HGBR	0.9710	1.3159	2.2280	13.9855	20.27
Active	7, 28	GBM	0.9707	1.2935	2.2418	13.8779	14.33
Active	16, 64	XGB	0.9704	1.3864	2.2521	15.8033	21.68
Active	16, 32	GBM	0.9694	1.3439	2.2906	14.6410	16.48
Active	7, 28	HGBR	0.9689	1.3411	2.3075	14.1527	12.90
Inactive	None	HGBR	0.9688	1.4826	2.3137	16.9109	16.83
Inactive	None	GBM	0.9687	1.4694	2.3171	16.8484	18.45
Inactive	16, 64	HGBR	0.9671	1.4338	2.3702	16.0135	30.58
Inactive	16, 64	GBM	0.9668	1.4425	2.3794	16.0757	33.18
Active	16, 32	HGBR	0.9654	1.4143	2.4354	15.0201	14.95
Active	None	GBM	0.9620	1.5512	2.5522	17.1613	11.75
Active	16, 64	GBM	0.9615	1.4950	2.5590	16.2681	23.35
Active	None	HGBR	0.9602	1.6040	2.6138	17.5737	9.95
Active	16, 64	HGBR	0.9547	1.6001	2.7779	17.0471	21.43
Inactive	16, 32	None	0.9385	1.7884	3.1841	13.8754	21.45
Inactive	7, 28	None	0.9324	1.8401	3.3404	13.5798	17.73
Active	16, 64	None	0.9292	1.9627	3.3977	15.9646	21.33
Active	16, 32	None	0.9282	1.9203	3.4427	14.6882	14.77
Inactive	16, 64	None	0.9263	1.9988	3.4799	15.6603	28.00
Active	7, 28	None	0.9122	2.0922	3.8127	14.7268	12.77
Inactive	7, 14	None	0.9010	2.2579	4.0207	15.3736	14.18
Active	7, 14	None	0.8975	2.3464	4.0686	16.6610	9.65
Active	None	None	0.8860	2.4338	4.1357	18.6236	9.87
Inactive	None	None	0.8168	2.9956	5.2940	21.1228	12.87

Table 5
Results of BiLSTM-based models (mean of 10 runs).

Seasonal Momentum	Multi-Patch Config	Ensemble Alg.	R^2	MAE	RMSE	MAPE	Time (min)
Inactive	7, 14	RF	0.9915	0.7363	1.2099	10.1201	26.13
Active	7, 14	RF	0.9910	0.7590	1.2462	10.2551	26.53
Inactive	7, 28	RF	0.9870	0.9517	1.4938	12.2128	28.18
Active	7, 28	RF	0.9869	0.9516	1.5009	12.1290	27.97
Active	16, 32	RF	0.9861	1.0078	1.5472	13.0848	33.93
Inactive	16, 32	RF	0.9860	1.0119	1.5508	13.1689	33.73
Inactive	7, 14	XGB	0.9820	1.0886	1.7582	12.8555	31.70
Active	7, 14	XGB	0.9819	1.0653	1.7635	12.6464	16.18
Inactive	None	RF	0.9818	1.2225	1.7702	16.0110	45.62
Active	None	RF	0.9807	1.2485	1.8233	16.1430	25.83
Inactive	16, 32	XGB	0.9800	1.1528	1.8502	13.2400	42.10
Active	7, 28	XGB	0.9796	1.1463	1.8738	13.0194	19.97
Inactive	7, 28	XGB	0.9795	1.1696	1.8727	13.3653	32.95
Inactive	None	XGB	0.9793	1.2260	1.8864	14.7019	35.77
Inactive	16, 64	RF	0.9782	1.3238	1.9354	16.4241	48.38
Active	16, 32	XGB	0.9779	1.1730	1.9406	13.3024	25.85
Active	16, 64	RF	0.9779	1.3317	1.9511	16.4174	48.45
Active	None	XGB	0.9777	1.2769	1.9577	15.0589	18.47
Inactive	16, 64	XGB	0.9755	1.2953	2.0496	15.0032	57.67
Inactive	7, 14	GBM	0.9740	1.2399	2.1124	13.3403	33.12
Inactive	None	GBM	0.9739	1.3144	2.1183	15.0658	32.82
Inactive	16, 32	GBM	0.9733	1.2708	2.1392	13.9878	35.93
Inactive	7, 14	HGBR	0.9728	1.2745	2.1609	13.7042	27.83
Inactive	None	HGBR	0.9725	1.3542	2.1726	15.4373	30.52
Active	16, 64	XGB	0.9723	1.3466	2.1768	15.2838	40.17
Active	7, 14	GBM	0.9708	1.2721	2.2367	13.4393	18.10
Active	None	GBM	0.9707	1.3722	2.2418	15.3990	20.52
Active	16, 32	GBM	0.9706	1.2871	2.2465	13.8375	27.27
Inactive	16, 32	HGBR	0.9700	1.3355	2.2686	14.3021	44.28
Active	None	HGBR	0.9691	1.3946	2.2944	15.4420	19.28
Active	7, 14	HGBR	0.9689	1.3267	2.3103	14.0054	16.52
Inactive	7, 28	HGBR	0.9682	1.3888	2.3273	14.8882	39.08
Active	16, 32	HGBR	0.9673	1.3436	2.3573	14.2366	26.17
Inactive	16, 64	GBM	0.9655	1.4502	2.4306	16.0293	63.57
Inactive	7, 28	GBM	0.9654	1.4073	2.4365	14.8648	36.47
Active	7, 28	HGBR	0.9632	1.4292	2.5113	14.6811	21.48
Active	7, 28	GBM	0.9630	1.4089	2.5134	14.5586	21.15
Active	16, 64	GBM	0.9608	1.4673	2.5815	15.7171	43.05
Inactive	16, 64	HGBR	0.9567	1.5896	2.7219	17.2107	58.58
Active	16, 64	HGBR	0.9566	1.5616	2.7248	16.5747	39.48
Inactive	16, 32	None	0.9550	1.6106	2.6747	14.0981	37.42
Active	16, 32	None	0.9429	1.7298	3.0031	14.0916	25.83
Inactive	7, 14	None	0.9403	1.7142	3.0910	12.5957	24.13
Active	7, 28	None	0.9335	1.8674	3.2790	14.3474	20.40
Inactive	7, 28	None	0.9328	1.8427	3.3393	13.3064	30.85
Active	7, 14	None	0.9246	1.9218	3.4373	13.8245	15.95
Inactive	16, 64	None	0.9246	1.9519	3.4668	14.9501	55.17
Active	16, 64	None	0.9125	2.1577	3.7878	16.4407	39.40
Active	None	None	0.9030	2.0900	3.6058	17.6011	18.47
Inactive	None	None	0.8469	2.9041	4.8691	21.3886	23.38

given in Eq. (18), computes the average percentage of the absolute error relative to the actual values, providing an interpretable error metric in percentage terms [50].

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{15}$$

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{16}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{17}$$

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{18}$$

Here, n represents the total number of observations in the dataset, y_i is the actual target value of the i-th observation, $\hat{y}_i$ denotes the corresponding predicted value by the model, and $\bar{y}$ is the mean of all actual target values.

3. Experimental results

In this section, the forecasting performance, computational efficiency, and structural advantages of the proposed MP-LRNet model are evaluated. To provide a comprehensive analysis, the experimental results are divided into two subsections. Section 3.1 presents the detailed findings obtained from the primary retail furniture sales dataset, including comparisons across 200 model configurations. Section 3.2 details the results of the generalizability experiments conducted on the public UCI energy consumption benchmark dataset to demonstrate the domain transferability of the proposed architecture.

3.1. Results on the furniture retail sales dataset

The proposed MP-LRNet model was tested on a real-world furniture sales dataset to validate its forecasting effectiveness. As illustrated in

Table 6
Results of GRU-based models (mean of 10 runs).

Seasonal Momentum	Multi-Patch Config	Ensemble Alg.	R^2	MAE	RMSE	MAPE	Time (min)
Inactive	7, 14	RF	0.9912	0.7527	1.2322	10.3653	17.73
Active	7, 14	RF	0.9902	0.7847	1.2965	10.4694	18.62
Inactive	7, 28	RF	0.9861	0.9836	1.5439	12.4218	20.87
Active	7, 28	RF	0.9855	0.9947	1.5781	12.5065	21.55
Inactive	16, 32	RF	0.9847	1.0611	1.6215	13.6505	22.27
Active	16, 32	RF	0.9834	1.0876	1.6908	13.6772	23.27
Inactive	7, 14	XGB	0.9820	1.0756	1.7598	12.8185	14.73
Inactive	None	RF	0.9794	1.2948	1.8818	16.7094	25.73
Active	7, 14	XGB	0.9790	1.1389	1.8991	13.3091	11.70
Inactive	7, 28	XGB	0.9789	1.1874	1.9014	13.4274	17.95
Inactive	16, 32	XGB	0.9782	1.2099	1.9337	13.7980	20.25
Active	None	RF	0.9773	1.3349	1.9769	17.0404	21.33
Inactive	16, 64	RF	0.9763	1.3688	2.0193	16.9637	29.30
Active	7, 28	XGB	0.9755	1.2361	2.0507	13.6879	14.68
Inactive	7, 14	GBM	0.9754	1.2268	2.0536	13.4143	15.90
Active	16, 64	RF	0.9751	1.3932	2.0703	17.1155	29.52
Active	16, 32	XGB	0.9744	1.2860	2.0990	14.4007	16.45
Inactive	7, 14	HGBR	0.9733	1.2784	2.1377	13.7791	15.10
Inactive	None	XGB	0.9714	1.4475	2.2171	16.9054	17.20
Inactive	7, 28	GBM	0.9712	1.3087	2.2231	13.9853	19.58
Inactive	16, 32	GBM	0.9695	1.3637	2.2858	14.8823	22.55
Inactive	16, 32	HGBR	0.9694	1.3783	2.2939	14.8172	20.38
Inactive	7, 28	HGBR	0.9693	1.3622	2.2976	14.4735	16.87
Active	7, 14	GBM	0.9690	1.3289	2.3090	14.1481	12.45
Active	7, 14	HGBR	0.9685	1.3263	2.3256	14.0131	10.67
Inactive	16, 64	XGB	0.9679	1.4615	2.3443	16.5935	27.80
Active	None	XGB	0.9637	1.5628	2.4982	17.6496	14.87
Inactive	16, 64	GBM	0.9631	1.5070	2.5126	16.4944	30.35
Active	7, 28	GBM	0.9617	1.4590	2.5620	15.1960	15.87
Active	16, 64	XGB	0.9617	1.5466	2.5652	17.0916	24.00
Inactive	None	GBM	0.9616	1.6000	2.5686	17.7007	15.80
Active	16, 32	HGBR	0.9603	1.5035	2.6065	15.7188	15.72
Active	7, 28	HGBR	0.9585	1.5060	2.6688	15.3496	13.95
Active	16, 32	GBM	0.9576	1.5336	2.6965	15.9795	17.47
Inactive	None	HGBR	0.9571	1.6716	2.7117	18.2212	13.83
Inactive	16, 64	HGBR	0.9541	1.6562	2.8007	17.7346	27.68
Active	16, 64	GBM	0.9493	1.7167	2.9472	18.0598	24.93
Active	None	HGBR	0.9491	1.7728	2.9575	18.9327	13.23
Active	None	GBM	0.9466	1.7896	3.0282	19.0009	16.43
Active	16, 64	HGBR	0.9440	1.7797	3.1015	18.6375	22.43
Inactive	16, 64	None	0.9263	1.9256	3.4619	15.4474	24.23
Inactive	7, 28	None	0.9219	1.9223	3.5152	13.7514	15.42
Inactive	16, 32	None	0.9121	2.1089	3.7553	15.6902	17.90
Inactive	7, 14	None	0.9084	2.1827	3.8851	15.1251	13.35
Active	7, 28	None	0.8723	2.5205	4.6508	16.8256	14.17
Active	7, 14	None	0.8718	2.5565	4.6122	17.0894	10.82
Active	16, 32	None	0.8655	2.6465	4.7060	18.4909	15.17
Active	16, 64	None	0.8604	2.6837	4.8459	18.3339	21.68
Active	None	None	0.8543	2.6191	4.7203	19.0308	15.00
Inactive	None	None	0.8007	3.0479	5.4754	21.0663	10.63

Fig. 10, the proposed MP-LRNet model demonstrated highly consistent performance across ten independent runs, confirming its stability and predictive strength in sales forecasting. The model achieved an average R^2 of 0.9918, MAE of 0.7321, RMSE of 1.1896, and MAPE of 10.1562. The best-performing run yielded an R^2 of 0.9927, MAE of 0.6984, RMSE of 1.1205, and MAPE of 9.7073, while even the least-performing iteration maintained robust accuracy with R^2 of 0.9907, MAE of 0.7772, RMSE of 1.2675, and MAPE of 10.8761. These results indicate that the MP-LRNet consistently delivers low forecasting errors and high explanatory power, making it a reliable model for time-series-based sales prediction tasks.

Beyond the explained variance, evaluating absolute error metrics is crucial for sales-related decision-making. The achieved Mean Absolute Error (MAE) of 0.7321 indicates that the average deviation from the actual sales values is notably low, which is essential for baseline inventory planning. The Root Mean Square Error (RMSE), which penalizes larger forecasting errors more heavily, remains at a well-controlled level of 1.1896. This demonstrates that the model produces acceptable and stable error margins even in the presence of extreme sales fluctuations. Furthermore, the Mean Absolute Percentage Error (MAPE) is calculated at 10.1562%; this ratio highlights a strong predictive reliability that is highly adequate for supporting strategic business decisions and supply chain optimization in the retail sector.

To compare the performance of the MP-LRNet model, a total of 200 distinct model configurations were generated by varying four core components: the inclusion of the Seasonal Momentum Layer (active or inactive), the Multi-Patch structures (None, 7–14, 7–28, 16–32, and 16–64), the recurrent neural network type (LSTM, BiLSTM, GRU, or BiGRU), and the ensemble learning algorithm (Random Forest, XGBoost, Gradient Boosting Machine, HistGradientBoosting Regressor, or None). Each configuration was executed ten times under identical conditions, and the mean R^2, MAE, RMSE, and MAPE values were computed together with the average training time. For clarity and to enable architecture-oriented interpretation, the 200 outcomes were grouped into four tables according to the underlying recurrent model type: Table 4 presents the LSTM-based results, Table 5 includes the BiLSTM-based results, Table 6

Table 7
Results of BiGRU-based models (mean of 10 runs).

Seasonal Momentum	Multi-Patch Config	Ensemble Alg.	R^2	MAE	RMSE	MAPE	Time (min)
Inactive	7, 14	RF	0.9910	0.7599	1.2423	10.4166	27.43
Active	7, 14	RF	0.9904	0.7753	1.2857	10.4847	27.97
Active	7, 28	RF	0.9859	0.9815	1.5538	12.3684	31.70
Inactive	7, 28	RF	0.9859	0.9821	1.5596	12.3203	34.07
Inactive	16, 32	RF	0.9852	1.0368	1.5981	13.1920	36.12
Active	16, 32	RF	0.9842	1.0589	1.6511	13.3780	49.93
Inactive	7, 14	XGB	0.9820	1.0788	1.7561	12.7429	23.72
Inactive	None	RF	0.9818	1.2177	1.7695	15.8745	41.62
Active	7, 14	XGB	0.9808	1.0872	1.8179	12.7523	17.23
Inactive	16, 32	XGB	0.9801	1.1671	1.8511	13.3194	33.38
Active	None	RF	0.9794	1.2802	1.8833	16.4149	31.35
Inactive	7, 28	XGB	0.9787	1.1938	1.9121	13.4547	30.88
Inactive	None	XGB	0.9778	1.2694	1.9518	15.0646	33.85
Active	16, 32	XGB	0.9763	1.2164	2.0180	13.6869	28.30
Inactive	16, 64	RF	0.9763	1.3648	2.0190	16.7788	50.37
Inactive	7, 14	GBM	0.9753	1.2122	2.0594	13.1370	25.28
Active	16, 64	RF	0.9752	1.3850	2.0646	16.9887	50.62
Active	7, 28	XGB	0.9742	1.2679	2.1024	14.0922	24.13
Inactive	16, 32	GBM	0.9739	1.2593	2.1165	13.7511	35.03
Inactive	None	GBM	0.9729	1.3512	2.1544	15.2961	27.02
Inactive	7, 14	HGBR	0.9723	1.2872	2.1810	13.8256	23.18
Active	None	XGB	0.9712	1.4169	2.2236	16.2233	25.72
Inactive	16, 64	XGB	0.9711	1.3766	2.2254	15.4562	49.70
Active	7, 14	GBM	0.9709	1.2871	2.2343	13.6623	18.47
Inactive	16, 32	HGBR	0.9707	1.3288	2.2425	14.3109	33.25
Inactive	None	HGBR	0.9705	1.4058	2.2478	15.7047	25.75
Active	7, 14	HGBR	0.9688	1.3146	2.3150	13.7957	16.65
Inactive	7, 28	GBM	0.9685	1.3504	2.3173	14.1757	32.28
Inactive	7, 28	HGBR	0.9666	1.4030	2.3914	14.6787	30.12
Active	16, 64	XGB	0.9637	1.5141	2.4900	16.8141	41.98
Inactive	16, 64	GBM	0.9627	1.4737	2.5225	15.7146	53.48
Active	16, 32	HGBR	0.9626	1.4359	2.5332	14.9235	27.77
Active	7, 28	GBM	0.9615	1.4525	2.5689	14.9297	25.62
Active	16, 32	GBM	0.9594	1.4832	2.6375	15.2886	29.80
Active	7, 28	HGBR	0.9589	1.5197	2.6557	15.4965	23.57
Inactive	16, 64	HGBR	0.9566	1.5938	2.7260	17.1748	51.02
Active	None	GBM	0.9565	1.6201	2.7332	17.4092	26.27
Active	None	HGBR	0.9547	1.6458	2.7887	17.4501	22.42
Inactive	16, 32	None	0.9500	1.5565	2.7559	12.9008	31.13
Active	16, 64	GBM	0.9490	1.6671	2.9479	17.3675	44.77
Active	16, 64	HGBR	0.9461	1.7317	3.0448	18.0224	42.88
Inactive	7, 28	None	0.9443	1.7389	2.9451	13.4015	26.08
Inactive	7, 14	None	0.9369	1.7708	3.1966	12.7699	20.37
Inactive	16, 64	None	0.9293	1.8365	3.4144	13.9044	42.70
Active	7, 14	None	0.9119	2.0910	3.8485	14.2307	17.12
Active	None	None	0.8855	2.4089	4.1504	18.7872	27.03
Active	16, 32	None	0.8756	2.5468	4.4809	17.5874	27.77
Active	16, 64	None	0.8675	2.5026	4.6932	16.9684	42.60
Active	7, 28	None	0.8609	2.8251	4.7917	19.2136	23.30
Inactive	None	None	0.8432	2.8874	4.8601	21.3977	19.12

contains the GRU-based results, and Table 7 reports the BiGRU-based results. In all these tables, the ensemble learning algorithms are abbreviated as Random Forest (RF), XGBoost (XGB), Gradient Boosting Machine (GBM), and HistGradientBoosting Regressor (HGBR).

Across all 200 configurations tested under different combinations of recurrent structures, Multi-Patch settings, ensemble algorithms, and Seasonal Momentum Layer states, the proposed MP-LRNet framework achieved highly consistent forecasting accuracy. The overall mean R^2 across all experiments was 0.9586, with the best configuration reaching 0.9918 and the lowest observed value at 0.8007, demonstrating the model's stability and robustness across diverse settings. These results confirm that MP-LRNet can reliably capture nonlinear dependencies in real-world furniture sales data, maintaining strong predictive capability under various architectural combinations.

To rigorously validate that the superiority of this best-performing MP-LRNet configuration is not merely due to random variance across training runs, a non-parametric Wilcoxon Signed-Rank test was conducted. The test compared the primary evaluation metrics (R^2) of the proposed MP-LRNet against the baseline LSTM model over 10 independent executions. As illustrated in the performance distribution boxplot (Fig. 11), MP-LRNet exhibits both a substantially higher median accuracy and a remarkably narrower variance compared to the baseline. The statistical analysis, implemented via the SciPy library, yielded a p-value of 0.00195 ($p < 0.01$). This definitively confirms that the performance improvement achieved by the multi-patch and ensemble learning integration is statistically significant.

In terms of computational efficiency, the proposed MP-LRNet was trained in an average of 16.87 minutes. While the training times for baseline single-patch LSTM + RF configurations generally ranged between 17 and 27 minutes, the dual-patch structure of MP-LRNet successfully remains at the lower boundary of this spectrum, proving that the multi-patch mechanism does not introduce any significant computational overhead. Furthermore, all experiments were conducted entirely without GPU acceleration on a local machine equipped with an 11th Gen

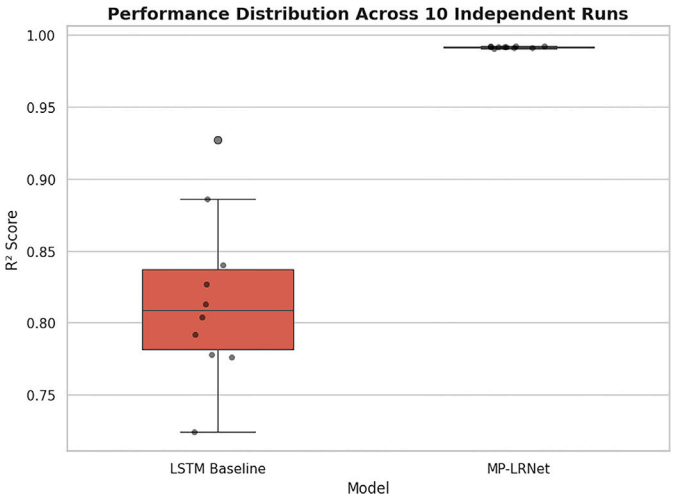

Fig. 11. Performance distribution of MP-LRNet and LSTM baseline.

Intel Core i7 processor and 16 GB of RAM. This confirms the model's high practical applicability, demonstrating that it can be efficiently deployed and retrained using standard hardware resources.

3.1.1. Structural analysis and component evaluation

To determine the most effective architectural combination, the overall impacts of the core components—Seasonal Momentum Layer (SML), recurrent network types, and ensemble algorithms—were evaluated across all 200 configurations.

First, the effect of the SML was analyzed and found to be negligible. When all configurations were averaged, activating the SML slightly decreased the mean R^2 from 0.9749 to 0.9737, while MAE and RMSE rose marginally. This suggests that the seasonal dynamics of the 1.5-year dataset were already sufficiently captured by the recurrent components and the Multi-Patch mechanism. Given the lack of performance gain and the added computational cost, the SML was explicitly excluded from the final MP-LRNet configuration.

In the subsequent stage, the performance of the recurrent architectures was examined. The LSTM-based structures achieved the most consistent results across all combinations, with an average R^2 of 0.9786, mean MAE of 1.09, and RMSE of 1.83. The BiGRU variant followed closely (R^2 = 0.9778) but exhibited slightly higher errors. While GRU models were computationally faster, both BiLSTM and GRU models showed higher variability. Overall, the LSTM architecture provided the best baseline trade-off between accuracy and stability.

Finally, examining the ensemble learning algorithms (Fig. 12), Random Forest (RF) consistently reached the highest predictive accuracy, achieving peak R^2 values of 0.9918. XGBoost (XGB) followed closely with minimal performance gaps, while GBM and HGBR produced stable but less competitive outcomes (R^2 mostly between 0.970–0.975). As expected, configurations trained without any ensemble component ("None") exhibited the weakest accuracy, often dropping below R^2 = 0.93. These general trends firmly indicate that integrating the RF ensemble yields the optimal balance of robustness and predictive precision.

Leveraging the interpretability advantage provided by the Random Forest component, the temporal contribution of the multi-patch configuration was analyzed. Fig. 13 illustrates the impact of different patch configurations on the R^2 and MAE metrics. The results reveal that the (7, 14) day patch pair yields the highest performance (R^2 = 0.9918, MAE = 0.7321). While wider patch intervals (16, 32 and 16, 64) lead to a decrease in performance, the configuration without patching (None + RF, R^2 = 0.9803) also falls short of the (7, 14) pair. These findings indicate that the 7-day patch effectively captures weekly sales cycles, whereas the 14-day patch captures the bi-weekly trend component, and the simultaneous use of these two temporal scales systematically enhances forecasting accuracy.

Building upon this performance evaluation, a feature importance analysis was conducted to quantitatively examine the individual contributions of these multi-patch components to the final prediction. Using the feature importances attribute of the Random Forest layer, the impact of the feature vectors derived by the LSTM was measured. The results show that the 7-day and 14-day patches accounted for 54.4% and 45.6% of the model's decision weight, respectively (Fig. 14). This balanced distribution confirms that the 7-day patch plays a dominant role in capturing weekly sales cycles, while the 14-day patch serves a complementary function by stabilizing bi-weekly trends.

To systematically evaluate the individual contribution of each component, an ablation study was conducted. Table 8 details the progressive impact of the Multi-Patch structure, the ensemble learning layer, and the Seasonal Momentum Layer on the overall model performance.

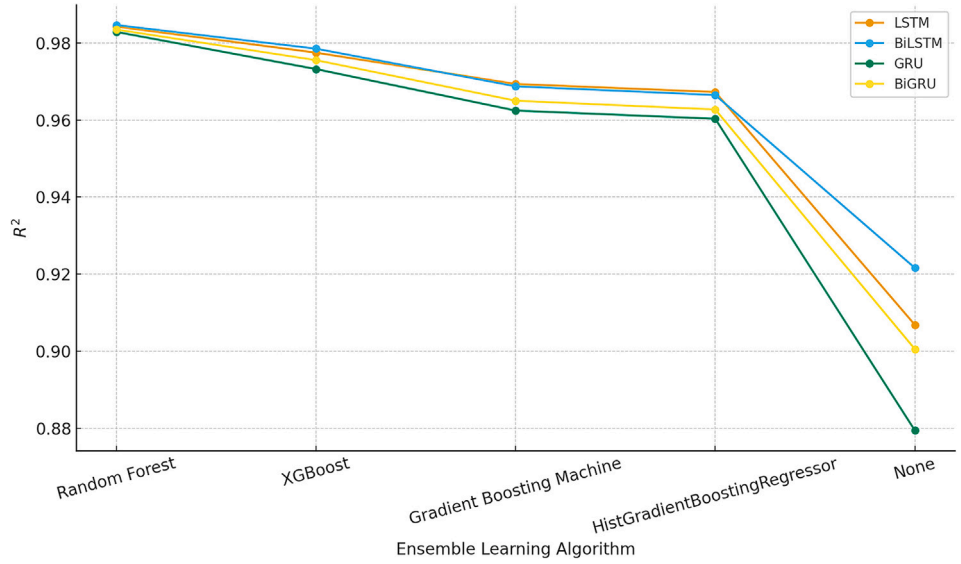

Fig. 12. Effect of ensemble algorithms on the R^2 score.

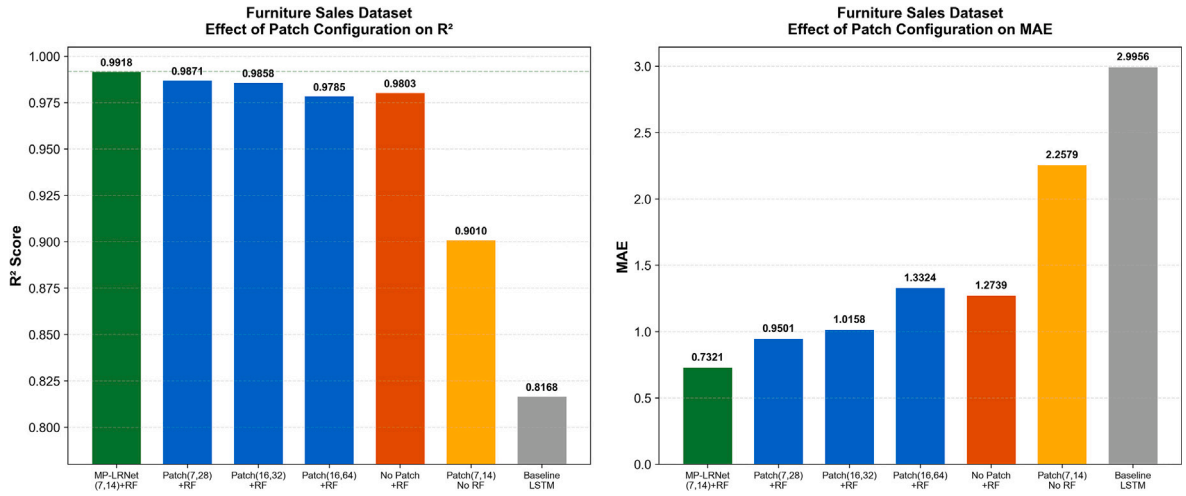

Fig. 13. Effect of different Multi-Patch configurations on forecasting performance for the furniture sales dataset.

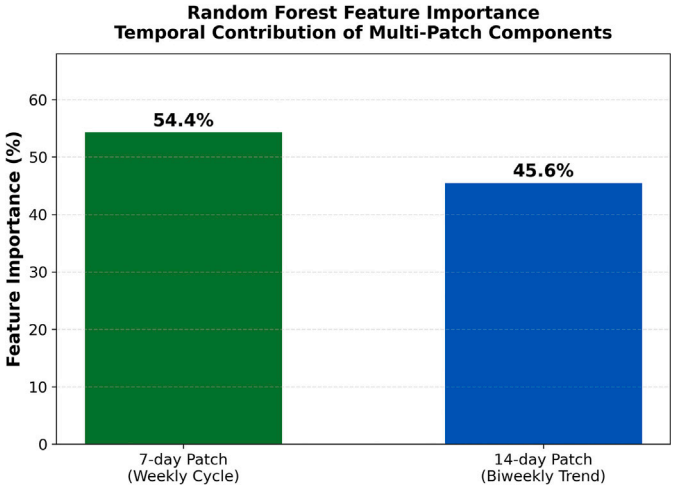

Fig. 14. Feature importance distribution of the 7-day and 14-day temporal patches within the Random Forest layer.

As observed in Table 8, the baseline standard LSTM model (C5) achieved an R^2 of 0.8168. The introduction of the Multi-Patch structure (C4) significantly enhanced the model's ability to capture short- and long-term periodicities, raising the R^2 to 0.9010. Alternatively, when only the Random Forest ensemble was integrated without the multi-patch mechanism (C3), the R^2 reached 0.9803, demonstrating the strong error-correction capability of ensemble learning. Finally, the complete MP-LRNet architecture (C1), seamlessly unifying both the Multi-Patch representation and the RF ensemble layer without the redundant SML (C2), attained the peak performance. This progressive escalation validates that each component provides a distinct and measurable contribution to the final predictive precision.

3.2. Results on the UCI household electric power consumption dataset

To validate the external generalizability and cross-domain applicability of the proposed architecture, the core model configurations were evaluated using the public UCI Household Electric Power Consumption dataset. Table 9 presents the performance metrics of MP-LRNet and its structural variations (baseline LSTM, multi-patch LSTM, and LSTM-RF ensemble) on this secondary dataset.

The experimental results on the UCI dataset mirror the structural benefits observed in the furniture sales data. The fully integrated MP-LRNet architecture, utilizing a (7, 14) multi-patch configuration and Random Forest ensemble, achieved the highest predictive accuracy with an R^2 score of 0.9201 ± 0.0024 and the lowest MAPE (6.95%). Notably, the integration of the multi-patch mechanism consistently enhanced the baseline ensemble approach, improving the R^2 score from 0.9066 (LSTM + RF) to 0.9201 (MP-LRNet). These findings confirm that the modular design of MP-LRNet effectively captures complex temporal patterns across completely different domains, proving its strong transferability.

4. Discussion

The results obtained from the experimental analyses clearly demonstrate the strong predictive capability of the proposed MP-LRNet model. Achieving an average R^2 of 0.9918, the model shows outstanding accuracy and stability in forecasting real-world furniture sales, outperforming the majority of reported deep learning-based time series models in the literature.

While an exceptionally high R^2 value of 0.9918 might naturally raise concerns regarding potential overfitting, multiple structural safeguards and empirical evidence demonstrate the model's robust generalization capabilities. First, strict regularization techniques were embedded within the architecture, specifically utilizing 20% dropout and 20% recurrent dropout rates in the LSTM layers to prevent the co-adaptation of neurons. Second, the Random Forest algorithm employed in the ensemble phase is inherently resistant to overfitting due to its reliance

Table 8
Ablation study demonstrating the individual contribution of each model component.

#	Configuration	SML	Multi-Patch	Ensemble	R^2	MAE	RMSE	MAPE (%)
C1	MP-LRNet (Proposed)	Inactive	7, 14	RF	0.9918	0.7321	1.1896	10.16
C2	+ SML Active	Active	7, 14	RF	0.9910	0.7518	1.2462	10.28
C3	– Multi-Patch	Inactive	None	RF	0.9803	1.2739	1.8422	16.56
C4	– Random Forest	Inactive	7, 14	None	0.9010	2.2579	4.0207	15.37
C5	Baseline LSTM	Inactive	None	None	0.8168	2.9956	5.2940	21.12

Table 9
Performance comparison of model configurations on the UCI dataset (mean of 10 runs).

Model	Multi-Patch Config	Ensemble Alg.	R^2	MSE	MAE	RMSE	MAPE (%)
MP-LRNet	(7, 14)	RF	0.9201 ± 0.0024	0.0076	0.0628	0.0872	6.95
MP-LRNet (7,28)	(7, 28)	RF	0.9116 ± 0.0022	0.0085	0.0678	0.0922	7.69
LSTM + RF	None	RF	0.9066 ± 0.0025	0.0088	0.0684	0.0940	7.54
LSTM + MP	(7, 14)	None	0.4581 ± 0.0096	0.0516	0.1696	0.2271	20.03
LSTM Only	None	None	0.3796 ± 0.0391	0.0587	0.1756	0.2423	20.15

Table 10
Comparison of recent hybrid and deep learning-based time series forecasting studies.

Authors	Year	Dataset	Model / Technique	Metrics and Results
Sherly A. et al. [51]	2025	Simulated and real-world datasets for edge computing	Hybrid ARIMA–Prophet model	Improvement in RMSE and MAE over baseline models; e.g., MAPE = 5.2% and ~8% lower RMSE on primary traffic task.
Mahmoud A. & Mohammed A. [52]	2024	Traffic Volume and Air Quality datasets	Hybrid TCN–BiLSTM model	R^2 = 0.976 (Traffic Volume), R^2 = 0.94 (Air Quality).
Xu Y. et al. [53]	2025	Nine real-world benchmark datasets	Multi-Scale Patch Mixer (MPMixer)	Best MSE/MAE in 36 evaluations on several datasets (details vary per dataset).
Balusani A. et al.[54]	2025	Kaggle retail dataset enriched with weather and product data	XGBoost; hybrid CNN–LSTM–XGBoost	R^2 = 0.953 (XGBoost), R^2 = 0.945 (hybrid).
Alsalem K. [55]	2025	Power consumption data from Tetouan, Morocco (52,417 records)	ML models with fuzzy clustering (RF, SVM, KNN, XGBoost, MLP)	Best by MLP: R^2 = 0.9889, RMSE = 355.42, MAE = 246.43.

on bootstrap aggregating (bagging) and randomized feature selection. Third, to avoid memorization of the training data, the network training was deliberately constrained to a maximum of 20 epochs, during which training loss convergence was continuously monitored, effectively mitigating the need for automated early stopping. Furthermore, the dataset was strictly split chronologically (80% training, 20% testing) to ensure absolutely no future data leakage into the training phase. Finally, empirical evidence from the 10 independent test executions shows R^2 values tightly clustered between 0.9907 and 0.9927. This exceptionally low variance (range = 0.0020) across completely separate test evaluations confirms that the model's high accuracy is a result of consistent pattern extraction rather than dataset memorization.

However, it should be noted that the dataset created from the furniture industry was specifically designed for this study; therefore, direct result comparison with other studies is not entirely meaningful. Nevertheless, when compared with similar hybrid architectures in the literature, the R^2 score (0.9918) achieved by MP-LRNet can be considered highly successful. Table 10 presents some recent studies published after 2024 that use time series analysis and hybrid model approaches. Although a direct superiority comparison is not possible due to differences in datasets and application domains, MP-LRNet demonstrates competitive results across overall performance metrics.

When compared with other recent hybrid time series forecasting approaches, the proposed MP-LRNet model demonstrates promising performance. Sherly A. et al. [51] reported that their hybrid ARIMA–Prophet model achieved a MAPE of 5.2% and an 8% lower RMSE across several traffic datasets. Xu et al. [53] evaluated their Multi-scale Patch Mixer model in 36 independent experiments and achieved the best MSE and MAE results in most of them. Similarly, Mahmoud and Mohammed [52] obtained an R^2 of 0.976 for traffic volume forecasting and 0.94 for air quality prediction, while Alsalem [55] reported an R^2 of 0.9889 for power consumption estimation. In comparison, the MP-LRNet model achieved an R^2 of 0.9918, indicating its ability to capture temporal dependencies effectively in time series forecasting tasks.

The experimental outcomes suggest that MP-LRNet may not only achieve high accuracy but also hold potential for real-world forecasting tasks. Its modular structure is believed to allow flexible integration of recurrent and ensemble components, which could facilitate adaptation to various time-dependent domains such as sales prediction, energy consumption analysis, and environmental monitoring. The Multi-Patch mechanism appears to contribute to multi-scale temporal learning without notably increasing model complexity. Moreover, integrating the ensemble layer with the RNN-based layers has been found to significantly improve overall predictive performance. Considering these characteristics, MP-LRNet can be considered a promising foundation for future hybrid forecasting systems requiring high predictive reliability.

The generalizability experiments conducted on the public benchmark dataset provided crucial insights of MP-LRNet. The fact that the performance ranking of the four core model configurations (MP-LRNet > LSTM + RF > LSTM + MP > LSTM-Only) remained completely identical across both datasets robustly validates the consistency of the hybrid architecture. Furthermore, the additional performance gain provided by the multi-patch mechanism over the baseline LSTM + RF model was positive in both domains (+ 0.0115 for furniture and + 0.0135 R^2 for UCI), demonstrating that the multi-patch approach offers a systematic advantage for daily time series containing weekly periodic patterns. When compared to other studies reported on the UCI dataset using daily aggregated data, the R^2 = 0.9201 achieved by MP-LRNet outperformed established models such as CNN-LSTM (R^2 = 0.894) and CNN-GRU (R^2 = 0.922).

5. Conclusion

Accurate time series forecasting is essential for supporting decisions that rely on data, especially in dynamic areas such as retail and energy management. In this study, a novel hybrid deep learning model, MP-LRNet, was proposed to improve the prediction accuracy and stability of temporal data. The model combines recurrent neural networks with ensemble learning algorithms within a modular Multi-Patch architecture designed to capture both short-term and medium-term temporal dependencies. Using a real-world furniture sales dataset, MP-LRNet achieved a maximum R^2 value of 0.9918, showing strong generalization ability and consistent performance across different configurations.

The experimental results indicate that the Multi Patch mechanism helps the model learn temporal features at various time intervals, while the combination of Random Forest and LSTM layers ensures a good balance between accuracy and computational efficiency. This architecture suggests that the joint use of recurrent networks and ensemble learning methods can improve forecasting performance in complex temporal problems.

Furthermore, generalizability experiments conducted on the public UCI Household Electric Power Consumption benchmark dataset confirmed that MP-LRNet exhibited promising forecasting performance (R^2 = 0.9201) even when applied to a completely different domain (energy consumption). The identical component-based performance ranking observed across both datasets strongly substantiates the cross-domain transferability of the proposed MP-LRNet architecture.

Given its strong performance and successful validation in both the retail and energy domains, MP-LRNet exhibits significant potential to be adapted for a wider range of forecasting tasks, such as traffic flow analysis and environmental climate prediction. Future studies may focus on enhancing interpretability and extending the model with architectures that use attention mechanisms and transformer structures to represent temporal information better. Overall, the proposed model provides a solid foundation for future exploration of hybrid neural architectures in time series forecasting.

CRediT authorship contribution statement

Onur Şahin: Writing – original draft, Software, Methodology, Data curation. **Burakhan Çubukçu:** Writing – review & editing, Writing – original draft, Supervision, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data and code availability

The dataset and code that support the findings of this study are openly available in GitHub at https://github.com/Onursahin-ai/MP_LRNet.

References

[1] Gupta A, Agarwal P. Enhancing sales forecasting accuracy through integrated enterprise resource planning and customer relationship management using artificial intelligence. In: Proceedings of the 2024 3rd international conference on artificial intelligence for internet of things (AIIoT 2024). 2024, https://doi.org/10.1109/AIIOT58432.2024.10574785

[2] Eyo-Udo NL, Toromade A, Adewale TT, Toromade AS, Igwe AN, Optimizing food and FMCG supply chains: a dual approach leveraging behavioral finance insights and big data analytics for strategic decision making, Contemp Res Rev J 2024. https://doi.org/10.57219/crrj.2024.2.1.0028

[3] Olaleye I, Olufemi-Phillips A, Olaleye IA, Mokogwu C, Olufemi-Phillips AQ, Adewale TT, Transforming supply chain resilience: frameworks and advancements in predictive analytics and data driven strategies, Open Access Res J Med Sci 2024. https://doi.org/10.53022/oarjms.2024.8.2.0065

[4] Madas MA, Pasupuleti V, Thuraka B, Kodete CS, Malisetty S, Enhancing supply chain agility and sustainability through machine learning: optimization techniques for logistics and inventory management, Logistics Jul 2024;8(3):73. https://doi.org/10.3390/LOGISTICS8030073

[5] Alsoussi A, Tahboub K, Inventory management practices and challenges: an exploratory study, An-Najah Univ J Res - (Nat Sci) 2025, available at [accessed 22 February 2025] https://doi.org/10.35552/anujr.a.40.1.2381

[6] Arqub OA, Mezghiche R, Maayah B, Fuzzy m-fractional integrodifferential models: theoretical existence and uniqueness results, and approximate solutions utilizing the hilbert reproducing kernel algorithm, Front Phys 2023;11. https://doi.org/10.3389/fphy.2023.1252919

[7] Arqub OA, Abukhaled M, Sweis H, Shawagfeh N, Numerical analysis of lucas polynomials innovative technique within the galerkin strategy for solving fuzzy-type differential models with an application in the electrical circuit engineering field, Results Eng 2025;27. https://doi.org/10.1016/j.rineng.2025.106298

[8] Badawi H, Arqub OA, Shawagfeh N, Existence, uniqueness, and numerical analysis of fractional m-derivative stochastic integrodifferential models within the shifted legendre gauss pseudospectral algorithm, Fluct Noise Lett 2025;24(6):2550055, https://doi.org/10.1142/S0219477525500555

[9] Badawi H, Arqub OA, Eid R, Theoretical predictions of existence and uniqueness for stochastic m-fractional differential models of random brownian motion, exhibited with matrix spectral collocation solutions and simulated with the electrical engineering RLC utility as a case interpretation, Fractals 2026;34(2):2540303, https://doi.org/10.1142/S0218348X25403035

[10] Ferdowsy F, Deep learning-based time series prediction techniques [Master's thesis]. Joensuu, Finland: School of Computing, University of Eastern Finland; 2024.

[11] Zhang PG, Time series forecasting using a hybrid ARIMA and neural network model, Neurocomputing Jan 2003;50:159–75. https://doi.org/10.1016/S0925-23120100702-0

[12] Hochreiter S, Schmidhuber J, Long short term memory, Neural Comput Nov 1997;9(8):1735–80. https://doi.org/10.1162/NECO.1997.9.8.1735

[13] Chung J, Gulcehre C, Cho K, Bengio Y, Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, arXiv preprint arXiv:1412.3555, 2014 Available: https://arxiv.org/abs/1412.3555v1 [accessed 22 February 2025].

[14] Breiman L, Random forests, Mach Learn Oct 2001;45(1):5–32. https://doi.org/10.1023/A:1010933404324

[15] Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In: Proc. ACM SIGKDD int. Conf. Knowledge discovery and data mining. Aug 2016. p. 785–94, https://doi.org/10.1145/2939672.2939785

[16] Friedman JH, Stochastic gradient boosting, Comput Stat Data Anal Feb 2002;38(4):367–78. https://doi.org/10.1016/S0167-94730100065-2

[17] Djaballah S, Saidi L, Meftah K, Hechifa A, Bajaj M, Zaitsev I, A hybrid LSTM random forest model with grey wolf optimization for enhanced detection of multiple bearing faults, Sci Rep Oct 2024;14(1):1–18. https://doi.org/10.1038/s41598-024-75174-x

[18] Qin X, Application of deep learning for stock prediction within the framework of portfolio optimization in quantitative trading, Hightech Innov J 2025;6(2):598–614. https://doi.org/10.28991/HIJ-2025-06-02-016

[19] Badykova IR, Dinmukhametova AA, Defining the determinants of corporate financial performance: a machine learning approach, Emerg Sci J 2025;9(4):1764–73. https://doi.org/10.28991/ESJ-2025-09-04-01

[20] Asrol M, Suharjito RJ, An optimized hybrid model for perishable product quality inference in the food supply chain, Emerg Sci J 2025;9(1):485–503. https://doi.org/10.28991/ESJ-2025-09-01-027

[21] Woo G, Liu C, Kumar A, Xiong C, Savarese S, Sahoo D. Unified training of universal time series forecasting transformers. In: Proc. 41st int. Conf. Machine learning (ICML 2024). vol. 235, 2024. Available: https://ink.library.smu.edu.sg/sis_research/9906 [accessed 22 February 2025].

[22] Nie Y, Nguyen NH, Sinthong P, Kalagnanam J, A Time Series is Worth 64 Words: Long-term Forecasting with Transformers, arXiv preprint, 2023 arXiv:2211.14730.

[23] Kabir MR, Bhadra D, Ridoy M, Milanova M, LSTM transformer based robust hybrid deep learning model for financial time series forecasting, Sci Jan 2025;7(1):7. https://doi.org/10.3390/SCI7010007

[24] Mariappan Y, Ramasamy K, Velusamy D, An optimized deep learning based hybrid model for prediction of daily average global solar irradiance using CNN SLSTM architecture, Sci Rep Dec 2025;15(1):1–19. https://doi.org/10.1038/S41598-025-95118-3

[25] Ma C, Hu Y, Xu X, Hybrid deep learning model with VMD-BiLSTM-GRU networks for short-term traffic flow prediction, Data Sci Manag Nov 2024. https://doi.org/10.1016/J.DSM.2024.10.004

[26] Shi J, Wang S, Qu P, Shao J, Time series prediction model using LSTM transformer neural network for mine water inflow, Sci Rep Dec 2024;14(1):1–16. https://doi.org/10.1038/S41598-024-69418-Z

[27] Abumohsen M, Owda AY, Owda M, Abumihsan A, Hybrid machine learning model combining CNN LSTM RF for time series forecasting of solar power generation, e-Prime - Adv Electr Eng Electron Energy Sept 2024;9:100636, https://doi.org/10.1016/J.PRIME.2024.100636

[28] Guo W, Liu S, Weng L, Liang X, Power grid load forecasting using a CNN LSTM network based on a multi modal attention mechanism, Appl Sci (Switzerland), vol. Mar 2025;15(5):2435. https://doi.org/10.3390/APP15052435

[29] He Z, Guo Q, Wang Z, Li X, A hybrid wavelet based deep learning model for accurate prediction of daily surface pm2.5 concentrations in guangzhou city, Toxics Mar 2025;13(4):254. https://doi.org/10.3390/TOXICS13040254

[30] Salman D, Direkoglu C, Altanneh N, Ahmed A, Hybrid wavelet LSTM transformer model for fault forecasting in power grids, SSRG Int, J Electr Electron Eng Dec 2024;11(12):314–26. https://doi.org/10.14445/23488379/IJEEE-V11I12P130

[31] Zhang J, Yan L, GRU enhanced attention mechanism for LSTM in hybrid CNN LSTM models for stock prediction, J Glob Trends Soc Sci 2025;2(3). https://doi.org/10.70731/RZVS8J53

[32] Wen X, Liao J, Niu Q, Shen N, Bao Y, Deep learning driven hybrid model for short term load forecasting and smart grid information management, Sci Rep Dec 2024;14(1):1–16. https://doi.org/10.1038/S41598-024-63262-X

[33] Dakheel F, Çevik M, Optimizing smart grid load forecasting via a hybrid long Short-Term Memory-XGBoost framework: enhancing accuracy, robustness, and energy management, Energies 2025;18(11):2842. https://doi.org/10.3390/en18112842

[34] Quang PD, Thang TQ, Analysis and forecasting of daily global gold price: an SARIMA-LSTM approach with random forest technique, Cogent Econ Finance 2025;13(1):2568969, https://doi.org/10.1080/23322039.2025.2568969

[35] Moritz S, Bartz-Beielstein T, imputeTS: time series missing value imputation in r, The R Journal Jun 2017 9(1). 207–18. Available: https://digitalcommons.unl.edu/r-journal [accessed 20 April 2025]..

[36] Hebrail G, Berard A, Individual household electric power consumption [Dataset], UCI Machine Learning Repository, https://doi.org/10.24432/C58K54. 2012.

[37] Oreshkin BN, Carpov D, Chapados N, Bengio Y. N-BEATS: neural basis expansion analysis for interpretable time series forecasting. In: Int. Conf. Learning representations (ICLR. 2020. arXiv preprint arXiv:1905.10437.

[38] Fan Y, Tang Q, Guo Y, Wei Y, BiLSTM MLAM: a multi scale time series prediction model for sensor data based on bi LSTM and local attention mechanisms, Sensors Jun 2024;24(12):3962. https://doi.org/10.3390/S24123962

[39] Rivas F, Sierra-Garcia JE, Camara JM, Comparison of LSTM and GRU type RNN networks for attention and meditation prediction on raw EEG data from

low cost headsets, Electronics Feb 2025;14(4):707. https://doi.org/10.3390/ELECTRONICS14040707
[40] Wang X, et al., The application of a BiGRU model with transformer based error correction in deformation prediction for bridge SHM, Buildings Feb 2025;15(4):542. https://doi.org/10.3390/BUILDINGS15040542
[41] Jiang K, Huang Z, Zhou X, Tong C, Zhu M, Wang H, Deep belief improved bidirectional LSTM for multivariate time series forecasting, Math Biosci Eng (MBE) 2023;20(9):16596–627. https://doi.org/10.3934/mbe.2023739
[42] Srivastava N, Hinton G, Krizhevsky A, Salakhutdinov R, Dropout: a simple way to prevent neural networks from overfitting, J Mach Learn Res 2014;15:1929–58.
[43] Kingma DP, Ba J, Adam: A Method for Stochastic Optimization, arXiv preprint, 2015 arXiv:1412.6980.
[44] Salman HA, Kalakech A, Steiti A, History A, Random forest algorithm overview, Babylon J Mach Learn Jun 2024;2024:69–79. https://doi.org/10.58496/BJML/2024/007
[45] Rizkallah LW, Enhancing the performance of gradient boosting trees on regression problems, J Big Data Dec 2025;12(1):1–34. https://doi.org/10.1186/S40537-025-01071-3
[46] Hu T, Financial fraud detection system based on improved random forest and gradient boosting machine (GBM). In: 2025 5th Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS); 2025. p. 761–65. https://doi.org/10.1109/ACCTCS66275.2025.00137.
[47] Chicco D, Warrens MJ, Jurman G, The coefficient of determination r-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation, Peerj Comput Sci 2021. https://doi.org/10.7717/peerj-cs.623
[48] Chai T, Draxler RR, Root mean square error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE in the literature, Geosci Model Dev 2014;7:1247–50. https://doi.org/10.5194/gmd-7-1247-2014
[49] Čalasan M, Aleem SHEA, Zobaa AF, On the root mean square error (RMSE) calculation for parameter estimation of photovoltaic models: a novel exact analytical solution based on Lambert W function, Energy Convers Manag 2020;210:112716.
[50] De Myttenaere A, Golden B, Grand BL, Rossi F, Mean absolute percentage error for regression models, 2016. Available: https://www.datascience.net [accessed 20 April 2025].
[51] Sherly A, Christo MS, Elizabeth JV, A hybrid approach to time series forecasting: integrating ARIMA and prophet for improved accuracy, in: Results in engineering, Elsevier), 2025, p. 105703.
[52] Mahmoud A, Mohammed A, Leveraging hybrid deep learning models for enhanced multivariate time series forecasting, Neural Process Lett Oct 2024;56(5):1–25. https://doi.org/10.1007/S11063-024-11656-3
[53] Xu Y, Zhang W, Lai H, Zhang S, Xiao R, MPMixer: multi-scale patch mixer for long-term time series forecasting, Neurocomputing Sept 2025;646:130398, https://doi.org/10.1016/J.NEUCOM.2025.130398
[54] Balusani A, Chowdary PJR, Paruchuri BVNP, Enhancing retail demand forecasting with XGBoost: a comparative study with machine learning and deep learning models, SSRN Electron J 2025. https://doi.org/10.2139/SSRN.5274510
[55] Alsalem K, A hybrid time series forecasting approach integrating fuzzy clustering and machine learning for enhanced power consumption prediction, Sci Rep Dec 2025;15(1):1–16. https://doi.org/10.1038/S41598-025-91123-8

Author biography

Onur Şahin was born in Bursa, Turkey, in 2000. He received his B.S. degree in Computer Engineering from Düzce University, Turkey, in 2023. In the same year, he was accepted into the M.S. program in Computer Engineering at Bilecik Şeyh Edebali University, where he is currently pursuing his graduate studies. He works as a Digitalization Specialist at an international furniture company located in İnegöl, Bursa, Turkey. His research interests include deep learning, time series analysis, artificial intelligence, data analytics, and business intelligence.

Burakhan Çubukçu was born in Ankara, Turkey, in 1988. He received the B.S. degree from the Mathematics Department, Hacettepe University, Ankara, Turkey, in 2012, and Department of Business Administration, Anadolu University, Eskişehir, Turkey in 2010. He received the M.S. degree from the Information Systems Department, Gazi University, Ankara, Turkey, in 2013. In 2020, he received Ph.D. degree from Department of Electronics and Computer Engineering, Bilecik Şeyh Edebali University, Bilecik, Turkey. Since 2021, he has been an Assistant Professor with the Computer Engineering Department, Faculty of Engineering, Bilecik Şeyh Edebali University, Turkey. His research interests include deep learning, telerehabilitation, computer applications, metaheuristic algorithms and intelligent systems.