

Deep mixture network for uncertainty quantification in time series forecasting

Wang Ruihuan, Wang Yiqun, Meng Xintong, Zhao Keqing, Chen Wenbai (✉)

College of Automation (College of Artificial Intelligence), Beijing Information Science and Technology University, Beijing 100192, China

Abstract

A novel uncertainty-aware forecasting model, termed Monte Carlo (MC) dropout-convolutional neural network-Kolmogorov-Arnold network-long short-term memory (MCKLNet), is proposed for high-volatility time series forecasting. The model combines a convolutional neural network (CNN) for local feature extraction, a convolutional Kolmogorov-Arnold network (KAN) for feature enhancement, long short-term memory (LSTM) networks for temporal dependency modeling, and MC dropout for Bayesian uncertainty quantification. It aims to address three major challenges in volatile time series prediction: insufficient local feature extraction, inadequate long-term dependency modeling, and limited reliability of probabilistic uncertainty estimation. To evaluate its effectiveness, pork price forecasting is employed as a representative case study due to its high volatility and sensitivity to multiple influencing factors. Experimental results show that MCKLNet consistently outperforms conventional baseline models in both point prediction accuracy and uncertainty interval calibration, demonstrating strong robustness, interpretability, and potential for broader application in economic forecasting, energy demand prediction, financial market analysis, and other complex sequential prediction tasks.

Keywords Monte Carlo (MC) dropout, uncertainty quantification, time series forecasting, convolutional neural network-Kolmogorov-Arnold network-long short-term memory (MCKLNet), probabilistic forecasting

1 Introduction

Accurate forecasting of high-volatility time series is essential in numerous domains, including economics, energy, finance, and environmental science. However, current forecasting models face three major challenges: inefficiency in extracting local features^[1], inadequate modeling of long-term dependencies^[2], and limited accuracy in probabilistic uncertainty quantification^[3]. These limitations substantially hinder

the performance and generalizability of traditional time series models in complex and dynamic environments.

Traditional approaches, such as the autoregressive integrated moving average (ARIMA) model and classical time series decomposition, perform well on stationary data but often fail to capture the high volatility and nonlinear dynamics present in real-world scenarios. To overcome these limitations, deep learning-based methods receive increasing attention. Recurrent neural networks (RNNs) and their variants, including LSTM networks and gated recurrent units (GRUs)^[4–5], show strong capabilities in modeling complex temporal dependencies.

CNNs are extensively utilized in time series

forecasting due to their capability to extract localized temporal patterns through sliding convolution operations^[6]. By learning hierarchical representations from sequential data, CNNs effectively capture short-term fluctuations, seasonal trends, and fine-grained structural variations. The transformer architecture, originally introduced by Vaswani et al. for machine translation, is adapted for time series applications to overcome the limitations of traditional recurrent mechanisms^[7]. Its self-attention mechanism facilitates parallel computation and enables the modeling of long-range dependencies without relying on recurrence. Zhou et al.^[8] demonstrated that, with temporal embeddings and architectural modifications, transformer-based models can outperform conventional recurrent networks on complex sequential datasets.

In the domain of uncertainty modeling, MC dropout provides an efficient approximation to Bayesian inference^[9]. By interpreting dropout as a form of variational approximation, this method enables neural networks to produce probabilistic predictions without altering their underlying architecture^[9]. Deep ensembles further enhance generalization and robustness by aggregating outputs from multiple independently trained models, particularly under conditions of distributional shift and data variability^[10].

Bayesian LSTM networks incorporate weight uncertainty through posterior sampling, offering a principled model for capturing both aleatoric and epistemic uncertainty in sequential forecasting tasks^[11]. To improve adaptability in complex environments, recent studies have investigated the integration of heterogeneous data sources. For instance, Zhao et al. proposed a multi-channel LSTM model that fuses historical price data, sentiment signals, and macroeconomic indicators to enhance forecasting responsiveness^[12]. Similarly, Wu et al. developed a graph-augmented LSTM model that leverages graph structures to encode dependencies among features, facilitating more accurate modeling of interactions within heterogeneous dynamic systems^[13].

Despite significant progress in deep learning, external information integration, and uncertainty

quantification, several critical challenges remain in practical high-volatility time series forecasting. First, most existing models primarily emphasize point estimates and often fail to accurately capture the uncertainty within prediction intervals, offering limited support for assessing the risks associated with fluctuation ranges^[14]. Second, many approaches inadequately model long-term dependencies or overlook local patterns, making it difficult to strike a balance between capturing overarching trends and accounting for short-term perturbations^[15]. Third, issues such as high computational overhead, sensitivity to data variability, and insufficient evaluation of model generalizability continue to limit the deployment of these methods in complex, real-world forecasting environments^[16]. To address these challenges, a hybrid MCKLNet model is designed to enhance predictive performance and uncertainty modeling capabilities for high-volatility time series across dynamic domains. The main contributions of this work are as follows.

1) A novel hybrid model, MCKLNet, is proposed, integrating convolutional local feature extraction, convolutional KAN, LSTM-based long-term dependency modeling, and MC dropout for Bayesian uncertainty estimation, providing an end-to-end solution for high-volatility time series forecasting.

2) Convolutional KAN is introduced into the time series domain, significantly enhancing the model's sensitivity to key fluctuation features and improving its representational capacity.

3) A comprehensive uncertainty evaluation mechanism is incorporated, employing prediction interval coverage probability (PICP) alongside traditional metrics such as mean absolute percentage error (MAPE), root mean square error (RMSE), mean absolute error (MAE), and the coefficient of determination R^2 to assess both predictive accuracy and uncertainty quantification.

4) The effectiveness and generalizability of the MCKLNet model are validated through comparative experiments and ablation studies, with pork price fluctuations selected as a representative high-volatility case study, demonstrating broad applicability across dynamic economic and industrial forecasting tasks.

2 Method and principle

This section describes the key components of the proposed MCKLNet model, including the basic principles of LSTM for temporal dependency modeling, CNN for local feature extraction, KAN for critical feature weighting, and MC dropout for uncertainty estimation.

2.1 LSTM

LSTM was first proposed by Hochreiter et al. in 1997 as a variant of RNNs designed to effectively model long-term dependencies in sequential data^[17]. Its key innovation lies in incorporating a memory unit and a gating mechanism, which alleviates the issues of gradient vanishing and explosion that commonly occur in traditional RNNs^[18–19].

In the MCKLNet model, LSTM is used to capture temporal dependencies in the feature sequences extracted by the preceding layers. As shown in Fig. 1, $\mathbf{x}_t$ denotes the input feature vector at the t th time step, $\mathbf{c}_{t-1}$ and $\mathbf{c}_t$ denote the previous and current cell states, $\mathbf{h}_{t-1}$ and $\mathbf{h}_t$ denote the previous and current hidden states, $\mathbf{f}_t$, $\mathbf{i}_t$, and $\mathbf{o}_t$ represent the forget gate, input gate, and output gate, respectively, $\tilde{\mathbf{c}}_t$ denotes the candidate cell state, $\sigma(\cdot)$ denotes the sigmoid activation function, and $\tanh$ denotes the hyperbolic tangent activation function. Through these components, the LSTM module can selectively retain historical information and incorporate new input information for temporal modeling.

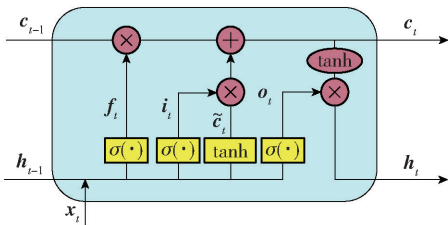


Fig. 1 Structure of LSTM model

2.2 CNN

CNN was originally introduced by LeCun et al.^[20], has been widely adopted for feature extraction in structured data, including image recognition and time series analysis. In agricultural domains, CNNs are

used in tasks such as pest detection and yield prediction^[21]. In the MCKLNet model, CNN acts as the front-end processor to extract local patterns from input sequences, laying the foundation for subsequent feature transformations.

The convolutional layer applies one-dimensional (1D) kernels along the temporal axis to extract short-term patterns, local trends, and fluctuations in the input^[22], as shown in Fig. 2.

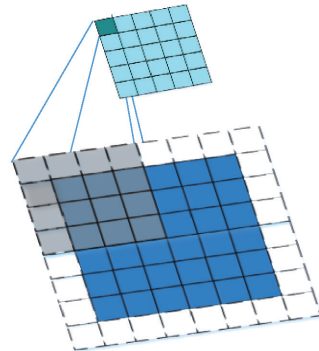


Fig. 2 Convolution layer principle

Pooling operations such as maximum pooling are applied to compress feature maps, reduce noise, and improve robustness^[23], as illustrated in Fig. 3. After several convolution and pooling operations, the fully connected layer synthesizes multi-scale local features into global representations. This enables the model to make informed inferences about overall trends^[24]. The structure is illustrated in Fig. 4.

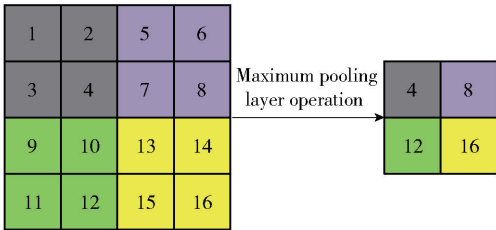


Fig. 3 Pool layer principle

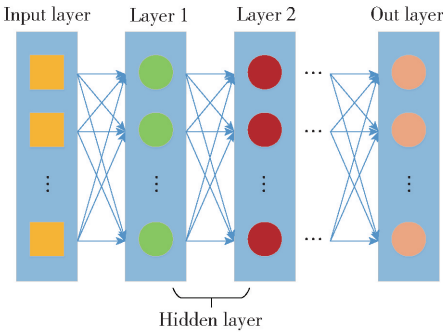


Fig. 4 Principle of fully connected layer

2.3 KAN

KAN is a novel neural network architecture inspired by the Kolmogorov-Arnold representation theorem^[25]. Unlike the traditional structure of a multilayer perceptron (MLP), KAN eliminates the conventional linear combination step and instead applies nonlinear activation functions directly to the inputs, followed by additive operations at the edges. A key distinction of this architecture is that the learned activation functions are applied to the edges (weights), while the nodes function purely as summation units.

The KAN network comprises multiple learnable functions and can be expressed as a two-layer formulation. In the first layer, univariate functions apply nonlinear transformations directly to the input variables. The second layer aggregates these transformed outputs to produce the final result. This design enables the KAN network to flexibly capture complex functional relationships, thereby enhancing both its expressive capacity and adaptability.

With this innovative design, KAN performs complex computations more efficiently by applying activation functions to the weight layers and integrating multilayer functional compositions^[26–27], thereby enhancing learning efficiency and effectively approximating complex multivariable functions. The core mechanism of KAN is illustrated in Fig. 5.

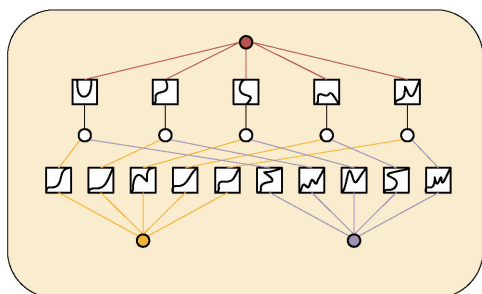


Fig. 5 Structure of KAN model

In the MCKLNet model, a convolutional variant of KAN is integrated following the CNN layers to enhance the nonlinear transformation of local features. The CNN module first extracts short-term temporal patterns from the input sequence, while the KAN module applies flexible and interpretable transformations to these

features using learnable spline functions. This architecture enables the model to approximate complex variable relationships that traditional MLP-based models struggle to capture.

By placing the KAN module between the CNN and LSTM layers, its functional approximation capability is leveraged to enrich feature representations prior to temporal modeling. This design enhances forecasting accuracy and improves uncertainty estimation, as demonstrated in experiments in Ref. [28].

2.4 MC dropout

Dropout is a widely used technique for preventing overfitting in deep neural networks by randomly deactivating neurons during training^[29–30]. This stochastic regularization enhances generalization by reducing co-adaptation among neurons, as illustrated in Fig. 6.

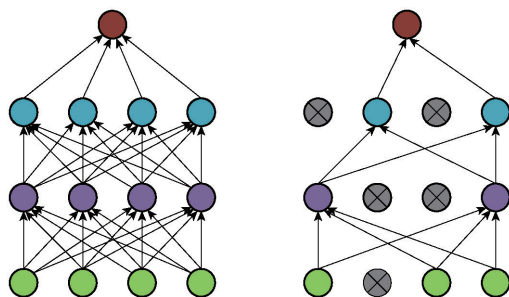


Fig. 6 Schematic diagram of MC dropout

Beyond regularization, recent studies have shown that dropout can also function as an approximate Bayesian inference method. Damianou et al.^[31] demonstrated that applying dropout to every layer of a neural network approximates a probabilistic deep Gaussian process. Hennighausen et al.^[32] further proved that this procedure is mathematically equivalent to performing approximate variational inference, thereby enabling uncertainty estimation through sampling.

Leveraging this insight, the MCKLNet model incorporates MC dropout during inference to quantify epistemic uncertainty. Specifically, MC dropout is applied to both the LSTM and KAN modules during training and testing. By performing multiple stochastic forward passes with dropout enabled, the model conducts MC integration to yield a predictive

distribution. This not only enhances robustness but also provides confidence intervals, enabling uncertainty-aware decision-making in scenarios such as pork price

forecasting. The overall model of this method integrated with LSTM is illustrated in Fig. 7.

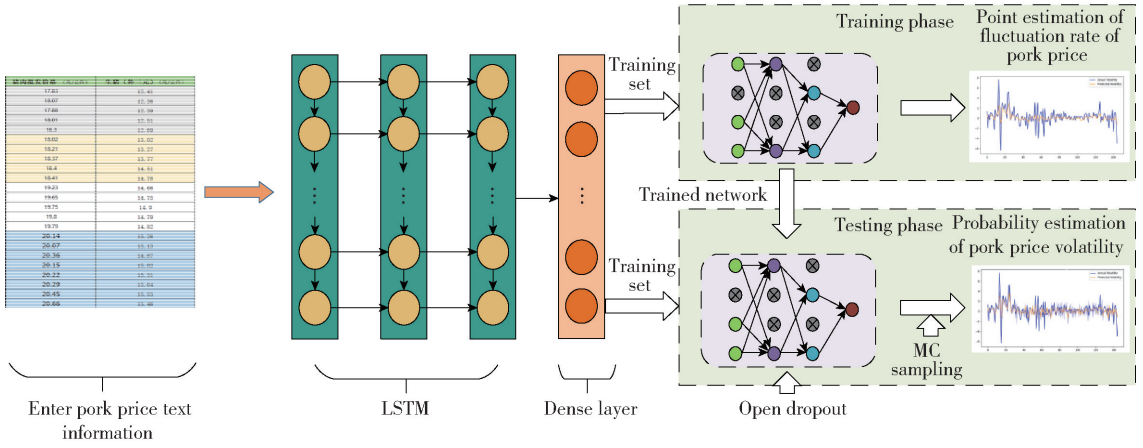


Fig. 7 MC dropout prediction model

2.5 MCKLNet

To overcome the limitations of conventional RNNs, such as weak hierarchical representation and slow convergence, a hybrid architecture named MCKLNet is

proposed, which integrates CNN, LSTM, and KAN. Additionally, MC dropout is incorporated for uncertainty-aware prediction. The full architecture is illustrated in Fig. 8.

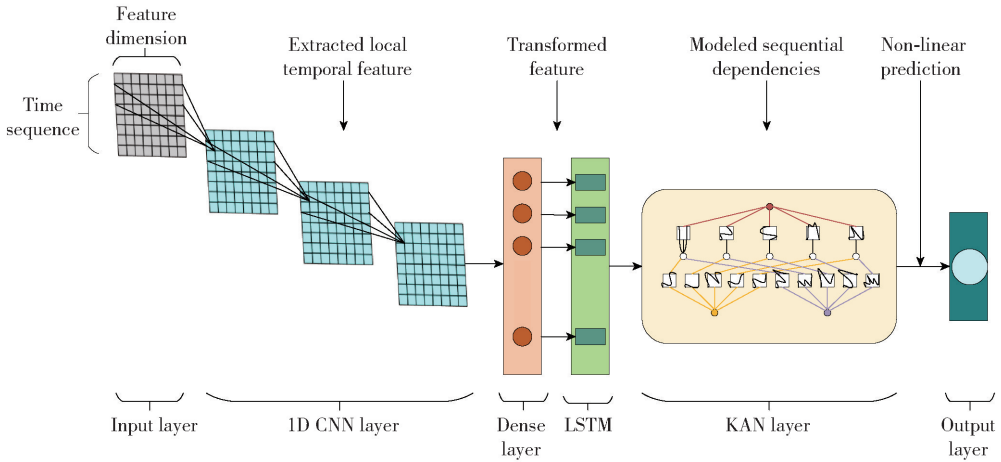


Fig. 8 MCKLNet hybrid model

In this paper, scalars are denoted by italic letters, vectors are denoted by bold italic lowercase letters, and matrices are denoted by bold italic uppercase letters. Let the input multivariate time series be denoted by $X \in \mathbb{R}^{T \times D}$, where T is the number of time steps and D is the number of input features. MCKLNet processes the input through the following modules.

First, a 1D convolutional layer is applied to X to extract local temporal features.

$$F = \Phi_{\text{CNN}}(X); \quad F \in \mathbb{R}^{T' \times C} \quad (1)$$
where $\Phi_{\text{CNN}}(\cdot)$ denotes the nonlinear mapping implemented by the CNN module, T' denotes the temporal length after convolution or pooling, and C denotes the number of output channels. Thus, F represents the local feature maps extracted by the CNN module.

Next, the extracted features are projected into a hidden representation through a fully connected layer

with rectified linear unit (ReLU) activation.

$$\mathbf{H} = \delta_{\text{ReLU}}(\mathbf{W}_H \mathbf{F} + \mathbf{b}_H); \quad \mathbf{H} \in \mathbb{R}^{T' \times d_h} \quad (2)$$

where $\delta_{\text{ReLU}}(\cdot)$ denotes the ReLU activation function, $\mathbf{W}_H$ and $\mathbf{b}_H$ are trainable parameters, and d_h denotes the hidden feature dimension.

The projected sequence $\mathbf{H}$ is then fed into the LSTM module to model temporal dependencies. At the t th time step, let $\mathbf{x}_t$ denote the input feature vector, $\mathbf{h}_{t-1}$ and $\mathbf{h}_t$ denote the previous and current hidden state vectors, and $\mathbf{c}_{t-1}$ and $\mathbf{c}_t$ denote the previous and current cell state vectors. The LSTM cell is updated as

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \quad (3)$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \quad (4)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \quad (5)$$

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

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c) \quad (7)$$

$$\mathbf{h}_t = \mathbf{o}_t \tanh \mathbf{c}_t \quad (8)$$

where $\mathbf{f}_t$, $\mathbf{i}_t$, and $\mathbf{o}_t$ are the forget-gate, input-gate, and output-gate vectors, respectively. And $\mathbf{W}_f$, $\mathbf{W}_i$, $\mathbf{W}_o$, and $\mathbf{W}_c$ denote the trainable weight matrices of the forget gate, input gate, output gate, and candidate cell state, respectively, while $\mathbf{b}_f$, $\mathbf{b}_i$, $\mathbf{b}_o$, and $\mathbf{b}_c$ denote the corresponding bias vectors.

The LSTM output is denoted by $\mathbf{H}^{\text{LSTM}}$, which is used as the input to the subsequent KAN module. To enhance representation capability and interpretability, KAN is employed to apply adaptive univariate nonlinear transformations.

$$\hat{y} = \sum_{j=1}^J \phi_j \left(\sum_{m=1}^M \psi_{m,j}(\mathbf{H}_m^{\text{LSTM}}) \right); j = 1, 2, \dots, J, m = 1, 2, \dots, M \quad (9)$$

where $\hat{y}$ is the predicted output, M denotes the input feature dimension of the KAN module, J denotes the number of output nodes, $\psi_{m,j}(\cdot)$ are spline-based internal functions, $\phi_j(\cdot)$ are external aggregation functions, and $\mathbf{H}_m^{\text{LSTM}}$ denotes the m th component of $\mathbf{H}^{\text{LSTM}}$.

To quantify prediction uncertainty, dropout is applied during inference. By performing K stochastic forward passes, the predictive mean $E(\cdot)$ and variance $V[\cdot]$ are estimated as

$$E[\hat{y}] \approx \frac{1}{K} \sum_{k=1}^K \hat{y}^{(k)} \quad (10)$$

$$V[\hat{y}] \approx \frac{1}{K} \sum_{k=1}^K (\hat{y}^{(k)} - E[\hat{y}])^2 \quad (11)$$

where $\hat{y}^{(k)}$ denotes the prediction obtained from the k th stochastic forward pass, and K denotes the total number of MC samples.

For an input sequence of shape (T, D) , the CNN module outputs local feature maps with a reduced temporal length T' and channel size C . These are then projected to a hidden representation of shape (T', d_h) , which is passed into the LSTM layer to capture sequential dependencies. The KAN module applies nonlinear spline-based transformations to the LSTM outputs, and the results are passed through MC dropout to generate predictive distributions.

This modular design combines the strengths of CNN for local pattern extraction, LSTM for temporal dependency modeling, KAN for flexible nonlinear function approximation, and MC dropout for uncertainty estimation, resulting in a unified and interpretable forecasting architecture.

Moreover, the modularity of MCKLNet makes it easy to extend or substitute individual components, enabling broad applicability to other high-volatility time series forecasting tasks beyond pork price prediction.

3 Experiment and analysis

3.1 Dataset

To validate the effectiveness of the MCKLNet model, a representative economic time series dataset characterized by high volatility and complex dependency structures is selected. The dataset spans the period from November 2, 2020, to November 3, 2022, comprising 733 daily observations. It includes eight variables: outer ternary crossbreed pig price (OTC, CNY/kg), inner ternary crossbreed pig price (ITC, CNY/kg), black pig price (BP, CNY/kg), corn price (CP, CNY/kg), soybean meal price (SMP, CNY/kg), average wholesale pork price (AWPP, CNY/kg), consumer price index (CPI, base is 100), and pig slaughter volume (PSV, 10^4 head/day).

The core market variables, including OTC, ITC,

BP, CP, SMP, AWPP, and PSV, are obtained from the China Pig Network, a widely recognized and frequently updated platform in Chinese pig industry known for its comprehensive data coverage. The CPI data are sourced from the National Bureau of Statistics of China, ensuring the credibility and consistency of the macroeconomic statistics.

Since both CPI and PSV are reported on a monthly basis, frequency alignment is required to integrate them with the other daily-resolution variables. For CPI, a linear interpolation method is applied to estimate intra-month values, based on the assumption that consumer prices evolve smoothly over time. This approach is supported by prior empirical studies, such as Refs. [33 – 34], where linear or spline interpolation techniques are employed to align lower-frequency macroeconomic indicators in mixed-frequency data models. For PSV, an equal distribution method is adopted to evenly allocate the total monthly volume across constituent days, maintaining the integrity of the original metric while ensuring appropriate temporal resolution for daily time series modeling^[35]. These preprocessing strategies ensure that the dataset remains complete, coherent, and consistent with actual data generation mechanisms, while minimizing potential biases introduced by temporal disaggregation.

By integrating heterogeneous market, cost, supply, and macroeconomic signals into a unified daily-resolution dataset, this case study establishes a challenging and representative testing environment to rigorously validate the robustness, adaptability, and uncertainty modeling capabilities of the MCKLNet model.

Missing values in the dataset may adversely affect the accuracy and reliability of sequential modeling and uncertainty estimation tasks. Therefore, it is crucial to address missing data effectively. In this study, a multivariate feature interpolation approach is employed using the Scikit-learn library in Python. For each feature with missing values, correlated features are treated as independent variables to iteratively predict

and impute the missing entries, ensuring consistency within the dataset.

In order to ensure the stability of the training and the applicability of the model, the data in Fig. 9 is divided. The data is divided into a training set and a test set. The training set contains the first 586 rows of data, and the test set contains the last 147 rows of data. To further analyze the relationships among variables and their potential impact on prediction results, the Pearson correlation coefficient is utilized to construct a correlation matrix between variables, which is visualized in the form of a heatmap shown in Fig. 9. The calculation formula of Pearson correlation coefficient is concluded as

$$\rho_{p,q} = \frac{\sum_{n=1}^N (z_n^{(p)} - \bar{z}^{(p)}) (z_n^{(q)} - \bar{z}^{(q)})}{\sqrt{\sum_{n=1}^N (z_n^{(p)} - \bar{z}^{(p)})^2} \sqrt{\sum_{n=1}^N (z_n^{(q)} - \bar{z}^{(q)})^2}} \quad (12)$$

where $\rho_{p,q}$ denotes the Pearson correlation coefficient between the p th and q th variables, N is the total number of samples, $z_n^{(p)}$ and $z_n^{(q)}$ denote the observed values of the p th and q th variables at the n th sample, respectively, and $\bar{z}^{(p)}$ and $\bar{z}^{(q)}$ denote the corresponding sample means.

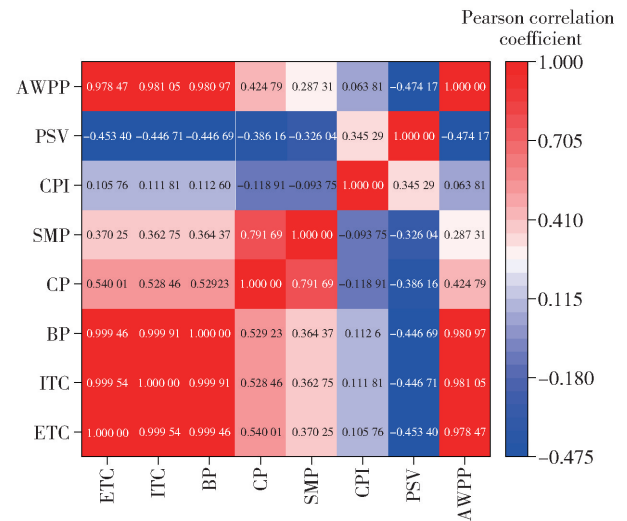


Fig. 9 Pearson correlation coefficient heatmap of variables

There exists a strong positive correlation between AWPP and the three pig price variables: OTC, ITC,

and BP, with $\rho_{p,q} \geq 0.996$, indicating that these serve as the dominant direct pricing indicators in the market. CP and SMP show moderate correlations with AWPP, which aligns with their role as indirect cost-related influencing factors. Although CPI and PSV exhibit relatively weaker linear correlations with AWPP, they still provide explanatory value due to their roles in the macroeconomic context and production supply, respectively.

Moreover, interactions among the variables, particularly between pig prices and feed costs, or between supply and macro indicators, suggest that the presence of complex coupling effects is not fully captured by linear correlation alone. These coupled relationships may exert lagged or nonlinear influences on price dynamics, which are expected to be effectively modeled by the application of deep neural network architectures employed in subsequent stages of this investigation.

In conclusion, the constructed multidimensional high-frequency dataset integrates key market pricing, production cost, supply-side activity, and macroeconomic trends, forming a robust and representative foundation for validating the effectiveness, robustness, and uncertainty modeling capabilities of the MCKLNet model.

3.2 Evaluation index

In the prediction and regression task of continuous values, it is very important to evaluate the accuracy and reliability of the model. In this paper, MAE, RMSE, MAPE, and R^2 are used to measure sexual energy.

$$R_{\text{MAE}} = \frac{1}{N} \sum_{n=1}^N |y_n - \hat{y}_n| \quad (13)$$

$$R_{\text{RMSE}} = \sqrt{\frac{1}{N} \sum_{n=1}^N (y_n - \hat{y}_n)^2} \quad (14)$$

$$R_{\text{MAPE}} = \frac{1}{N} \sum_{n=1}^N \left| \frac{y_n - \hat{y}_n}{y_n} \right| \quad (15)$$

$$R^2 = 1 - \frac{\sum_{n=1}^N (y_n - \hat{y}_n)^2}{\sum_{n=1}^N (y_n - \bar{y})^2} \quad (16)$$

where N denotes the total number of samples, y_n and $\hat{y}_n$ denote the ground-truth value and the model output of the n th sample, respectively, and $\bar{y}$ denotes the mean of the ground-truth values.

However, the three metrics MAE, RMSE, and MAPE may not fully evaluate the effectiveness of the model. To evaluate uncertainty quantification, PICP is used. The prediction interval is constructed as

$$\begin{cases} l_n = \hat{y}_n - \lambda \hat{v}_n \\ u_n = \hat{y}_n + \lambda \hat{v}_n \end{cases} \quad (17)$$

$$R_{\text{PICP}} = \frac{1}{N} \sum_{n=1}^N I\{l_n \leq y_n \leq u_n\} \quad (18)$$

where l_n and u_n denote the lower and upper bounds of the prediction interval for the n th sample, respectively. $\lambda = 1.96$ under the 95% confidence level (95% CI) corresponding to the two-tailed critical value of the standard normal distribution. $\hat{v}_n$ denotes the predictive standard deviation of the n th sample, and $I\{\cdot\}$ is the indicator function.

4 Analysis of experimental result

In economic research, growth indicators are commonly employed to represent the overall trends in market or economic activity expansion. In contrast, commodity price indicators often demonstrate pronounced volatility, which not only reflects immediate supply-demand imbalances but also reveals the long-term evolution of price dynamics. Volatility is a critical factor in understanding and forecasting market behavior, as it captures fluctuation patterns and structural variations across multiple temporal scales. Due to its ability to characterize both short-term disturbances and long-term trends, volatility is widely utilized in forecasting models and early warning systems across various domains. The specific transformation method for computing volatility is presented as

$$D_a = (\ln P_a - \ln P_{a-1}) \times 100\% \quad (19)$$

where D_a denotes the price volatility on the a th day, and P_a , P_{a-1} are prices on the a th day and the $(a-1)$ th day respectively.

4.1 Comparison of uncertainty models

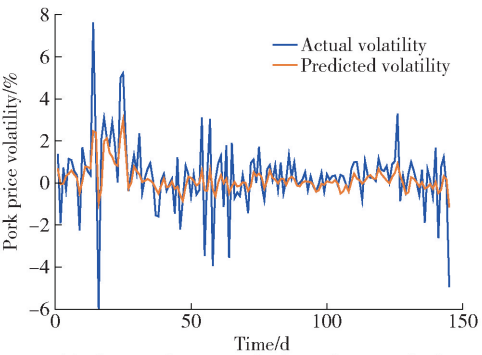
To validate the superiority of MC dropout uncertainty modeling in this study, a comparative analysis among four models: LSTM, Bayesian LSTM (BAYES-LSTM), variational autoencoder LSTM (VAE-LSTM), and MC dropout-LSTM (MC-LSTM). Several evaluation metrics are used, MAPE, RMSE, MAE, PICP, and R^2 , to assess the predictive performance and uncertainty coverage.

Table 1 shows that, compared with the standard LSTM, three models incorporating uncertainty estimation provide effective quantification through prediction intervals. Among them, BAYES-LSTM demonstrates limited generalization. Although it theoretically reduces epistemic uncertainty via Bayesian inference, the complexity of its estimation process can hinder resulting in higher prediction errors. VAE-LSTM, which captures latent variability through variational inference, performs more consistently, but its improvements are modest. In contrast, MC-LSTM delivers the best overall performance, significantly reducing prediction errors and yielding well-calibrated uncertainty estimates. These findings highlight the clear advantage of dropout-based regularization in enhancing both predictive accuracy and uncertainty representation for time series forecasting.

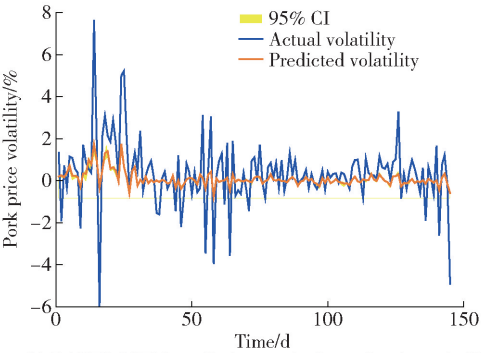
Table 1 Comparison of uncertainty model

Model	MAPE/%	RMSE	MAE	PICP	R^2
LSTM	13.26	4.77	4.18	–	–0.54
BAYES-LSTM	17.10	6.09	5.38	0.02	–1.51
VAE-LSTM	11.81	4.23	3.71	0.11	–0.22
MC-LSTM	8.82	4.13	1.62	0.28	0.10

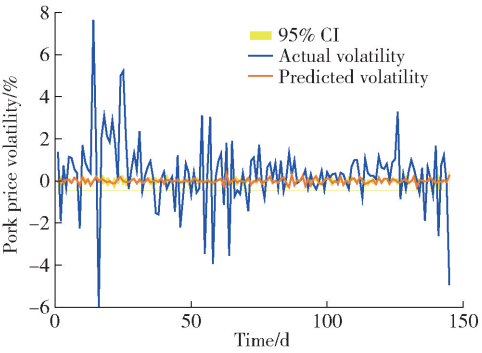
Compared to Figs. 10 (b) – 10 (d) extend point predictions to interval forecasts by incorporating confidence bands, MC dropout not only provides point estimates but also captures the potential range of volatility. Such representation is particularly crucial in applications involving price fluctuations and market forecasting, where quantifying uncertainty is essential.



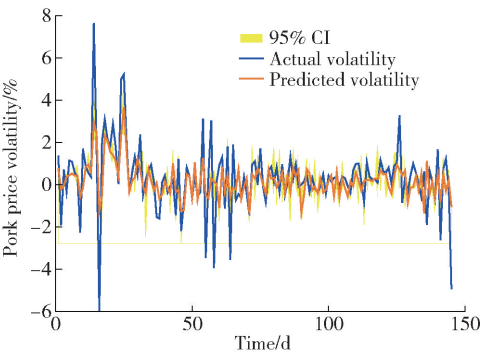
(a) LSTM prediction results for pork price volatility



(b) BAYES-LSTM prediction results for pork price volatility



(c) VAE-LSTM prediction results for pork price volatility



(d) MC-LSTM prediction results for pork price volatility

Fig. 10 Comparison of uncertainty model

The inclusion of confidence intervals enables the model to convey the reliability of its predictions,

thereby supporting more informed and robust decision-making. The results show that MC dropout not only enhances the model’s predictive capability but also quantifies prediction uncertainty, providing valuable insights for practical applications. This enhances the model’s generalization ability and practicality, particularly in environments with high uncertainty.

4.2 Comparison of different models

However, the prediction results of the MC-LSTM network show a large discrepancy from the actual values. Therefore, it is necessary to try changing the network to improve prediction accuracy, and the MCKLNet model is proposed.

To evaluate the performance of different uncertainty modeling methods in the task of pork price prediction, five representative models are constructed: MC dropout multi-layer perceptron (MC-MLP), MC dropout gated recurrent unit (MC-GRU), MC dropout informer (MC-Informer), MC dropout temporal fusion transformer (MC-TFT), and the proposed MCKLNet. These models are selected to cover a wide range of architectures, from traditional shallow networks and classical time series models to the current mainstream transformer-based models, ensuring the representativeness and scientific rigor of the comparative experiments. The performance of each model is evaluated using several key metrics, including MAE, RMSE, MAPE, R^2 , and PICP, the comparison results are shown in Table 2.

relies solely on fully connected layers for feature extraction, while MC-GRU, despite utilizing gating mechanisms, struggles to capture the complex dynamic patterns in time series data.

The MC-Informer and MC-TFT models, as transformer-based models, excel at modeling long-term dependencies and allocating attention. As a result, they significantly outperform MC-MLP and BAYES-LSTM in metrics such as MAE and RMSE. However, their complex architectures make them more prone to sensitivity in data scale and hyperparameter tuning, limiting PICP coverage. In particular, the predicted intervals are often too narrow under extreme price fluctuations, highlighting the need for further improvement in uncertainty estimation.

The MCKLNet model incorporates modular enhancements at several key stages. The CNN layer captures local price variation patterns, the KAN mechanism improves feature representation, and the LSTM unit effectively models temporal dependencies. Meanwhile, the introduction of dropout ensures robustness and interpretability of the predictions. In the experimental results, MCKLNet achieved an MAPE as low as 0.02%, RMSE of 0.78, MAE of 0.69, and R^2 of 0.95. The PICP reached 0.44, demonstrating not only superior point prediction performance but also broader uncertainty coverage, reflecting strong generalization ability and practical applicability.

To further validate the differences in uncertainty among the various prediction models, a comparison experiment is conducted on pork price prediction using the aforementioned five models.

From Fig. 11, it can be observed that by incorporating the MC dropout uncertainty modeling technique, all models extend point predictions to interval predictions. Among these models, MC-MLP and MC-GRU exhibit the weakest predictive fidelity, with predicted results significantly deviating from the actual volatility trend and a narrower coverage of the confidence interval. This is due to the relatively simple structure of these two models, which makes it difficult for them to capture the complex dynamic patterns in the time series.

Table 2 Comparative experiment of uncertainty model

Model	MAPE/%	RMSE	MAE	PICP	R^2
MC-MLP	10.13	3.52	3.16	0.29	0.23
MC-GRU	11.07	4.01	3.49	0.25	-0.09
MC-Informer	5.96	2.10	1.87	0.30	0.70
MC-TFT	5.08	1.79	1.59	0.35	0.78
MCKLNet	0.02	0.78	0.69	0.44	0.95

The experimental results show that the MC-MLP and MC-GRU models exhibit large prediction errors, with weak PICP coverage, failing to adequately capture prediction uncertainty. This is attributed to the simpler architectures of MC-MLP and MC-GRU. MC-MLP

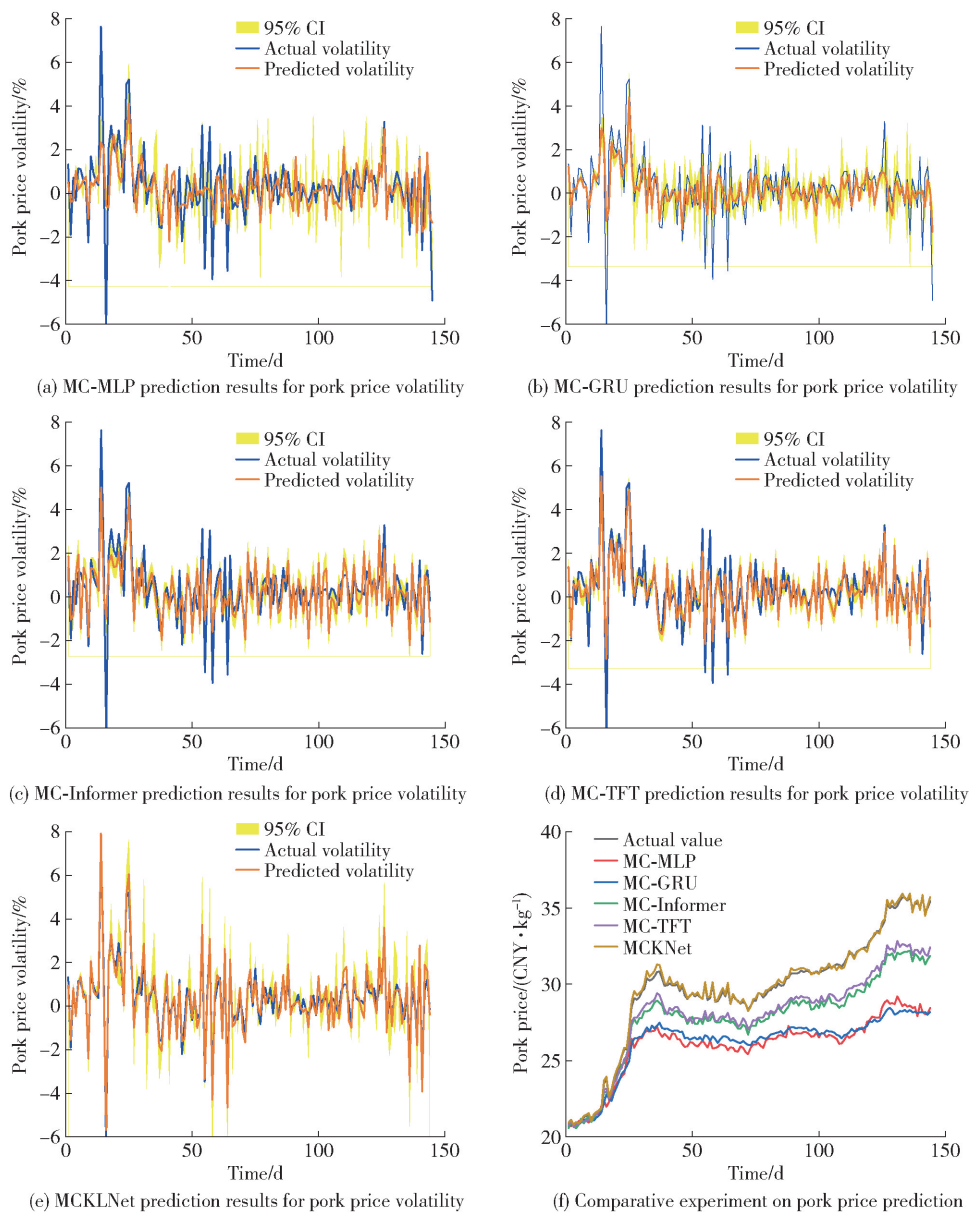


Fig. 11 Prediction results of different models

In Fig. 11, leveraging the transformer architecture, the MC-Informer and MC-TFT models possess a natural advantage in capturing long-range dependencies, generating smooth volatility predictions, and a tighter confidence interval. However, in extreme intervals with drastic price fluctuations, the PICP performance is relatively lower, suggesting that the uncertainty coverage still needs improvement. The MCKLNet model demonstrates its superiority in aligning the predicted volatility with the actual volatility trend. Its confidence intervals are reasonably distributed, with a

higher coverage range and a smoother volatility prediction. Finally, in Fig. 11(f), it further validates its overall optimal performance, as the predicted pork price curve closely matches the actual trend, demonstrating a dual advantage in both accuracy and uncertainty coverage.

4.3 Ablation experiment

The ablation experiment uses four models, LSTM, MC-LSTM, MC dropout-CNN-LSTM (MC-CNN-LSTM), and MC dropout-KNN-ISTM (MC-KAN-

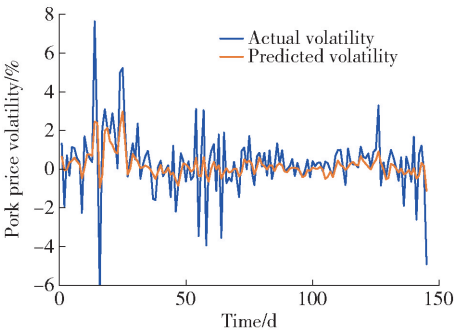
LSTM), to compare with MCKLNet. The experimental setup is consistent with the comparative experiments. The overall results of the ablation experiments are shown in Table 3. In Table 3, the MC-LSTM model without any optimization shows the worst performance.

Table 3 Ablation experiment

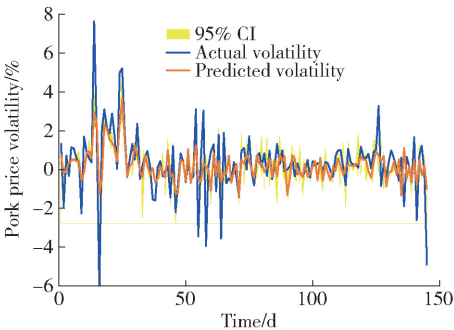
Model	MAPE%	RMSE	MAE	PICP	R^2
LSTM	13.26	4.77	4.18	–	–0.54
MC-LSTM	8.82	4.13	1.62	0.28	0.10
MC-CNN-LSTM	7.35	2.71	2.33	0.30	0.35
MC-KAN-LSTM	0.05	2.00	1.76	0.37	0.72
MCKLNet	0.02	0.78	0.69	0.44	0.95

Adding the CNN alone results in some improvement, but it is not significant. However, adding the KAN alone greatly enhances the LSTM network, resulting in significant performance improvement. Table 3 also indicates that both KAN and CNN improve the accuracy of uncertainty estimation.

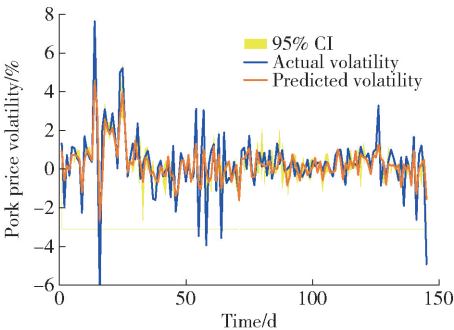
Fig. 12 illustrates the price fluctuation prediction results of LSTM, MC-LSTM, MC-CNN-LSTM, MC-KAN-LSTM, and MCKLNet, and the comparison of pork price predictions in the ablation experiment. From Fig. 12, it can be seen that, whether for price or fluctuation predictions, the predicted values of the MCKLNet model curve are closer to the actual values, indicating that MCKLNet has better accuracy.



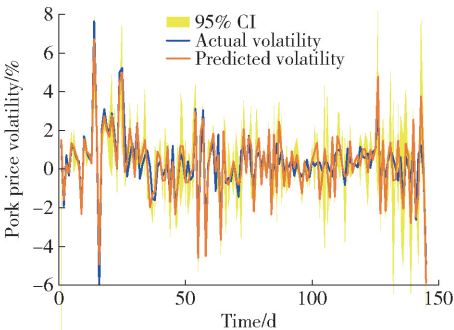
(a) LSTM prediction results for pork price volatility



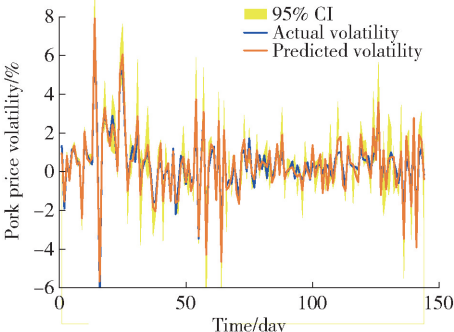
(b) MC-LSTM prediction results for pork price volatility



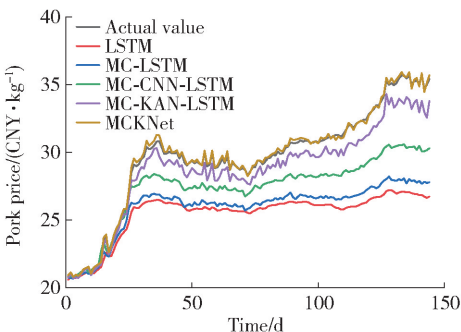
(c) MC-CNN-LSTM prediction results for pork price volatility



(d) MC-KAN-LSTM prediction results for pork price volatility



(e) MCKLNet prediction results for pork price volatility



(f) Ablation experiment on pork price prediction

Fig. 12 Ablation experiment result

4.4 Assessment of real-time prediction capability

To validate the real-time prediction capability of the MCKLNet in practical applications, a rolling forecast experiment is conducted over a ten-day period from October 1 to October 10, 2024. As empirically demonstrated in Fig. 13, the model’s multi-step predictions exhibit tight synchronization with actual market trajectories. In this experiment, the model utilized the most recent market data available each day to forecast pork prices for the following day, covering key variables such as ETC, ITC, BP, and CP, as well as PSV and CPI. A comparison with the officially published prices indicates that the model consistently achieved high accuracy and stability over multiple consecutive days, demonstrating strong timeliness and practical applicability.

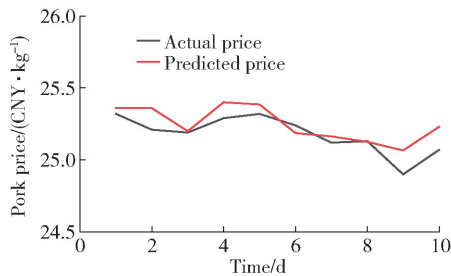


Fig. 13 Comparison of predicted and actual pork price

The input data used in this experiment are all publicly available market data from the previous day, aligning with the real-world production workflow of “data-driven-rapid prediction-decision support”. No lagging or future variables are incorporated, ensuring the causal validity of the prediction process. Furthermore, the MCKLNet demonstrates high computational efficiency, with each prediction requiring less than 0.2 s, making it suitable for daily and even higher-frequency forecasting tasks.

This continuous forecasting experiment not only demonstrates the model’s sensitivity and adaptability to short-term price fluctuations during a specific period but also confirms that the MCKLNet is capable of delivering near real-time price forecasts in practical

agricultural market analysis. In future work, integrating higher-frequency data sources may further enhance the model’s responsiveness and predictive accuracy, thereby expanding its potential for deployment in intelligent agricultural decision-support systems.

5 Conclusions

In this paper, an innovative uncertainty-aware time series forecasting model MCKLNet was proposed. MCKLNet, integrates CNN, KAN, LSTM, and MC dropout into a unified framework to capture local features, enhance nonlinear mapping, model temporal dependencies, and estimate prediction uncertainty. Experimental results show that MCKLNet outperforms multiple baseline models in both prediction accuracy and uncertainty quantification. Ablation studies verify the effectiveness of the KAN, CNN, and MC dropout modules. Tested on a typical high-volatility time series dataset, the model achieves accurate and well-calibrated forecasts. Benefiting from its modular design, MCKLNet can be readily extended to various time series forecasting tasks. Future work will focus on improving computational efficiency and cross-domain generalization ability.

Acknowledgements

This work was supported by the Scientific and Technological Innovation 2030-Major Project (2021ZD0113603), and the Young Backbone Teacher Support Plan of Beijing Information Science Technology University (YBT202414).

References

[1] BAI S J, ZICO KOLTER J, KOLTUN V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv Preprint, arXiv:1803.01271, 2018.

[2] DAI Z H, YANG Z L, YANG Y M, et al. Transformer-XL: attentive language models beyond a fixed-length context. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, Jul 28 – Aug 2, Florence, Italy. Stroudsburg, PA, USA: Association for Computational

- Linguistics, 2019: 2978 – 2988.
- [3] KENDALL A, GAL Y. What uncertainties do we need in Bayesian deep learning for computer vision? Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS'17), 2017, Dec 4 – 9, Long Beach, CA, USA. Red Hook, NY, USA: Curran Associates Inc, 2017: 5575 – 5585.
 - [4] HOCHREITER S, SCHMIDHUBER J. Long short-term memory. Neural Computation, 1997, 9(8): 1735 – 1780.
 - [5] CHO K, VAN MERRIËNBOER B, GULCEHRE C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14), 2014, Oct 25 – 29, Doha, Qatar. Stroudsburg, PA, USA: Association for Computational Linguistics, 2014: 1724 – 1734.
 - [6] ZHU D, ZHAO Q Y, CHOI H, et al. A scalable multi-photon coincidence detector based on superconducting nanowires. Nature Nanotechnology, 2018, 13(7): 596 – 601.
 - [7] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), 2017, Dec 4 – 9, Long Beach, CA, USA. Red Hook, NY, USA: Curran Associates Inc, 2017, 6000 – 6010.
 - [8] ZHOU H Y, ZHANG S H, PENG J Q, et al. Informer: beyond efficient transformer for long sequence time-series forecasting. Proceedings of the 35th AAAI Conference on Artificial Intelligence, the 33rd Conference on Innovative Applications of Artificial Intelligence, the 11th Symposium on Educational Advances in Artificial Intelligence (AAAI'21/IAAI'21/EAAI'21), 2021, Feb 2 – 9, held virtually. Technical Tracks 12. Palo Alto, CA, USA. AAAI Press, 2021: 11106 – 11115.
 - [9] GAL Y, GHAHRAMANI Z. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. Proceedings of the 33rd International Conference on Machine Learning (CML'16), 2016, Jun 19 – 24, New York, NY, USA. Journal of Machine Learning Research (JMLR), 2016: 1050 – 1059.
 - [10] LAKSHMINARAYANAN B, PRITZEL A, BLUNDELL C. Simple and scalable predictive uncertainty estimation using deep ensembles. Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), 2017, Dec 4 – 9, Long Beach, CA, USA. Red Hook, NY, USA: Curran Associates Inc, 2017: 6405 – 6416.
 - [11] FORTUNATO M, BLUNDELL C, VINYALS O. Bayesian recurrent neural networks. arXiv Preprint, arXiv:1704.02798, 2019.
 - [12] ZHOU H J, ZHOU F T, ZHAO C Y, et al. Multimodal data integration for precision oncology: challenges and future directions. arXiv Preprint, arXiv:2406.19611, 2024.
 - [13] WU Z H, PAN S R, CHEN F W, et al. A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 2020, 32(1): 4 – 24.
 - [14] SUN F H, MENG X Y, ZHANG Y, et al. Agricultural product price forecasting methods: a review. Agriculture, 2023, 13(9): Article 1671.
 - [15] WANG Y, WANG Z, WANG X Y, et al. Multi-step-ahead and interval carbon price forecasting using transformer-based hybrid model. Environmental Science and Pollution Research, 2023, 30(42): 95692 – 95719.
 - [16] YANG X D, TANG G Q. Deep study on marketing decision under the price fluctuation of pig market based on Bayesian method. Proceedings of the 16th Dahe Fortune China Forum and Chinese High-educational Management Annual Academic Conference (DFHMC'20), 2020, Dec 4 – 6, Zhengzhou, China. Piscataway, NJ, USA: IEEE, 2020: 73 – 76.
 - [17] CHO K, VAN MERRIËNBOER B, BAHDANAU D, et al. On the properties of neural machine translation: encoder-decoder approaches. Proceedings of the 8th Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST'14), 2014, Oct 25, Doha, Qatar. Stroudsburg, PA, USA: Association for Computational Linguistics, 2014: 103 – 111.
 - [18] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks. arXiv Preprint, arXiv:1409.3215, 2014.
 - [19] ZAREMBA W, SUTSKEVER I, VINYALS O. Recurrent neural network regularization. arXiv Preprint, arXiv:1409.2329, 2015.
 - [20] LECUN Y, BOSER B, DENKER J S, et al. Backpropagation applied to handwritten zip code recognition. Neural Computation, 1989, 1(4): 541 – 551.
 - [21] LIU B, ZHANG Y, HE D J, et al. Identification of apple leaf diseases based on deep convolutional neural networks. Symmetry, 2018, 10(1): Article 11.
 - [22] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks. Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'12), 2012, Dec 3 – 6, Lake Tahoe, NV, USA. Red Hook, NY, USA: Curran Associates Inc, 2012: 1097 – 1105.
 - [23] BOUREAU Y L, PONCE J, LECUN Y. A theoretical analysis of feature pooling in visual recognition. Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML'10), 2010, Jun 21 – 24, Haifa Israel. Madison, WI, USA: Omnipress, 2010: 111 – 118.
 - [24] ZEILER M D, FERGUS R. Visualizing and understanding convolutional networks. Proceedings of the 13th European Conference on Computer Vision (ECCV'14): Part I, 2014, Sep 6 – 12, Zurich, Switzerland. LNCS 8689. Berlin, Germany: Springer, 2014: 818 – 833.
 - [25] HECHT-NIELSEN R. Kolmogorov's mapping neural network existence theorem. Proceedings of the IEEE 1st Annual International Conference on Neural Networks, 1987, Jun 21 – 24, San Diego, CA, USA. Piscataway, NJ, USA: IEEE, 1987: 11 – 14.
 - [26] HORNIK K, STINCHCOMBE M, WHITE H. Multilayer feedforward networks are universal approximators. Neural Networks, 1989, 2(5): 359 – 366.
 - [27] DAS S, CHIANG Y K, MATHUR S. Thermal Sunyaev-Zel'dovich effect in the circumgalactic medium. I. detection and a surprising pattern in self-similarity and baryon sufficiency. The Astrophysical

- Journal, 2023, 951(2): Article 125.
- [28] XU K P, CHEN L F, WANG S R. Kolmogorov-Arnold networks for time series: bridging predictive power and interpretability. arXiv Preprint, arXiv:2406.02496, 2024.
- [29] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014, 15(1): 1929 – 1958.
- [30] HINTON G E, SRIVASTAVA N, KRIZHEVSKY A, et al. Improving neural networks by preventing co-adaptation of feature detectors. arXiv Preprint, arXiv:1207.0580, 2012.
- [31] DAMIANOU A C, LAWRENCE N D. Deep Gaussian processes. *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS'13)*, 2013, Apr 29 – May 1, Scottsdale, AZ, USA. *Journal of Machine Learning Research (JMLR)*, 2013: 207 – 215.
- [32] HENNIGHAUSEN L G, SIPPEL A E. Characterization and cloning of mRNAs specific for mouse mammary gland during lactation. *European Journal of Biochemistry*, 1982, 125(1): 131 – 141.
- [33] GHYSELS E, SINKO A, VALKANOV R. MIDAS regressions: further results and new directions. *Econometric Reviews*, 2007, 26(1): 53 – 90.
- [34] MARCELLINO M, SCHUMACHER C. Factor-MIDAS for now-and forecasting with ragged-edge data: a model comparison for German GDP. *Discussion Paper, Series 1: Economic Studies*, No. 34/2007.
- [35] SAX C, STEINER P. Temporal disaggregation of time series. *The R Journal*, 2013, 5(2): 80 – 87.

(Editor: Wang Xuying)