Journal of Organizational and End User Computing
Volume 36 • Issue 1

Intelligent Productivity Transformation:
Corporate Market Demand Forecasting With the Aid of an AI Virtual Assistant

Bojing Liu, Institute of Medical Humanities, Wenzhou Medical University, China
Mengxiang Li, Hebi Vocational College of Automotive Engineering, China
Zihui Ji, Moray House School of Education and Sport, Old College, University of Edinburgh, UK
Hongming Li, University of Florida, USA
https://orcid.org/0000-0002-6024-677X
Ji Luo, School of Economics and Management, Shanghai Polytechnic University, China*

ABSTRACT

With the penetration of deep learning technology into forecasting and decision support systems, enterprises have an increasingly urgent need for accurate forecasting of time series data. Especially in fields such as finance, retail, and production, immediate and accurate predictions of market trends are the key to maintaining a competitive advantage. This study aims to address the limitations of traditional time series forecasting methods, such as the difficulty in adapting to the nonlinearity and non-stationarity of the data, through an innovative deep learning framework. The authors propose a Prophet model that combines deep learning with LSTNet and statistics. In this way, they combine the ability of LSTNet to handle complex time dependencies and the flexibility of the Prophet model to handle trends and periodicity. The particle swarm optimization algorithm (PSO) is responsible for tuning this hybrid model, aiming to improve the accuracy of predictions. Such a strategy not only helps capture long-term dependencies in time series, but also models seasonality and holiday effects well.

KEYWORDS

AI Virtual Assistants, Business Intelligence, Data Analysis, Deep Learning, Enterprise Collaboration, Forecasting Tools

INTRODUCTION

In our highly globalized and competitive business landscape, the survival and advancement of companies depend not only on the accurate prediction of market demand but also on their ability to respond swiftly to market fluctuations, particularly in an era of rapid technological advancement in artificial intelligence (AI) and big data (Angelopoulos et al., 2019). AI virtual assistants are increasingly becoming a vital force for enhancing internal collaboration and boosting productivity within enterprises.

DOI: 10.4018/JOEUC.336284
*Corresponding Author

Although the synergy of deep learning and business intelligence has opened new avenues for forecasting market demand, practical integration of these technologies into corporate collaboration environments presents notable challenges (Trakadas et al., 2020). Firms encounter specific obstacles when applying deep learning in real-world business processes, especially when it comes to handling seasonal and trend patterns, optimizing algorithms, and selecting hyperparameters—issues that warrant thorough research (H. Zhang et al., 2023).

To address these research gaps, this article introduces a novel research topic focused on delving into the best practices for deploying AI virtual assistants in internal corporate collaboration environments, and their impact on enhancing production efficiency. Specifically, our attention centers on constructing intelligent virtual assistants employing a blend of the LSTNet-Prophet model. By exploring the proficiency of long short-term memory (LSTM) technology in capturing the long-term dependencies of time series data and the adeptness of the Prophet model in managing trend and seasonality, we aim to cultivate an AI virtual assistant that can foresee market demand with impressive accuracy. Consequently, this should aid businesses in making more informed and nimble decisions and consultations.

Through this innovative approach, we intend to validate the practicality of intelligent assistants in authentic business contexts and unearth their potential role in fostering intra-enterprise collaboration and amplifying productivity. By presenting the core issues, the challenges encountered in real application scenarios, and the opportunities they present, we hope to offer readers clear insight into the direction and contributions of our research. We appreciate your guidance and will ensure that future submissions thoroughly consider and meet the expectations of reviewers, achieving greater comprehensibility and depth in our work.

In the literature exploring the fields of market demand forecasting and enterprise efficiency improvement, we note that existing research mainly tends to adopt classic forecasting methods. Especially when dealing with individual differences and dynamic environmental changes, researchers often combine traditional statistical models with advanced machine learning techniques (Min et al., 2019). These methods include artificial neural networks (ANN), convolutional neural networks (CNN), recurrent neural networks (RNN) and long short-term memory (LSTM). These methods are favored for their ability to identify and automatically extract complex patterns in market trends, and they have made significant contributions to improving the quality of corporate decision-making and prediction accuracy.

RNN is a powerful sequence data processing model whose basic principle is to capture temporal dependencies in data using recurrent connections between neurons (Pagliari et al., 2020). RNN has made breakthrough progress in the fields of speech recognition and natural language processing (X. Gao et al., 2021). However, traditional RNN is prone to gradient disappearance or gradient explosion problems when processing long sequences, which limits its application in long-term sequence prediction, such as market demand forecasting (Liang et al., 2022). Nonetheless, RNN has gradually shown its potential in areas such as financial market analysis and sales trend prediction through variants and improvements, such as gated recurrent units (GRU).

LSTM, as an extension of RNN, effectively solves gradient-related problems by introducing a “gate” structure (Yang et al., 2022). LSTM can store information over longer time intervals and sequences, which makes it outstanding in tasks such as text generation and speech recognition that require understanding long-term dependencies (Zheng et al., 2021). In the field of internal market forecasting of enterprises, the LSTM model has been used to predict sales trends, inventory requirements, etc. Its advantage is that it can model complex patterns in data sets, thereby improving the accuracy of predictions (W. Wang et al., 2023).

GRU is a sequence model similar to LSTM, but is more concise in structure because it uses fewer parameters and a simpler gate structure (Yuan et al., 2022). This feature makes GRU computing more efficient, especially suitable for scenarios where the data set is relatively small or computing resources are limited (Lu et al., 2019). GRU shows similar performance to LSTM, when processing time series data, and in some cases even outperforms LSTM (Wang et al., 2022). In enterprise applications, GRU is used to predict short-term market changes, especially in resource-constrained small and medium-sized enterprises.

The Transformer model has recently caused a revolutionary impact in many fields, especially because of its outstanding performance in the field of natural language processing (Shen et al., 2023). The Transformer model uses the self-attention mechanism to allow the model to process sequence data in parallel while capturing long-distance dependencies, significantly improving computing efficiency and performance (G. Gao et al., 2023). In the field of market demand forecasting, Transformer shows great potential, especially when dealing with complex and large-scale time series data (J. Zhang et al., 2022). Its ability to understand and model deep patterns in time series data is critical for accurately predicting market trends.

The above four classic models have improved the accuracy and efficiency of market demand forecasting during their development process. They have been widely used in research on market demand forecasting and enterprise efficiency improvement, providing enterprises with valuable insights. However, they still show shortcomings. The black-box characteristics of ANN when dealing with nonlinear problems, the limitations of CNN in extracting time series data features, the insufficient memory ability of RNN on long sequences, and the challenges of parameter adjustment and computational cost of LSTM all limit the practical applications of the models. Application efficiency and prediction accuracy under the circumstances. Based on the shortcomings of the above methods, this paper proposes a particle swarm optimization (PSO-optimized) LSTNet-Prophet model. Our model consists of two main components: The LSTNet part is designed to capture and analyze the long-term dependency structure of time series data, while the integrated Prophet module is responsible for capturing the trends and seasonal patterns of time series. The PSO algorithm plays the role of an optimizer in this hybrid model, fine-tuning model parameters to maximize predictive performance.

This model provides a multi-level, composite forecasting framework that fully combines LSTNet's complex model processing capabilities with Prophet's sensitive grasp of nonlinear trends and seasonal factors. The integration of PSO further improves the adaptability of the model, enabling it to maintain accurate predictions in a changing market environment. Compared with traditional models, our method provides improvements in forecast accuracy and calculation efficiency, which means that, in actual commercial applications, companies can more effectively predict future demand to optimize inventory management and improve operational efficiency, thus gaining a competitive advantage.

The main contributions of this study are as follows:

1. An innovative LSTNet-Prophet model was constructed, which combines the advantages of deep learning and machine learning algorithms, especially showing excellent performance when dealing with nonlinear time series prediction problems.
2. The PSO strategy was introduced to optimize the hyperparameters of the LSTNet-Prophet model.
3. Through experimental verification, it was proved that the LSTNet-Prophet model performs better than existing prediction technologies on market demand and enterprise efficiency data sets.

In the following article structure, we will organize the content as follows: In the Related Works section, we elaborate on related works. The Methodology section will introduce the key details of our proposed LSTNet-Prophet model and PSO optimization strategy in detail. The Experiments section will focus on our experimental design and experimental results. Finally, the Conclusion and Discussion section will be the summary and discussion of this study.

RELATED WORKS

Market Demand Forecast

Market demand forecasting is an analytical method used to predict the demand for specific products or services in a specific market within a specific time period in the future. In this research area, the accuracy of forecasts is directly related to the enterprise's inventory management, production planning, sales strategy, and revenue maximization. As the business environment becomes increasingly complex, traditional statistical forecasting methods are gradually unable to meet the needs of enterprises for accurate forecasts (Nguyen et al., 2021). The introduction of machine learning and deep learning technologies has opened a new stage of market demand forecasting.

Current research focuses on how to improve the accuracy and robustness of predictive models. To this end, researchers continue to try to combine time series analysis, causality models, and machine learning algorithms to deal with the nonlinear characteristics and potential dynamic changes of data. In addition, with the development of data collection technology, how to efficiently process and utilize massive data in a big data environment has become a research hotspot (Collaboration, 2020).

However, market demand forecasting still faces some difficulties in practical applications. For example, market data are often affected by changing external factors, which makes it difficult for models to capture all relevant factors that affect predictions (Dogan & Birant, 2021). In addition, the balance between model complexity and computational cost is also a challenge, especially in scenarios where real-time or near-real-time predictions are required.

AI Virtual Assistant

AI virtual assistant uses artificial intelligence technology to imitate the behavior of human assistants to provide users with help or services. In this research field, the development of virtual assistants has penetrated into many aspects such as customer service, personal life management, and even enterprise automated decision-making (Chowdhury et al., 2022). They are often based on natural language processing, machine learning, and deep learning technologies and are able to understand and respond to user commands.

The focus of research is mainly on improving the naturalness of interaction, context understanding ability, and intelligent task execution of AI virtual assistants (Wu & Horng, 2022). Taking dialogue systems as an example, the trend is towards multi-round dialogue, emotion recognition, and personalized recommendations. This not only enhances the user experience but also makes the assistant more proficient in serving users in complex tasks.

Current challenges include the limited intelligence of virtual assistants, which still struggle to fully understand complex context and unstructured human language (Bai et al., 2022). In addition, privacy and security issues have become increasingly prominent, especially when assistants need to handle sensitive information and perform personal data analysis.

Deep Learning Technology

Deep learning technology is an algorithm in machine learning that imitates the neural network structure and function of the human brain and is used to identify complex patterns and features in data. In many fields, especially visual recognition, speech recognition, and natural language processing, deep learning technology has demonstrated extraordinary capabilities (Belhadi et al., 2021). Its development provides powerful tools for processing and analyzing large-scale data. Especially in this field, deep learning technology has become a key force in promoting innovation.

The trend of research is shifting from pure model performance optimization to model interpretability and reliability. More and more research is devoted to making the decision-making process of deep learning models more transparent, and new network architectures are also being explored to improve the model's generalization ability on unseen data (Allal-Chérif et al., 2021).

Although deep learning technology has made significant progress, model training is still costly and requires large amounts of data and computing resources, which limits its application in resource-constrained environments (Wu, 2022). In addition, the "black box" characteristics of deep learning models have also caused widespread discussion, and the interpretability of models has become equally important in practical applications.

METHODOLOGY

Overview of Our Model

This research aims to develop a highly accurate market demand forecasting system by combining deep learning and machine learning methods. We designed and implemented an advanced AI virtual assistant called LSTNet-Prophet to accurately predict and analyze market demand, and also used a particle swarm optimization (PSO) algorithm to improve the performance of the forecast model. LSTNet-Prophet is the core part of our research, which comprehensively utilizes the powerful capabilities of long short-term memory networks (LSTM) and the flexibility of the Prophet method. LSTM is a special type of recurrent neural network capable of capturing long-term dependencies in time series data through its internal self-adjusting "gate" structure. This is critical to understanding and predicting enterprise market needs with complex patterns. At the same time, we integrated the Prophet forecast model, which is particularly good at modeling trends and cyclical factors. The addition of Prophet enhances the model's ability to understand irregular changes in time series (for example, holiday effects). Through this integration, the LSTNet-Prophet module can effectively handle complex market demand changes and capture key signals within time series. PSO is an optimization algorithm based on swarm intelligence, inspired by the predatory behavior of bird flocks. This algorithm optimizes hyperparameters by simulating the process of particles in the population moving in the solution space to find the optimal solution. In this study, PSO is used to find the optimal network weights and hyperparameter configurations, which are crucial to improve the performance of the LSTNet-Prophet model. Each particle in the optimization process represents a potential solution. They learn from each other and adjust their flight direction during the iterative process, gradually approaching the optimal hyperparameter combination.

The overall model building process is as follows: First, we thoroughly preprocess the market demand data to ensure data quality and provide a solid foundation for subsequent analysis. The preprocessed data is then input into the LSTNet-Prophet module for training, and the model learns and extracts key features of the time series. At this time, the initial prediction results output by LSTNet-Prophet will be further refined in the PSO optimization module. The PSO algorithm starts a search and continuously updates the position of each particle until it finds the most suitable parameter settings for the LSTM and Prophet parts.

Through the organic combination of the LSTNet-Prophet model and the PSO algorithm, this model not only learns the complex dynamics of market demand but also has the ability to quickly adjust to adapt to new trends. This intelligent system provides an automated, highly accurate demand forecasting tool, providing a scientific basis for companies to quickly respond and adjust strategies in a changing market. We expect that through such smart productivity transformation, companies will be able to significantly improve operational efficiency and market adaptability.

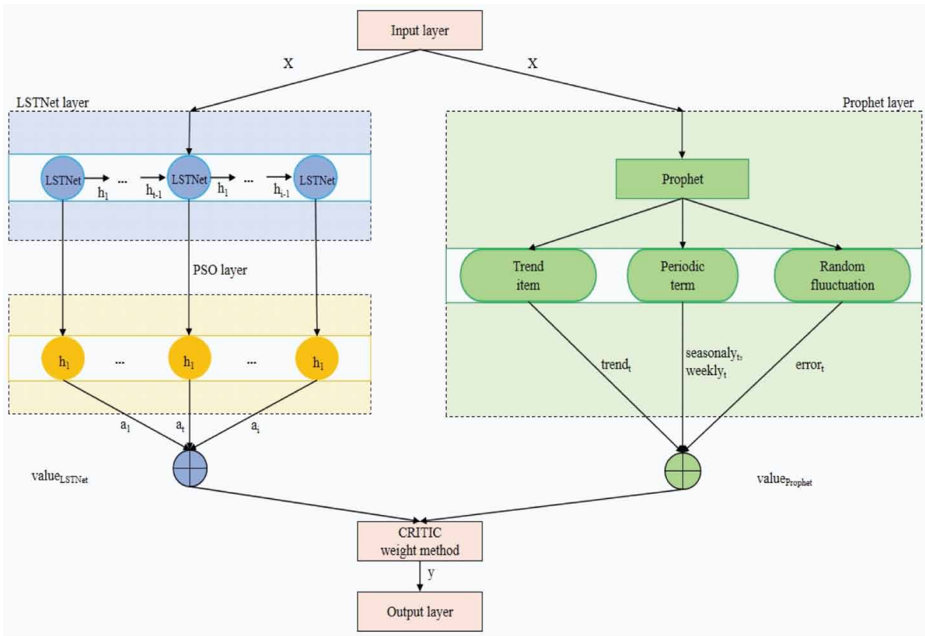
The overall model structure diagram is shown in Figure 1.

The running process of the model is shown in Algorithm 1.

LSTNet Model

LSTNet is a hybrid time series prediction model that combines convolutional neural networks (CNN) and recurrent neural networks (RNN) and is dedicated to capturing short-term and long-term dependencies in time series data. The CNN layer specializes in spatial feature extraction of input

Figure 1. Overall flow chart of the model



data and can effectively identify short-term local trends and periodic patterns, while the RNN layer is used to handle the time dependence of time series, especially the dependence in the long term (Zhu et al., 2022). This combination helps the model take into account both the local behavior and the long-term trend of the time series.

In market demand forecasting, LSTNet has demonstrated the ability to model complex dynamics of time series and has unique advantages in processing continuous data streams and solving real-time forecasting problems. It can make full use of time series information such as historical sales data to provide retailers and supply chain managers with accurate demand forecasts (Wu et al., 2023). The power of LSTNet is also reflected in its processing of multi-variable time series, which can simultaneously consider the impact of multiple related factors, such as marketing activities, price changes, etc., providing a more comprehensive perspective.

In the entire LSTNet-Prophet architecture, LSTNet is a key technical pillar. Its CNN part is responsible for quickly identifying important patterns and outliers in data that are critical for short-term decision-making. The RNN part supplements the identification of long-term dependencies and provides support for the long-term prediction of the model. This ability to capture long-term and short-term dynamics and deeply analyze patterns is an indispensable part of LSTNet in the integrated prediction framework. Its combination with the Prophet model allows the forecasting system to not only capture conventional time series dynamics but also adapt to nonlinear and irregular market changes. The integration of LSTNet significantly enhances the performance of the overall forecasting framework and provides a deeper insight and analysis basis for market demand forecasting.

The structure diagram of the LSTNet model is shown in Figure 2.

The main formula of LSTNet is as follows:

$$X_c = tanh\left(W_C * X + b_c\right) \tag{1}$$

Algorithm 1. Training LSTNet-Prophet model with PSO optimization

Input: Online Retail DataSet, Supply Chain Dataset, Bureau of Labor Statistics Data, Awesome

Output: Trained LSTNet-Prophet model with optimized hyperparameters

Variables: $epochs$, $batch\_size$, lr, mae, $rmse$

Hyperparameters: $c1$, $c2$, $num\_particles$, max_itern

itialization:

for $i \leftarrow 1$ **to** $num\_particles$ **do**

  Initialize particle position x_i and velocity v_i

  Evaluate particle by training LSTNet-Prophet and compute $Fitness(x_1)$ based on MAE

  Set personal best $pbest_i \leftarrow x_i$

end

$$gbest \leftarrow arg\,min_i\left(Fitness\left(pbest_i\right)\right)$$

while not converged and $iter < mar\_iter$ **do**

  for $i \leftarrow 1$ **to** $num\_particles$ **do**

    Update v_1 using $c1$, $c2$, $gbest$ and $pbest_i$

    Update position x_i with v_i

    Evaluate new position and update $Fitness(x_1)$ based on MAE

    if $Fitness(x_i) < Fitness(pbest_i)$ **then**

      $pbest_i \leftarrow x_i$

      if $Fitness(x_i) < Fitness\left(gbest\right)$ **then**

        $gbest \leftarrow x_i$

      end

    end

  end

  $iter \leftarrow iter + 1$

end

for $e \leftarrow 1$ **to** $epochs$ **do**

  for each $batch$ in dataset **do**

    Train LSTNet-Prophet on $batch$ using $gbest$ as hyperparameters

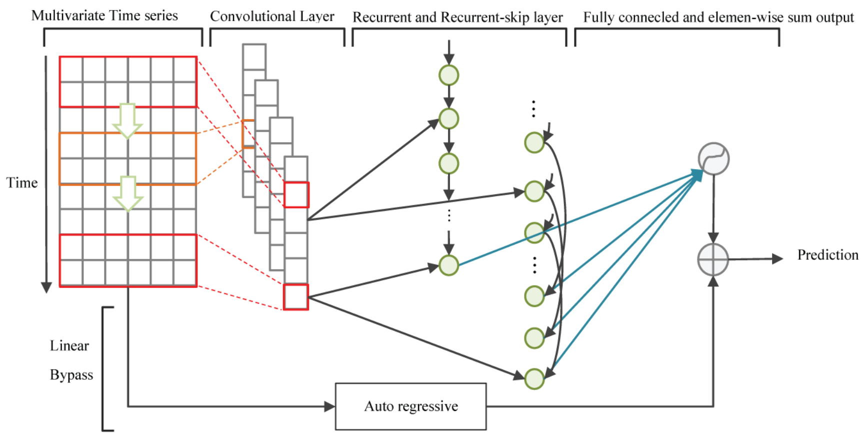
    Compute mac, $rmse$ for $batch$

  end

  Evaluate model on validation set and compute mae, $mape$, $rmse$, mse

end

Figure 2. Flow chart of the BERT model



where X_c is the convolution output, W_C represents the weights of the convolutional layer, $*$ denotes the convolution operator, X is the input time-series matrix, and b_c is the bias for the convolutional layer.

$$H = ReLU\left(W_r \cdot X_c + b_r\right) \tag{2}$$

where H is the hidden representation after a recurrent layer, $ReLU$ is the rectified linear unit activation function, W_r is the weight matrix of the recurrent layer, and b_r is the recurrent layer bias.

$$S = softmax\left(W_S \cdot H + b_S\right) \tag{3}$$

where S is the output of the softmax function, W_S is the weight matrix of the softmax layer, and b_S is the bias of the softmax layer.

$$\hat{Y} = H\left[-1\right] \tag{4}$$

where $\hat{Y}$ is the predicted output and $H\left[-1\right]$ is the last hidden state of the recurrent layer, capturing the temporal dependencies.

$$L_{reg} = \sum_i W_{iF}^2 \tag{5}$$

where L_{reg} is the regularization loss to prevent overfitting, W_i represents the weight matrices of the model, and $\left|\cdot\right|_F$ is the Frobenius norm.

$$L = \frac{1}{N}\sum_{n=1}^{N}\left(\hat{Y}_n - Y_n\right)^2 + \lambda L_{reg} \tag{6}$$

where L is the overall loss function, N is the number of samples, $\hat{Y}_n$ is the predicted output for the n -th sample, Y_n is the true output for the n -th sample, and λ is the regularization parameter.

$$W_i \leftarrow W_i - \eta \frac{\partial L}{\partial W_i} \tag{7}$$

where W_i are the parameters updated during training, η is the learning rate, and $\frac{\partial L}{\partial W_i}$ represents the gradient of the loss L with respect to the parameters.

Prophet Model

Prophet is an open source time series forecasting tool developed by Facebook. The main principle is to decompose time series into several components that are easy to model. It typically models time series data as an additive model that incorporates components such as trend, seasonality, and holidays (Grek, 2020). The trend component is used to simulate the long-term trend of the time series; the seasonal component is used to capture cyclical changes, such as patterns within a week or a year, and the holiday component can adjust the impact of abnormal dates (Zhou & Zhang, 2023). Prophet is particularly suitable for calendar-related data and is highly robust to missing values and trend changes. Its main purpose is to provide interpretable and accurate forecast results for business analysis and forecasting and is ideal for processing time series data with strong seasonal and historical trends, such as sales and website traffic.

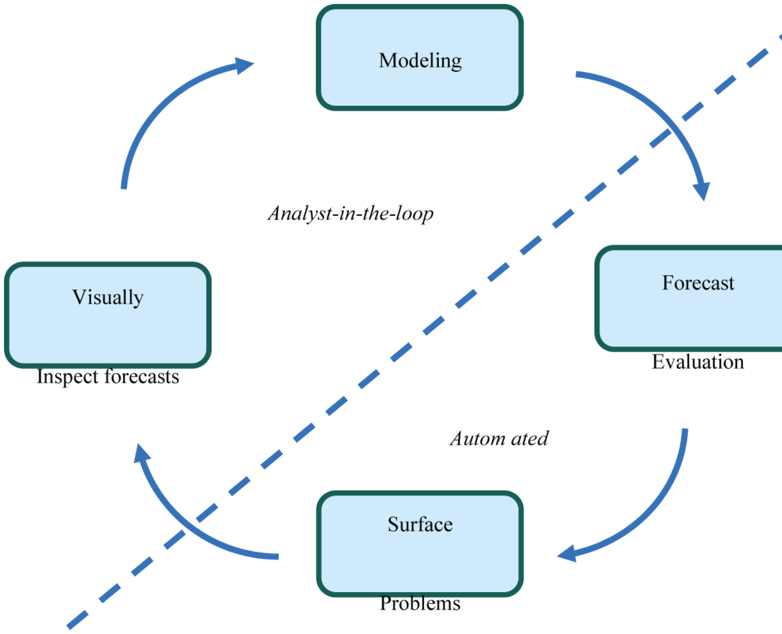
In business areas such as market demand forecasting, the Prophet model is a popular tool praised for its ease of use and flexibility. The advantage of Prophet is that it is beginner-friendly, simplifies the adjustment process of forecast models, and can provide intuitive trend component decomposition results, which is particularly helpful for business analysts to understand and communicate the results (Li et al., 2023). Furthermore, Prophet's ability to model holiday effects makes it very practical in industries such as retail and e-commerce where demand forecasts are easily affected by holidays, making forecasts not only limited to regular patterns but also providing accurate forecasts during special periods.

In the entire LSTNet-Prophet architecture, the Prophet module plays an indispensable role. It provides a seasonal context for non-linear trends in the forecast results of the long short-term memory network (LSTNet) model. By combining the Prophet model with LSTNet, our system is able to leverage the deep features captured by the LSTNet module and place them within Prophet's structured trend analysis framework. The combination of the two significantly optimizes the accuracy and stability of predictions. In addition, the Prophet model's adaptability to emergencies and unconventional changes also strengthens the generalization ability of the overall model, enabling it to adapt to complex and ever-changing market environments. By identifying and adjusting special dates (such as promotions, holidays) that may affect market demand, the LSTNet-Prophet model can be called a powerful forecasting tool that comprehensively integrates market dynamics, providing real-time and accurate information required for business decisions.

The structure diagram of the Prophet model is shown in Figure 3.

The main formula of Prophet is as follows:

Figure 3. Flow chart of the Prophet model



$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \tag{8}$$

where $y(t)$ is the predicted value at time t, $g(t)$ represents the trend component, $s(t)$ is the seasonality component, $h(t)$ accounts for the effects of holidays, and ε_t is the error term.

$$g(t) = \left(k + a(t)^T \delta\right)t + \left(m + a(t)^T \gamma\right) \tag{9}$$

where $g(t)$ describes the trend using piecewise linear or logistic growth, with k as the growth rate, $a(t)$ as the adjustment vector for changepoints, δ representing the rate adjustment at changepoints, m the offset parameter, and γ the parameter vector controlling the change in the offset at changepoints.

$$s(t) = \sum_{n=1}^{N} (a_n cos\left(\frac{2\pi nt}{P}\right) + b_n sin\left(\frac{2\pi nt}{P}\right) \tag{10}$$

where $s(t)$ captures seasonality with Fourier series, a_n and b_n are coefficients for the seasonality model, P represents the period (e.g., yearly, weekly), and n indexes the Fourier terms.

$$h(t) = \sum_{i=1}^{L} K_i I\left(t \in D_i\right) \tag{11}$$

where $h\left(t\right)$ represents the holiday effect, K_i are the effects associated with the i-th holiday, $I\left(\cdot\right)$ is the indicator function which equals 1 if t belongs to the set of days D_i associated with the i-th holiday and 0 otherwise, and L is the total number of holidays.

$$a\left(t\right) = \left\{1\left(t > t_1\right), 1\left(t > t_2\right), \cdots, 1\left(t > t_{S-1}\right)^T\right\} \tag{12}$$

where $a\left(t\right)$ is an adjustment vector whose components are 0 or 1 indicating whether the time t is past the corresponding changepoint t_j, and S is the number of total change points.

$$\delta = \left(\delta_1, \delta_2, \cdots, \delta_{S-1}\right)^T \tag{13}$$

where δ is the rate adjustment vector, with δ_j representing the rate change at each change point.

$$\gamma = -s \cdot \delta \tag{14}$$

where γ controls the adjustments to the offset at change points, and s is the vector with components representing the sign of each rate change.

PSO Model

The PSO model is a swarm intelligence algorithm. The basic principle is derived from a simplified model of the foraging behavior of bird flocks. In PSO, each solution can be regarded as a "particle" in the search space, and all particles together form a "group." Each particle has its own position (representing a candidate solution) and speed (determining the search direction and step size) and searches toward the best position for individual experience and the best position for the entire group (Shahaab et al., 2023). By iteratively updating their speed and position, the particles try to find the global optimal solution to the optimization problem. PSO is widely used in function optimization, neural network training, control system design, and various optimization problems. It is favored for its simple implementation and fast convergence speed.

In fields such as market demand forecasting, the PSO model is used to optimize the parameter selection of the forecast model, such as the weights and biases of the neural network, to improve the accuracy of the forecast. The application of PSO in this field demonstrates its rapid optimization capabilities and adaptability. It can overcome the shortcomings of traditional optimization methods that easily fall into local optimality and have high computational complexity (Wang, 2022). Compared with other global optimization algorithms, PSO relies less on specific knowledge of the problem and does not need to make assumptions about the gradient information of the optimization problem, making it more flexible and efficient when solving actual optimization problems. In addition, the PSO algorithm has few parameters and is easy to adjust, which can effectively save time in model development and tuning. It is especially suitable for market environments with high dynamics and uncertainty.

In the overall LSTNet-Prophet model, PSO plays the role of key parameter optimizer. Through PSO, we can optimize multiple key parameters of the LSTNet-Prophet model, such as the learning rate of the neural network, the number of hidden layer units, and even the holiday component parameters of the Prophet model. The introduction of PSO enables the entire framework to find the best parameter combination globally, thereby matching the characteristics of time series data and improving prediction accuracy. This parameter optimization process not only enhances the model's ability to fit historical data but also improves its accuracy in predicting future trends. In short, as an optimization engine, the use of PSO is crucial to improving the comprehensive performance of the entire time series forecast model, ensuring that the model can effectively respond to complex and changing market demands and provide more accurate decision support.

The structure diagram of the PSO model is shown in Figure 4.

The main formula of PSO is as follows:

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \tag{15}$$

where $x_i^{(t+1)}$ is the updated position of particle i at iteration $(t+1)$:

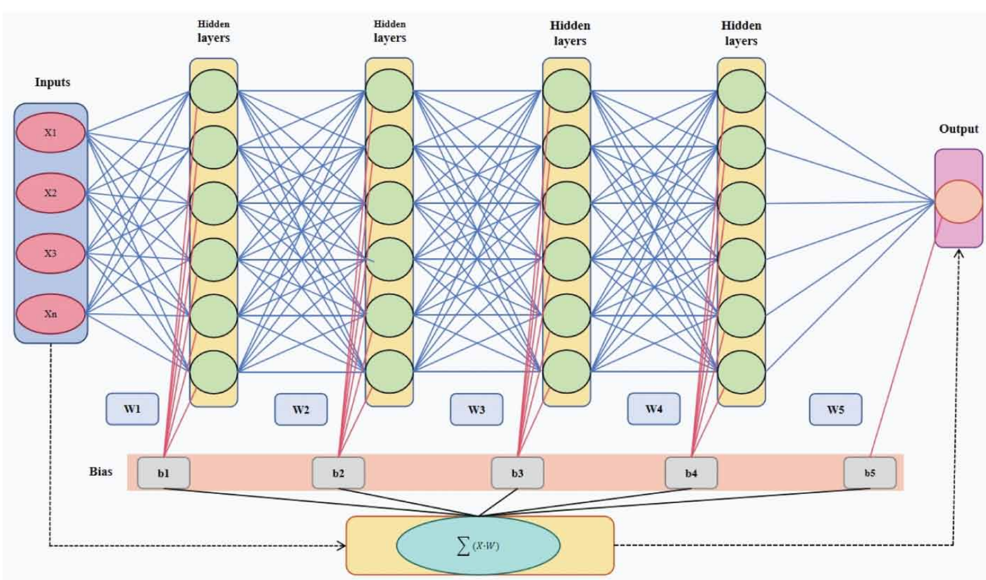
$$p_i^{(t+1)} = \begin{cases} x_i^{(t+1)}, & if\, f\left(x_i^{(t+1)}\right) < f\left(p_i^{(t)}\right) \\ p_i^{(t)}, & otherwise \end{cases} \tag{16}$$

where $p_i^{(t+1)}$ is the best-known position of particle i at iteration $(t+1)$, $f(\cdot)$ is the objective function to be minimized.

$$g^{(t+1)} = \begin{cases} x_i^{(t+1)}, & if\, f\left(x_i^{(t+1)}\right) < f\left(g_i^{(t)}\right) \\ g_i^{(t)}, & otherwise \end{cases} \tag{17}$$

where $g^{(t+1)}$ is the best-known position among all particles in the swarm at iteration $(t+1)$.

Figure 4. Flow chart of the Softmax model



$$\omega^{(t+1)} = \omega^{(t)} - \frac{\omega^{(t)} - \omega_{min}}{max\_iterations} \times t \tag{18}$$

where $\omega^{(t+1)}$ is the inertia weight at iteration $(t+1)$, ω_{min} is the minimum desired inertia weight, and $max\_iterations$ is the maximum number of iterations.

$$c_1^{(t+1)} = c_{1_{initial}} + \frac{c_{1_{initial}} - c_{1_{final}}}{max\_iterations} \times t \tag{19}$$

where $c_1^{(t+1)}$ is the cognitive acceleration coefficient at iteration $(t+1)$, $c_{1_{initial}}$ is the initial cognitive acceleration coefficient, and $c_{1_{final}}$ is the final desired cognitive acceleration coefficient.

$$c_2^{(t+1)} = c_{2_{initial}} + \frac{c_{2_{final}} - c_{2_{initial}}}{max\_iterations} \times t \tag{20}$$

where $c_2^{(t+1)}$ is the social acceleration coefficient at iteration $(t+1)$, $c_{2_{initial}}$ is the initial social acceleration coefficient, and $c_{2_{final}}$ is the final desired social acceleration coefficient.

EXPERIMENTS

Experimental Environment

Hardware Configuration

In this study, we used a high-performance computing cluster for model training and experiments. The computing cluster includes dozens of server nodes, and each server is equipped with the following hardware configuration:

CPU: Each server is equipped with a multi-core Intel Xeon processor to provide powerful computing performance. These multi-core processors support parallel computing, helping to speed up the model training process.Specifically, we are using Intel Xeon E5 series processors, with 2 processors per node for a total of 32 physical cores.

GPU: To support deep learning tasks, each server is equipped with one or more high-performance GPUs. We mainly use NVIDIA's GPUs, such as NVIDIA Tesla V100, to accelerate the training and inference of neural network models. Each server node is equipped with four NVIDIA Tesla V100 GPUs. Each GPU has 5,120 CUDA cores, providing powerful parallel computing capabilities.

Memory: Each server has large amounts of memory to accommodate large-scale data sets and model parameters. Typically, we configure at least 256 GB of memory to ensure sufficient memory space to load large deep learning models and data.

Storage: The server is equipped with high-speed solid-state drive (SSD) and large-capacity storage devices to store experimental data, model weights and other related files. Each server node is equipped with 1 TB of SSD for fast data reading and writing, and at least 20 TB of storage capacity for long-term data storage and backup.

Software Configuration

To build and train deep learning models, we use the following main software and tools:

Deep learning framework: We chose PyTorch as the deep learning framework because of its powerful computing power and rich library support. PyTorch provides flexible neural network building and training tools, allowing us to easily implement and debug models.Specifically, we are using PyTorch version 1.8.0.

Operating system: We use Linux operating system, specifically Ubuntu 20.04 LTS. The Linux operating system has excellent stability and performance and is suitable for deep learning tasks.

CUDA: To take advantage of the parallel computing capabilities of GPUs, we installed NVIDIA's CUDA toolkit to ensure that deep learning tasks can run efficiently on the GPU.We are using CUDA version 11.1, which is compatible with NVIDIA Tesla V100 GPU.

Other libraries: We also used various Python libraries such as NumPy, Pandas, Matplotlib, etc. for data processing, visualization and analysis.These libraries played a key role in the experimental process and helped us process and analyze the data.

Experimental Data Set

In order to deeply study the changing patterns of market demand and verify the effectiveness of the LSTNet-Prophet-PSO model in demand forecasting, this study designed a series of experiments involving data sets from different sources and properties. Four main data sets, including Online Retail Data Set, Supply Chain Data Set, Bureau of Labor Statistics Data and Awesome Public Data Sets, were used in the experiment. These data sets provide actual values in retail, supply chain, labor statistics, and multiple public fields, respectively, aiming to obtain a comprehensive view of market demand from different angles and dimensions and to test the generalization ability and accuracy of the forecast model.

The Online Retail Data Set is a collection of international online retail transaction data, sourced from a company registered in the UK. This data set contains all transaction records from 2010 to 2011, including fields such as order number, product code, product description, quantity, transaction time, unit price, customer ID, and country. Its data characteristics can help us understand consumer purchasing patterns and preferences (Jeena et al., 2023). This data set plays the role of evaluating the model's ability to process real-time, dynamic market data in this experiment and provides real and rich sample data in the retail industry to help achieve more accurate consumer demand forecasts.

Supply Chain Data Set provides data on each link in the supply chain, collected from manufacturing companies of different sizes, and records information on each link such as material procurement, production scheduling, inventory management, and product distribution (Hamledari & Fischer, 2021). The data set details the operational details of supply chain processes, reflecting the complexity of supply chain operations and their connection to market demand. In this study, this data set helps the model grasp the influencing factors of the supply chain, thereby accurately predicting the market demand for specific products and its changing trends and effectively improving supply chain efficiency.

Bureau of Labor Statistics Data is provided by the U.S. Bureau of Labor Statistics and compiles statistical data on employment, wages, consumer spending, and price indexes (Raghupathi & Raghupathi, 2020). With its official and authoritative nature, this data set provides reliable support for studying macroeconomic trends and labor market changes. In this experiment, this data set can be used to verify the applicability of the model in predicting broader economic indicators, especially how to correlate labor statistics and market demand forecasts, reflecting the predictive value of the model at the macro level.

Awesome Public Data Sets is a collection of public data covering many industries and fields, such as finance, medical care, social networks, transportation, and many other fields. Due to the diversity and extensiveness of these data, they provide rich scenario verification for the model and test the model's versatility and adaptability in different fields. In this article, through Awesome Public Data Sets, the model has the opportunity to come into contact with various nonstandardized and unstructured data sources, testing and enhancing its ability to process data in complex environments (Maroufkhani et al., 2019). Using these public data sets, this experiment is able to demonstrate the model's predictive performance in a wide range of real-world application contexts.

Experimental Setup and Details

This article builds the LSTNet-Prophet model and uses the PSO algorithm for parameter tuning to predict changes in market demand, thereby promoting enterprises to optimize inventory management, improve operational efficiency, and gain competitive advantages. To ensure accuracy and reproducibility, experimental details need to be carefully designed. The experimental setup and details are as follows.

Data Preprocessing

In the first step of the experiment, we performed data preprocessing to ensure data quality and usability. The key steps of data preprocessing include the following points:

Data cleaning: First, we performed data cleaning on the data set used to remove outliers and missing data.
Data standardization: In order to unify data from different data sets to the same scale, we standardize the data and use normalization to map it to the range 0 to 1.
Data splitting: In order to train and verify the model, we divide the data set into a training set, a verification set, and a test set.

Model Training and Optimization

Hyperparameter Optimization Settings. PSO hyperparameters:

- Number of particles: 20-50, balancing the breadth and depth of search.
- Number of iterations: 100-200 times to ensure sufficient search.
- Inertia weight (w): Adjust within the range of 0.5 to 0.9 to control the particle's ability to explore and exploit.
- Learning factors: Both the individual learning factor (c1) and the social learning factor (c2) are set to 2.0 to balance individual and group learning.

LSTNet structural hyperparameters (initial search range of PSO):

- Number of CNN layers: 1-2 layers to adapt to different abstraction requirements.
- Number of CNN filters: 32-64, used to capture different patterns in time series.
- Number of RNN layers: 1-2 layers to capture the deep timing dependencies of time series data.
- Number of RNN units: 20-100, balancing model complexity and resource consumption.
- Learning rate: 0.001-0.01, ensuring a balance between convergence speed and accuracy.
- Batch size: 98-128, fits within memory capacity and maintains sufficient batch diversity.

Prophet model hyperparameters (initial search range of PSO):

- Seasonal cycle length: Adapt to seasonal changes in the data set.

- Changepoint prior scale (changepoint_prior_scale): Adjusted in the range 0.01 to 0.1 to allow the model to capture trend changes.
- Seasonality prior scale (seasonality_prior_scale): In the range of 10 to 25, used to adjust the flexibility of the seasonal component.

Model Training and Optimization Settings

- Training cycle: Set the number of Epochs to a maximum of 500, but if the performance on the validation set does not improve within 50 consecutive epochs, the training will be terminated early to prevent overfitting.
- Validation interval: Use the validation set after each epoch to evaluate model performance.
- Learning rate adjustment strategy: Initially set to 0.001. If the performance no longer improves, multiply the decay rate by 0.95 to reduce the learning rate.

Ablation Experiments

This paper designed a series of ablation experiments to evaluate the impact of the LSTNet-Prophet model combined with various components of the PSO algorithm on the performance of the final model. Experiments were divided into the following groups:

Without PSO optimization:The first set of ablation experiments removed the PSO optimization component and only used the traditional parameter optimization method of the LSTNet-Prophet model for experiments. This set of experiments revealed the additional value of PSO in model performance optimization compared with traditional methods.

Remove the Prophet module:In the second set of ablation experiments, we only used the LSTNet module for prediction, excluding the role of the Prophet module. In this way, we independently evaluated the performance of LSTNet in processing time series data and further understood the gain when combining LSTNet with Prophet.

Remove the LSTNet module:In the third set of ablation experiments, we removed the LSTNet module and only used Prophet alone for time series prediction. This allowed us to evaluate the baseline performance of the Prophet model without LSTNet support and thus understand the contribution of LSTNet to the performance of the entire combined model.

Comparative Experiments

This article also conducted a series of comparative experiments to compare and analyze the optimization process of the LSTNet-Prophet model: Adam optimizer, Bayesian optimization, optimizer based on attention mechanism, and PSO performance of the algorithm. The following are the specific settings for these different optimization strategies:

- **Adam optimizer:** The learning rate is set to 0.001 to avoid excessive convergence leading to local optimality. The $\beta 1$ parameter is set to 0.9, which controls the exponential decay rate of the first-order moment estimate to reduce oscillations in the running average of the gradient. The $\beta 2$ parameter is set to 0.999, which is used to control the exponential decay rate of the second-order moment estimate to reduce oscillations in the running average of the squared gradient.
- **Bayesian optimization:** Use Gaussian process to establish a proxy for a black box function, whose model is unknown, and improve the proxy model through continuous updates. The optimizer explores and exploits a balance of trade-offs to further explore and find the global optimum near a known local optimum.
- **Optimizer based on attention mechanism:** Pays attention to the importance of different parts in the model, and dynamically adjusts the direction and magnitude of parameter updates. One

can try to cooperate with or replace the traditional gradient descent method to focus on features that contribute greatly to prediction to optimize model performance.

- **PSO:** Set 50 particles as candidate solutions in the search space and guide the particle flight to the best global position. The maximum number of iterations is set to 100 to ensure that the search process can converge to a better solution within a limited time.

In these experiments, the LSTNet-Prophet model will be trained using the different optimization strategies mentioned above. Each optimization method is run independently on the same data set and under the same hardware conditions to ensure a fair comparison of results. Performance evaluation will be based on multiple indicators such as prediction accuracy, convergence speed, and computing resource consumption. Through these comparative experiments, we will be able to deeply understand the performance of various optimization strategies in complex time series prediction tasks and their potential advantages and disadvantages, thereby providing more accurate model training guidance for practical applications.

Model Evaluation

Model Performance Metrics: Evaluate the accuracy of the model through performance indicators such as MAE, MAPE, RMSE, MSE, etc. These indicators can intuitively reflect the model's prediction error on time series data and help us understand how well the model fits the real values. MAE was chosen because it is insensitive to outliers and can provide a robust assessment of the overall performance of the model. MAPE expresses the error in the form of a percentage, which can better reflect the performance of the model on data of different scales, making the evaluation results more interpretable. At the same time, RMSE and MSE, as common mean square error indicators, can effectively measure the difference between the model prediction value and the actual value.

Evaluate the efficiency of the model through performance indicators such as parameters, flops, inference time, training time, etc. These indicators help evaluate the model's performance in terms of resource utilization and computing efficiency, providing a more comprehensive performance evaluation for actual application scenarios.

Taking these indicators into consideration, we can more comprehensively and objectively evaluate the model's performance in market demand forecasting.

Cross-Validation: To ensure the reliability and generalization ability of the model, we used cross-validation techniques. Typically, we use k-fold cross-validation, where k is usually set to 5 or 10 to avoid overfitting and evaluate the model's performance on different subsets. We randomly divide the data set into k subsets, and then use each subset in turn as a validation set, and the remaining subsets as a training set. We perform this process multiple times to obtain performance evaluation results on different validation sets, and then calculate the average performance to obtain a more reliable evaluation. This helps ensure model performance stability and consistency.

Experimental Results and Analysis

As shown in Table 1, the performance of various models on different data sets is measured by four key indicators: mean absolute error (MAE), mean absolute percentage error (MAPE), root mean square error (RMSE), and mean square error (MSE). Together, these indicators reflect the accuracy and stability of model predictions. MAE and MAPE focus more on measuring the average performance level, while RMSE and MSE give greater weight to large errors and therefore can reflect fluctuations in forecast results.

It can be seen from the experimental results that this model generally demonstrates relatively strong performance on all four indicators. Taking the Online Retail Data Set as an example, our method achieves a MAE of only 19.36, which is much lower than the second-placed Zhang's 21.38, and significantly lower than the top-ranked Ahmed's 46.65. MAPE also performs well, and our 6.24% is far better than all other models and well below the average, indicating that our proportional error is small and the model's prediction accuracy is high. On the Supply Chain Data Set, our method also shows obvious advantages in MAE and RMSE indicators. Especially on RMSE, our 2.16 is significantly lower than the lowest value of 5.42 among other models (obtained by the Ahmed model). The significant gap highlights the significant improvement in the prediction stability of our model. When analyzing the Bureau of Labor Statistics Data and Awesome Public Data Sets Collection, our method still maintains the lead, with the others showing MAPE of 4.9% and 4.97% respectively, which is the lowest 8.55% among other models (Jackwerth model in Bureau of Labor Statistics Data, performance on Awesome Public Data Sets Collection) and 9.98% (the performance of Zhang model on Awesome Public Data Sets Collection) are significantly lower. This demonstrates that our predictive models demonstrate high data adaptability and accuracy, both in the labor market and on broader public data sets. Overall, our method can be regarded as a highly accurate and stable prediction tool, suitable for a variety of data sets and analysis scenarios. In all tested data sets, its MAE, MAPE, RMSE and MSE indicators are better than other models, which shows that our method has significant advantages in reducing prediction errors and improving prediction accuracy. These results demonstrate the significant advantages of our method in reducing prediction errors and enhancing prediction accuracy.

Figure 5 is an intuitive visual display of the contents of Table 1. It clearly depicts the performance comparison of different models on multiple indicators, further emphasizing the stability and efficiency of our method.

As shown in Table 2, the performance analysis of our LSTNet-Prophet model on four data sets highlights its advantages. On the Online Retail Data Set, the LSTNet-Prophet model has a more optimized parameter amount and computational efficiency. Its parameter amount (Parameters(M)) is only 339.58M, which is lower than other models, such as the Wang model's 482.48M and the Mutamimah model's 692.49M. In addition, in terms of the number of floating-point operations (Flops(G)), the LSTNet-Prophet model performed at 3.54G, which is relatively low, reducing computing resource consumption. The inference time (Inference Time (ms)) is optimized to only 5.36 milliseconds, which is far better than competitors, such as the 12.94ms of the Mutamimah model. The crucial training time (Training Time(s)) is significantly reduced to 326.86 seconds, which is much faster than the 580.97 seconds of the Wang model and the 585.21 seconds of the Jackwerth

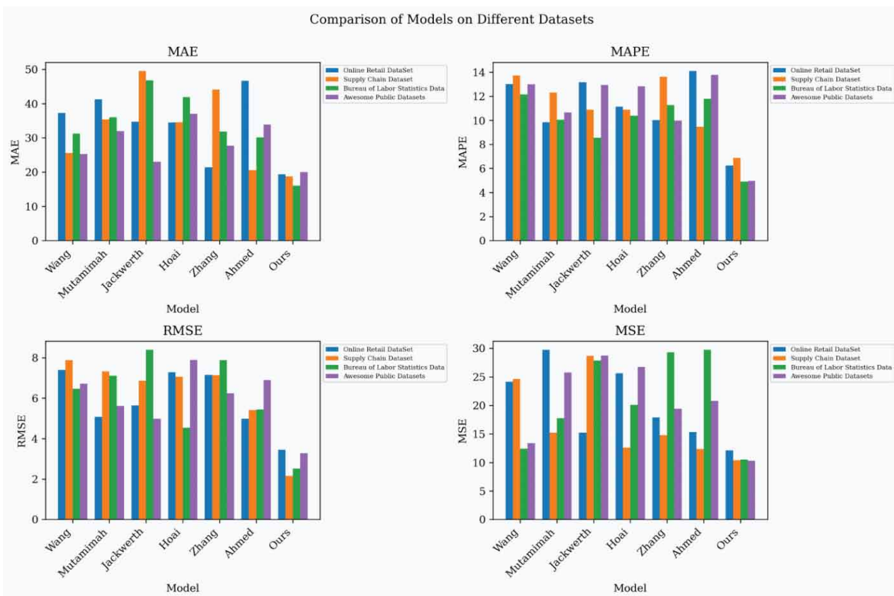
Table 1a. The model accuracy verification and comparison of different indicators on different data sets (part 1)

Model	Datasets							
	Online Retail Dataset				Supply Chain Dataset			
	MAE	MAPE	RMSE	MSE	MAE	MAPE	RMSE	MSE
Wang(H. Wang et al., 2023)	37.27	13.03	7.4	24.13	25.57	13.73	7.88	24.64
Mutamimah(Mutamimah et al., 2021)	41.21	9.85	5.08	29.75	35.32	12.31	7.32	15.22
Jackwerth(Jackwerth et al., 2023)	34.66	13.16	5.64	15.23	49.57	10.89	6.86	28.68
Hoai(Hoai et al., 2022)	34.45	11.13	7.29	25.66	34.57	10.88	7.06	12.6
Zhang(Y. Zhang et al., 2023)	21.38	10.03	7.15	17.9	44.11	13.62	7.14	14.78
Ahmed(Ahmed et al., 2022)	46.65	14.12	4.99	15.31	20.61	9.47	5.42	12.35
Ours	19.36	6.24	3.45	12.11	18.72	6.86	2.16	10.39

Table 1b. The model accuracy verification and comparison of different indicators on different data sets (part 2)

Model	Datasets							
	Bureau of Labor Statistics Data				Awesome Public Datasets			
	MAE	MAPE	RMSE	MSE	MAE	MAPE	RMSE	MSE
Wang	31.21	12.16	6.47	12.39	25.26	13	6.72	13.37
Mutamimah	36.02	10.04	7.12	17.74	31.95	10.65	5.62	25.76
Jackwerth	46.8	8.55	8.4	27.86	23.01	12.93	4.98	28.73
Hoai	41.87	10.38	4.54	20.09	37.06	12.84	7.9	26.76
Zhang	31.8	11.27	7.89	29.31	27.71	9.98	6.24	19.43
Ahmed	30.11	11.79	5.44	29.76	33.86	13.78	6.9	20.81
Ours	16.06	4.9	2.52	10.51	20.02	4.97	3.28	10.27

Figure 5. Comparison of model performance on different data sets



model. Our model also demonstrated its efficiency when processing the Supply Chain Data Set, with significant improvements in the number of parameters and inference time. Specifically, the number of parameters is 319.24M, which is much lower than the 722.90M of the Hoai model. The inference time is 5.62ms, and the training time is only 338.09 seconds. This once again highlights its rapid iteration capabilities and is ideal for enterprises under rapidly changing market conditions. For the Bureau of Labor Statistics Data and Awesome Public Data Sets Collection data sets, the LSTNet-Prophet model also demonstrated similar performance advantages. The optimization of its parameter configuration and processing time ensured that the overall prediction accuracy was improved without sacrificing prediction accuracy, computational efficiency, and responsiveness.

Figure 6 visualizes the contents of the table and intuitively represents the comparison between our LSTNet-Prophet model and other models on multiple key performance indicators. The histograms

Table 2a. The model efficiency verification and comparison of different indicators on different data sets (part 1)

Model	Datasets							
	Online Retail Dataset				Supply Chain Dataset			
	Parameters (M)	Flops (G)	Inference Time (ms)	Training Time (s)	Parameters (M)	Flops (G)	Inference Time (ms)	Training Time (s)
Wang	482.48	6.32	9.34	580.97	502.58	5.53	9.44	500.87
Mutamimah	692.49	7.27	12.94	755.82	721.30	7.56	12.85	659.14
Jackwerth	595.59	6.00	12.34	585.21	754.83	7.09	8.60	732.87
Hoai	678.99	7.11	10.84	692.86	722.90	8.14	11.09	726.11
Zhang	472.01	4.48	7.69	437.12	448.62	5.25	8.34	409.89
Ahmed	337.64	3.55	5.35	326.21	319.30	3.66	5.60	338.20
Ours	339.58	3.54	5.36	326.86	319.24	3.64	5.62	338.09

Table 2b. The model efficiency verification and comparison of different indicators on different data sets (part 2)

Model	Datasets							
	Bureau of Labor Statistics Data				Awesome Public Datasets			
	Parameters (M)	Flops (G)	Inference Time (ms)	Training Time (s)	Parameters (M)	Flops (G)	Inference Time (ms)	Training Time (s)
Wang	490.77	6.06	8.62	500.37	563.76	5.38	9.89	599.39
Mutamimah	839.11	7.55	12.76	759.83	622.30	7.61	13.62	811.04
Jackwerth	691.61	5.44	8.24	780.99	470.54	8.05	6.24	650.86
Hoai	790.84	6.67	10.98	694.72	760.05	7.05	10.75	724.35
Zhang	413.82	4.38	7.29	399.12	460.04	5.08	7.64	480.19
Ahmed	337.53	3.55	5.32	325.55	318.15	3.65	5.59	338.64
Ours	337.16	3.55	5.35	328.02	319.78	3.63	5.62	337.04

and curve graphs highlight the relative reductions in parameter volume, inference time, and training time, as well as the increase in computational resource usage efficiency, emphasizing the significant advantages of our model in enabling smart productivity transformation.

Table 3 shows the performance comparison of different modules based on various indicators (MAE, MAPE, RMSE and MSE) of multiple data sets. These results are derived from a series of carefully designed ablation experiments designed to evaluate the contribution of each component of our combined model (especially the PSO algorithm) to the overall performance. In this experiment, we evaluated the performance of the LSTNet-Prophet model after removing PSO to determine the role of particle swarm optimization on prediction accuracy. LSTNet-Prophet tends to perform worse without PSO optimization, which highlights the important role of PSO in fine-tuning model performance. Taking Online Retail Data Set as an example, after applying the PSO algorithm to the LSTNet-Prophet model (LSTNet-PSO), the MAE is reduced from 32.39 to 39.77. Although the MAE has increased, the decrease in RMSE and MSE shows that the PSO algorithm is reducing the prediction accuracy and utility in terms of volatility. Our method has significant improvements in all indicators compared to LSTNet-Prophet and LSTNet-PSO, which reflects the performance optimization of other components in our model structure. Our method outperforms the other three model configurations on the Supply Chain Data Set, Bureau of Labor Statistics Data, and Awesome Public Data Sets Collection data sets. The performance on MAE and MAPE is particularly outstanding. For example, on the Awesome

Figure 6. Comparison of model performance on different data sets

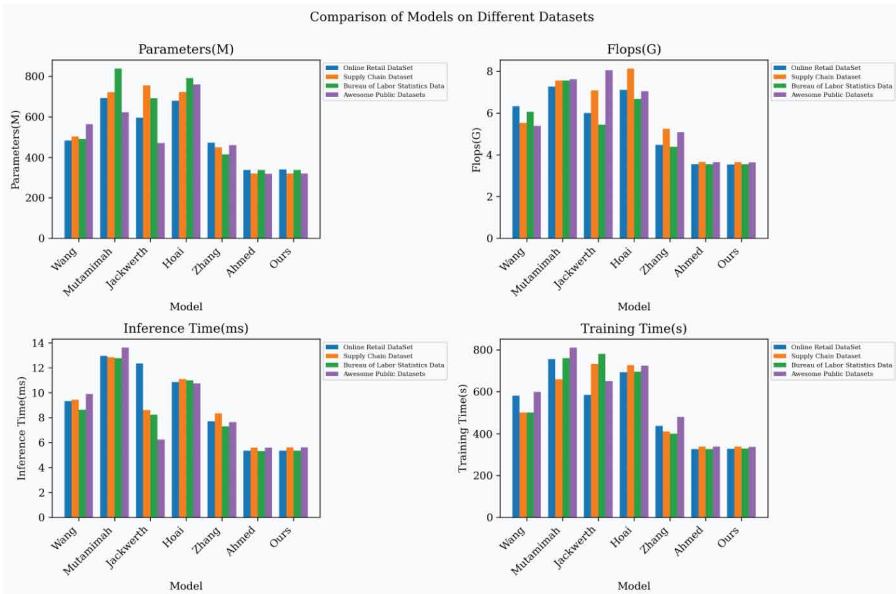


Table 3(a). Ablation experiments of PSO-optimized LSTNet-Prophet model based on different data sets (part 1)

Model	Datasets							
	Online Retail Dataset				Supply Chain Dataset			
	MAE	MAPE	RMSE	MSE	MAE	MAPE	RMSE	MSE
LSTNet-Prophet	32.39	14.43	8.33	22.84	38.62	14.02	5.81	21.31
LSTNet-PSO	39.77	13.72	5.78	15.66	31.83	13.93	6.72	21.05
Prophet-PSO	46.68	8.62	5.89	19.11	31.05	10.36	6.18	29.41
Ours	13.83	7.14	4.20	10.49	12.98	7.54	2.77	7.29

Table 3b. Ablation experiments of PSO-optimized LSTNet-Prophet model based on different data sets (part 2)

Model	Datasets							
	Bureau of Labor Statistics Data				Awesome Public Datasets			
	MAE	MAPE	RMSE	MSE	MAE	MAPE	RMSE	MSE
LSTNet-Prophet	44.71	12.54	4.79	29.08	46.67	8.71	6.29	30.37
LSTNet-PSO	48.72	10.93	6.90	25.40	47.45	15.11	5.62	13.66
Prophet-PSO	27.80	11.61	5.39	20.54	41.85	13.93	8.02	25.62
Ours	12.13	4.43	2.76	5.36	14.77	6.64	2.49	6.96

Public Data Sets Collection, our MAE (14.77) and MAPE (6.64%) are far lower than the performance of LSTNet-Prophet, LSTNet-PSO, and Prophet-PSO models.

Figure 7 visualizes the results in Table 3 and uses a line chart to compare the performance of different models on various performance indicators. Each subgraph corresponds to a performance indicator, showing the performance difference of the model on the same data set, including all model components and removing specific components, such as not using the PSO algorithm, one can clearly see at a glance after adding PSO optimization. Model performance has improved significantly, as well as the extent to which each individual component contributes to overall performance.

As shown in the results in Table 4, we analyzed the performance of four different optimization algorithms—Adam, Bayesian, AM (attention mechanism-based optimizer), and our own optimizer PSO—when processing different data sets.

It can be seen from the data results that the Adam optimizer is similar in parameter size, which is about 370M, and shows moderate training and inference time, especially the shortest inference time on the Supply Chain Data Set. The Bayesian optimizer has a slightly higher number of

Figure 7. Comparison of model performance on different data sets

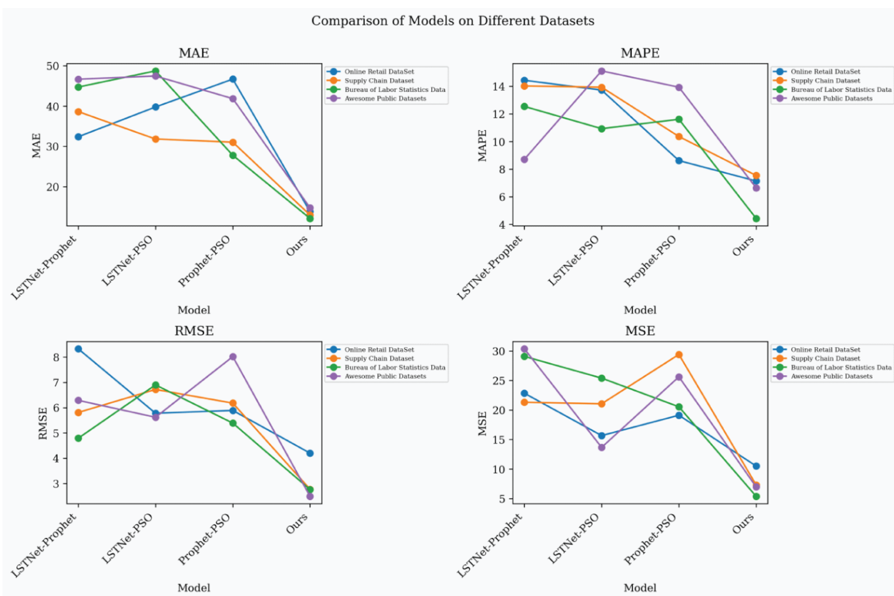


Table 4a. Ablation experiments on PSO module based on different data sets (part 1)

Model	Datasets							
	Online Retail Dataset				Supply Chain Dataset			
	Parameters (M)	Flops (G)	Inference Time (ms)	Training Time (s)	Parameters (M)	Flops (G)	Inference Time (ms)	Training Time (s)
Adam	370.83	275.99	249.75	304.42	365.18	372.69	222.85	406.38
Bayesian	384.4	301.06	262.97	285.33	270.53	395.54	369.28	330.6
AM	331.87	375.23	265.95	325.26	351.79	317.11	263.69	362.08
Ours	211.19	189.02	203.68	223.66	166.39	189.58	192.95	120.71

Table 4b. Ablation experiments on PSO module based on different data sets (part 2)

Model	Datasets							
	Bureau of Labor Statistics Data				Awesome Public Datasets			
	Parameters (M)	Flops (G)	Inference Time (ms)	Training Time (s)	Parameters (M)	Flops (G)	Inference Time (ms)	Training Time (s)
Adam	376.65	309.93	307.48	386.16	280.7	251.69	334.52	382.31
Bayesian	380.55	272.64	242.29	280.35	374.29	293.63	212.76	399.93
AM	306.09	309.22	232.24	288.21	351.22	287.13	391.81	400.06
Ours	144.97	142.56	234.91	190.86	212.54	214.71	204.53	202.5

parameters (384.4M) on the Online Retail Data Set and lower on other datasets, but overall shows higher computational complexity and inference time. The AM optimizer with attention mechanism is overall lower than Adam and Bayesian in various indicators, although it has the longest inference time in the Awesome Public Data Sets Collection. The most noteworthy is PSO, which shows clear advantages in all indicators, especially in terms of computational complexity and inference time, which are significantly lower than other competitors. Through comparison, we can see that our optimizer has significant advantages in efficiency and performance and is especially suitable for use in environments with limited computing resources or where fast response is required. The Adam optimizer shows good balance and is suitable for dealing with various situations, provided that the parameters are adjusted appropriately. The Bayesian optimizer seems to have an advantage when dealing with complex nonconvex optimization problems because it can effectively explore and utilize known information to find the global optimal solution. The AM optimizer based on the attention mechanism may be more suitable for processing data sets with a large number of parameters and complex model structures, even if its inference time is not always optimal.

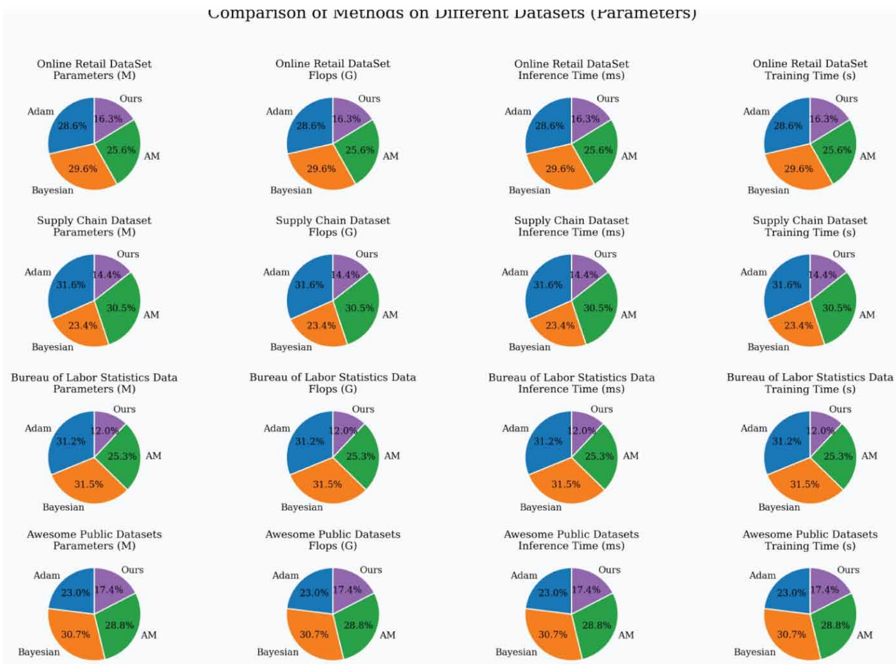
Figure 8 is a visualization of the results in Table 4. It includes 16 subgraphs, each subgraph corresponding to a specific indicator and data set combination. These subgraphs detail the performance of different optimizers in specific scenarios, with color coding and graph size used to differentiate the performance of each optimizer. Such visual comparison further demonstrates the superiority of the PSO optimization mechanism.

CONCLUSION AND DISCUSSION

In the current rapidly changing business environment, companies increasingly rely on accurate market demand forecasts and rapid decision-making capabilities to maintain competitiveness. This research successfully built an AI virtual assistant that integrates deep learning and business intelligence, using the LSTNet-Prophet model. The assistant successfully captures the complex characteristics of time series data by simultaneously mining long and short-term patterns. Combining the LSTNet and the Prophet model, it can not only handle the long-term dependence of time series data but also predict the trend and seasonal changes of time series. Significant improvements in prediction performance were achieved by fine-tuning hyperparameters using the PSO method.

Despite the satisfactory performance on the selected data sets, we would like to frankly point out that the current study has some limitations. First, we realize that the model's generalization ability and adaptability to unknown market data require further verification. Future research will focus on expanding the experimental data set to more comprehensively evaluate the applicability of the model in different scenarios. Second, given the factors of availability of computing resources across different enterprises, we acknowledge that our approach needs to be optimized to reduce dependence on high-

Figure 8. Comparison of model performance on different data sets



performance computing hardware. We will strive to explore more lightweight model structures and algorithm adjustments to ensure efficient operation in a variety of computing environments.

Future work will address these issues and aim to expand the model's application scope and improve its generalizability across enterprises of different sizes. Through further research and improvement, we expect the AI virtual assistant to be able to better handle the huge and complex market data, thereby performing demand forecasting and strategic decisions in a more efficient and accurate manner. We expect this technology to become a powerful tool to promote the intelligent transformation of enterprises and improve the efficiency of internal collaboration, providing enterprises with solid data support in a rapidly changing market.

ACKNOWLEDGMENT

This research was supported by Zhejiang Province Philosophy and Social Science Planning, Zhijiang Youth Special Project (Grant No. 24ZJQN011Y). This research was also supported by Research on Risk Optimization Path of Green Financial Market Under the Goal of "Double Carbon" (Grant No. EGD23QD04) and the Open Project Program of Shanghai Innovation Center of Reverse Logistics and Supply Chain, Shanghai Polytechnic University.

REFERENCES

Ahmed, F., Faraz, N. A., Ahmad, N., & Iqbal, M. K. (2022). Supportive leadership and post-adoption use of MOOCs: The mediating role of innovative work behavior. *Journal of Organizational and End User Computing*, *34*(1), 1–23. doi:10.4018/JOEUC.308813

Allal-Chérif, O., Simón-Moya, V., & Ballester, A. C. C. (2021). Intelligent purchasing: How artificial intelligence can redefine the purchasing function. *Journal of Business Research*, *124*, 69–76. doi:10.1016/j.jbusres.2020.11.050

Angelopoulos, A., Michailidis, E. T., Nomikos, N., Trakadas, P., Hatziefremidis, A., Voliotis, S., & Zahariadis, T. (2019). Tackling faults in the industry 4.0 era—A survey of machine-learning solutions and key aspects. *Sensors (Basel)*, *20*(1), 109. doi:10.3390/s20010109 PMID:31878065

Bai, S., Yin, Y., Wu, Y., Zhang, J. Z., Yu, Y., & Jasimuddin, S. M. (2022). Consumer engagement on social networking sites: The antecedents and mediating mechanism. *Journal of Organizational and End User Computing*, *34*(1), 1–19. doi:10.4018/JOEUC.307567

Belhadi, A., Mani, V., Kamble, S. S., Khan, S. A. R., & Verma, S. (2021). Artificial intelligence-driven innovation for enhancing supply chain resilience and performance under the effect of supply chain dynamism: An empirical investigation. *Annals of Operations Research*, 1–26. doi:10.1007/s10479-021-03956-x PMID:33551534

Chowdhury, S., Budhwar, P., Dey, P. K., Joel-Edgar, S., & Abadie, A. (2022). AI-employee collaboration and business performance: Integrating knowledge-based view, socio-technical systems and organisational socialisation framework. *Journal of Business Research*, *144*, 31–49. doi:10.1016/j.jbusres.2022.01.069

Collaboration, C. (2020). Identification of heavy, energetic, hadronically decaying particles using machine-learning techniques. *Journal of Instrumentation : An IOP and SISSA Journal*.

Dogan, A., & Birant, D. (2021). Machine learning and data mining in manufacturing. *Expert Systems with Applications*, *166*, 114060. doi:10.1016/j.eswa.2020.114060

Gao, G., Xu, Z., Li, J., Yang, J., Zeng, T., & Qi, G.-J. (2023). CTCNet: A CNN-transformer cooperation network for face image super-resolution. *IEEE Transactions on Image Processing*, *32*, 1978–1991. doi:10.1109/TIP.2023.3261747 PMID:37030697

Gao, X., Yan, L., Wang, G., & Gerada, C. (2021). Hybrid recurrent neural network architecture-based intention recognition for human-robot collaboration. *IEEE Transactions on Cybernetics*. PMID:34637387

Grek, S. (2020). Prophets, saviours and saints: Symbolic governance and the rise of a transnational metrological field. *International Review of Education*, *66*(2-3), 139–166. doi:10.1007/s11159-020-09844-z

Hamledari, H., & Fischer, M. (2021). The application of blockchain-based crypto assets for integrating the physical and financial supply chains in the construction & engineering industry. *Automation in Construction*, *127*, 103711. doi:10.1016/j.autcon.2021.103711

Hoai, T. T., Hung, B. Q., & Nguyen, N. P. (2022). The impact of internal control systems on the intensity of innovation and organizational performance of public sector organizations in Vietnam: The moderating role of transformational leadership. *Heliyon*, *8*(2), e08954. doi:10.1016/j.heliyon.2022.e08954 PMID:35243075

Jackwerth, T., Afghani, N., Daimer, S., Lindner, R., & Wittmann, F. (2023). *Public sector organisations as agents of transformations: A framework for analysing structural changes within public sector organisations*. PSO.

Jeena, S., Chaudhary, A., & Thakur, A. (2023). Implementation & analysis of online retail dataset using clustering algorithms. *2023 4th International Conference on Intelligent Engineering and Management (ICIEM)*.

Li, X., Sengupta, T., Mohammed, K. S., & Jamaani, F. (2023). Forecasting the lithium mineral resources prices in China: Evidence with Facebook prophet (Fb-P) and artificial neural networks (ANN) methods. *Resources Policy*, *82*, 103580.

Liang, G., Zhao, A., & Xin, X. (2022). Path of regional economic transformation and upgrading based on recurrent neural network. *Computational Intelligence and Neuroscience*, *2022*, 2022. doi:10.1155/2022/1547837 PMID:35685129

Lu, Y., Liu, L., Panneerselvam, J., Yuan, B., Gu, J., & Antonopoulos, N. (2019). A GRU-based prediction framework for intelligent resource management at cloud data centres in the age of 5G. *IEEE Transactions on Cognitive Communications and Networking*, *6*(2), 486–498. doi:10.1109/TCCN.2019.2954388

Maroufkhani, P., Wagner, R., Wan Ismail, W. K., Baroto, M. B., & Nourani, M. (2019). Big data analytics and firm performance: A systematic review. *Information (Basel)*, *10*(7), 226. doi:10.3390/info10070226

Min, Q., Lu, Y., Liu, Z., Su, C., & Wang, B. (2019). Machine learning based digital twin framework for production optimization in petrochemical industry. *International Journal of Information Management*, *49*, 502–519. doi:10.1016/j.ijinfomgt.2019.05.020

Mutamimah, M., Alifah, S., Gunawan, G., & Adnjani, M. D. (2021). ICT-based collaborative framework for improving the performance of zakat management organisations in Indonesia. *Journal of Islamic Accounting and Business Research*, *12*(6), 887–903. doi:10.1108/JIABR-05-2020-0154

Nguyen, H., Vu, T., Vo, T. P., & Thai, H.-T. (2021). Efficient machine learning models for prediction of concrete strengths. *Construction & Building Materials*, *266*, 120950. doi:10.1016/j.conbuildmat.2020.120950

Pagliari, D. J., Chiaro, R., Macii, E., & Poncino, M. (2020). Crime: Input-dependent collaborative inference for recurrent neural networks. *IEEE Transactions on Computers*, *70*(10), 1626–1639.

Raghupathi, V., & Raghupathi, W. (2020). Healthcare expenditure and economic performance: Insights from the United States data. *Frontiers in Public Health*, *8*, 156. doi:10.3389/fpubh.2020.00156 PMID:32478027

Shahaab, A., Khan, I. A., Maude, R., Hewage, C., & Wang, Y. (2023). Public service operational efficiency and blockchain–A case study of Companies House, UK. *Government Information Quarterly*, *40*(1), 101759. doi:10.1016/j.giq.2022.101759

Shen, X., Han, D., Guo, Z., Chen, C., Hua, J., & Luo, G. (2023). Local self-attention in transformer for visual question answering. *Applied Intelligence*, *53*(13), 16706–16723. doi:10.1007/s10489-022-04355-w

Trakadas, P., Simoens, P., Gkonis, P., Sarakis, L., Angelopoulos, A., Ramallo-González, A. P., Skarmeta, A., Trochoutsos, C., Calvo, D., Pariente, T., Chintamani, K., Fernandez, I., Irigaray, A. A., Parreira, J. X., Petrali, P., Leligou, N., & Karkazis, P. (2020). An artificial intelligence-based collaboration approach in industrial iot manufacturing: Key concepts, architectural extensions and potential applications. *Sensors (Basel)*, *20*(19), 5480. doi:10.3390/s20195480 PMID:32987911

Wang, F., Laili, Y., & Zhang, L. (2022). Trust evaluation for service composition in cloud manufacturing using GRU and association analysis. *IEEE Transactions on Industrial Informatics*, *19*(2), 1912–1922. doi:10.1109/TII.2022.3199268

Wang, H., Ye, X., Li, Y., & Zhu, G. (2023). Remaining useful life prediction for lithium-ion batteries based on improved mode decomposition and time series. *Sustainability (Basel)*, *15*(12), 9176. doi:10.3390/su15129176

Wang, L. (2022). Performance evaluation of knowledge sharing in an industry-university-research alliance based on PSO-BPNN. *Computational Intelligence and Neuroscience*, *2022*, 2022. doi:10.1155/2022/1283588 PMID:35592715

Wang, W., Ma, B., Hua, X., Zou, B., Zhang, L., Yu, H., Yang, K., Yang, S., & Liu, X. (2023). End-cloud collaboration approach for state-of-charge estimation in lithium batteries using CNN-LSTM and UKF. *Batteries*, *9*(2), 114. doi:10.3390/batteries9020114

Wu, C. H. (2022). An empirical study on selection, evaluation, and management strategies of green suppliers in manufacturing enterprises. *Journal of Organizational and End User Computing*, *34*(1), 1–18. doi:10.4018/JOEUC.307568

Wu, C. L., & Horng, S.-M. (2022). Social commerce intention, social interaction, and social support: Moderating role of social anxiety. *Journal of Organizational and End User Computing*, *34*(1), 1–23. doi:10.4018/JOEUC.307565

Wu, H., Liang, Y., & Gao, X.-Z. (2023). Left-right brain interaction inspired bionic deep network for forecasting significant wave height. *Energy*, *278*, 127995. doi:10.1016/j.energy.2023.127995

Yang, M., Moon, J., Yang, S., Oh, H., Lee, S., Kim, Y., & Jeong, J. (2022). Design and implementation of an explainable bidirectional lstm model based on transition system approach for cooperative ai-workers. *Applied Sciences (Basel, Switzerland)*, *12*(13), 6390. doi:10.3390/app12136390

Yuan, Q., Ma, C., Liu, J., Gui, H., Li, M., & Wang, S. (2022). Correlation analysis-based thermal error control with ITSA-GRU-A model and cloud-edge-physical collaboration framework. *Advanced Engineering Informatics*, *54*, 101759. doi:10.1016/j.aei.2022.101759

Zhang, H., Wang, C., Tian, S., Lu, B., Zhang, L., Ning, X., & Bai, X. (2023). Deep learning-based 3D point cloud classification: A systematic survey and outlook. *Displays*, *79*, 102456. doi:10.1016/j.displa.2023.102456

Zhang, J., Li, C., Liu, G., Min, M., Wang, C., Li, J., Wang, Y., Yan, H., Zuo, Z., Huang, W., & Chen, H. (2022). A CNN-transformer hybrid approach for decoding visual neural activity into text. *Computer Methods and Programs in Biomedicine*, *214*, 106586. doi:10.1016/j.cmpb.2021.106586 PMID:34963092

Zhang, Y., Shang, K., Cui, Z., Zhang, Z., & Zhang, F. (2023). Research on traffic flow prediction at intersections based on DT-TCN-attention. *Sensors (Basel)*, *23*(15), 6683. doi:10.3390/s23156683 PMID:37571466

Zheng, Y., Zuo, X., Zuo, W., Liang, S., & Wang, Y. (2021). Bi-LSTM+GCN causality extraction based on time relationship. *Journal of Jilin University Science Edition, 59*(3), 643-648. http://xuebao.jlu.edu.cn/lxb/CN/abstract/article_4363.shtml

Zhou, F., & Zhang, Y. (2023). Binding prediction algorithm of HLA-I and polypeptides based on pre-trained model ProtBert. *Journal of Jilin University Science Edition, 61*(3), 651-657. http://xuebao.jlu.edu.cn/lxb/CN/abstract/article_4774.shtml

Zhu, Y.-Q., Cai, Y.-M., & Zhang, F. (2022). Motion capture data denoising based on LSTNet autoencoder. *Wangji Wanglu Jishu Xuekan*, *23*(1), 11–20. doi:10.53106/160792642022012301002

Hongming (Chip) Li's current research focuses on leveraging various technologies to support education and multidisciplinary fields in the digital age. He specializes in using artificial intelligence and data strategies to solve practical problems. His recent research projects include using learning analytics for modeling and fine-tuning large language models for specific applications. He is also interested in multimodal data in NLP and AI.