MAGRU: Multi-layer Attention with GRU for Logistics Warehousing Demand Prediction

Ran Tian[1*], Bo Wang[1], and Chu Wang[1]
[1] College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070 China
[e-mail: tianran@nwnu.edu.cn]
[*]Corresponding author: Ran Tian

Received February 10, 2023; revised December 1, 2023; revised January 12, 2024; accepted March 7, 2024; published March 31, 2024

Abstract

Warehousing demand prediction is an essential part of the supply chain, providing a fundamental basis for product manufacturing, replenishment, warehouse planning, etc. Existing forecasting methods cannot produce accurate forecasts since warehouse demand is affected by external factors such as holidays and seasons. Some aspects, such as consumer psychology and producer reputation, are challenging to quantify. The data can fluctuate widely or do not show obvious trend cycles. We introduce a new model for warehouse demand prediction called MAGRU, which stands for Multi-layer Attention with GRU. In the model, firstly, we perform the embedding operation on the input sequence to quantify the external influences; after that, we implement an encoder using GRU and the attention mechanism. The hidden state of GRU captures essential time series. In the decoder, we use attention again to select the key hidden states among all-time slices as the data to be fed into the GRU network. Experimental results show that this model has higher accuracy than RNN, LSTM, GRU, Prophet, XGboost, and DARNN. Using mean absolute error (MAE) and symmetric mean absolute percentage error(SMAPE) to evaluate the experimental results, MAGRU's MAE, RMSE, and SMAPE decreased by 7.65%, 10.03%, and 8.87% over GRU-LSTM, the current best model for solving this type of problem.

Keywords: Gated Recurrent Unit, Attention mechanism, Commodities demand forecast, Time series, Neural Network

This work is supported in part by the National Natural Science Foundation of China (71961028), the Key Research and Development Program of Gansu(22YF7GA171), the University Industry Support Program of Gansu(2023QB-115), the Innovation Fund for Science and Technology-based Small and Medium Enterprises(23CXGA0136) and the Scientific Research Project of the Lanzhou Science and Technology Program (2018-01-58).

http://doi.org/10.3837/tiis.2024.03.001 **ISSN : 1976-7277**

1. Introduction

Logistics and warehouse lean management is a new interdisciplinary discipline combining the Internet and traditional manufacturing. It is gaining more and more attention in the industry because it provides an essential basis for product manufacturing and warehouse operations and significantly saves business costs. The evolution of logistics and warehouse management, driven by advancements in technology, especially the Internet and IoT, has revolutionized these fields. The application of lean management principles has further enhanced the efficiency, reduced waste, and optimized processes in these operations. This interdisciplinary discipline is increasingly important in the context of global supply chain complexities and the rising demand for rapid, cost-effective delivery systems.

In some scenarios, some commodities with high requirements for a preservation time frame, such as food, medicine, etc., need more accurate inventory forecasts to improve warehouse management sales and save operational costs while ensuring demand. Furthermore, logistics and warehousing demand forecasting is a time series forecasting problem for which the balance between model complexity and practical applicability is crucial, there is a need for integrating external data sources and employing hybrid models to address these challenges, potential improvements or new approaches could include exploring new machine learning techniques and enhancing model adaptability to varying data scales and external factors. The common forecasting methods include the auto regressive (AR) [1], the auto regressive integrated moving average (ARIMA) [2], etc. Neural network-based models such as recurrent neural network (RNN) and its variant long-short term memory (LSTM), gated recurrent unit (GRU) are also widely used in time series prediction [3-5], but the existing time series forecasting methods have the following limitations:

1) Machine learning methods cannot take full advantage of the complex external factors that influence demand, making it challenging to meet the requirements for forecast accuracy.

2) For highly customized commodities, which require long-term demand forecasting, the performance of existing models degrades when making long series forecasts, leading to bias in the prediction.

3) Commodity demand sampling is usually done on a weekly or daily basis, thus the data scale is small. Complex models are computationally expensive and tend to overfit for small-scale data sets. This makes many complex models unsuited to logistics warehouse demand forecasting scenarios.

All the above limitations show that the existing prediction models are challenging to explore the characteristics of commodity demand data and do not apply to the logistics and warehousing demand forecasting problem. Therefore, the implications of improved forecasting models are significant for business efficiency, cost savings, and customer satisfaction in the logistics sector, highlighting the necessity for ongoing research and development in this area.

In this paper, we propose a new prediction model based on a multi-layer attention mechanism combined with GRU to solve the logistics demand forecasting problem. The main contributions of this paper are summarized as follows:

1) We propose embedding the temporal features of the original data, capturing the mapping between demand and time, and mining the influence of hidden factors that are difficult to quantify on demand.

2) We propose a multi layer attention model, it focuses on long-term dependence and improves the prediction performance of the model in a long sequence.

3) We propose a new model that combines the base model GRU with the attention mechanism, improving model convergence speed, saving arithmetic power, and performing well on small-scale datasets.

The rest of this paper is organized as follows: Section 2 reviews the time series forecast and demand prediction problem and the literature. Then, we describe the warehouse demand prediction problem in Section 3. Section 4 describes the methodology of this study. Section 5 details the experiment setup and discusses the findings. Section 6 provides conclusions.

2. Related Works

Time series forecasting is the basis of warehouse demand forecasting and has remained a widely studied hot issue. Classical time series forecasting methods are ARIMA, ARCH, and GARCH [6-7]. However, there is a limitation that only linear models can be built, while realistic prediction tasks contain linear features and are often influenced by nonlinear features.

2.1 Machine learning methods

The application of machine learning in time series forecasting has revolutionized the way we approach these problems, transforming them into a form suitable for supervised learning. This transformation is primarily achieved through feature engineering, where raw time series data is manipulated to extract meaningful features that can be fed into various machine learning models. Such models excel in handling complex data structures and solving nonlinear problems through multi-variable collaborative regression. Among the most representative methods in this domain are XGboost [8], Prophet [9], and LightGBM [10-11]. These algorithms have been widely acknowledged for their efficiency and accuracy in forecasting tasks.

However, a significant limitation of these methods is the necessity of preprocessing data via feature engineering before model training. This process can often be laborious and time-consuming, requiring substantial domain knowledge and expertise. To circumvent these challenges, deep learning techniques have been increasingly employed in time series analysis [12]. Long Short-Term Memory (LSTM) networks and Gated Recurrent Unit (GRU) models, with their inherent ability to capture temporal dependencies, are specifically tailored to address the nuances of time series data. Additionally, sequence-to-sequence (Seq2seq) models [13] and WaveNet [14] have gained prominence for their effectiveness in forecasting. CNN model is often used to solve the problem of image classification. After improvement and development, it can also be used to solve the time series problem [15-16].

2.2 Combine the attention mechanism

Furthermore, the integration of Attention Mechanisms in models like the Transformer [17] has opened new avenues in time series forecasting. By focusing on specific parts of the data sequence, these mechanisms allow for more nuanced and accurate predictions, especially in long sequences. The Transformer model, originally designed for natural language processing tasks, has demonstrated remarkable results in time series forecasting due to its ability to capture long-range dependencies. Li et al. [18] in their study developed an attention mechanism recognition framework using recurrent neural networks (RNN). This framework is specialized for efficient classification of pulse streams with complex PRI modulation and in high spurious pulse and missing pulse ratio environments. While Zeng et al. [19] introduced a deep attention residual neural network (DARNN) which is designed to predict the remaining useful life (RUL) of a machine. The core advantage of DARNN is its ability to efficiently extract degraded features from signals. Zucchet et al. [20] demonstrated an RNN equipped with linear recursive layers and a feedforward path with multiplicative gating. This RNN is capable of accurately realizing linear self-attention, which is a key component of the Transformer Network. Ye et al. [21] proposed a novel temporal attention model that can assign appropriate weights to time-varying features during the prediction process. A bidirectional gated recurrent unit (GRU) based network intrusion detection model with hierarchical attention mechanism is presented [22].

Wu et al. [23], on the other hand, devised a hybrid network model that combines both shallow and deep networks and constructed these networks with separate positional attention mechanism and interactive multiple attention mechanism to capture multilevel features. Du et al. [24] proposed an innovative temporal attention encoder-decoder model specialized in multivariate time series prediction. Kavianpour et al. [25] used a hybrid network model that combines a convolutional neural network (CNN), a bi-directional long and short-term memory network (Bi-LSTM), and an Attentional Mechanisms deep learning model with Zero Order Holding (ZOH) preprocessing method as a way to predict the maximum magnitude and number of earthquakes in the next month. Cao et al. [26] proposed a novel model called Spatial-Temporal Adaptive Graph Convolutional Network (ST-AGCN) for skeleton-based action recognition by utilizing graph neural networks with the aim of extracting spatial-temporal features to improve the accuracy of action recognition.

Since current clustering algorithms lack effective representation learning, Diallo et al. [27] proposed a shrinkage self-encoder-based deep embedding clustering method for solving the problem of large-scale high-dimensional document data clustering. A systolic autoencoder is a variant of an autoencoder that introduces additional constraints in learning a low-dimensional representation of the data in order to facilitate the learning of a representation with local contrast preserving properties. The article then proposes a new framework, DECCA, which learns embedded representations of documents by maximizing the clustering loss and the reconstruction loss and uses local contrast preservation to improve the accuracy and efficiency of clustering. And Khan et al. [28] Wang presents an innovative approach to multiview clustering, tackling the issue that current methods overlook the flow structure of consensus representations in kernel space. This oversight often results in the neglect of the interrelations among different multiviews. In terms of processing multi-view data, current deep learning methods need to drive different neural networks independently for different viewpoints, resulting in low efficiency and high computational resource consumption, Diallo et al. [29] proposed an innovative Multi-view Deep Embedded Clustering (MDEC) model that employs a triple fusion technique designed to reduce the errors incurred in learning the features of each view and correlating data from multiple views. None of the existing methods in attributed graph clustering realize that the nonlinearity between two consecutive GCN layers is unnecessary for improving the performance of attributed graph clustering, and may even impair the efficiency of the model. Liao et al. [30] propose a novel deep linear graph attention model for attributed graph clustering consisting of an attention-based aggregation module and a similarity preserving module (DLGAMC) is used to solve the above problem.

2.3 Commodity demand prediction

Fite et al. [31] used a stepwise multiple linear regression model to predict the future logistics demand of the region. Bergman et al. [32] used Bayes' rule to improve the prediction accuracy of parts for which the established demand model in the new equipment program did not have enough time to develop. Bhuwalka et al. [33] proposed a Bayesian hierarchical model. This model is capable of simultaneously estimating specific demand parameters (e.g., price and income elasticities) as well as overall parameters for different regions and sectors. To improve the long-term prediction accuracy of feed requirement, Yang et al. [34] established a long-term dynamic prediction model of feed requirement by using the combined multiple regression model and predicted the variation trend of various factors affecting feed grain demand by using the ARIMA model. Huber et al. [35] proposed a decision support system that provides hierarchical forecasts at different organizational levels based on up-to-date point-of-sale data to support daily operations. Markers are used to extend the hierarchy of commodity clusters based on day sales patterns and apply multivariate ARIMA models to predict daily perishable food demand. He et al. [36] proposed a long short-term memory with particle swarm optimization method for e-commerce enterprises. The particle swarm optimization meta heuristic optimizes the number of iterations for training. Li et al. [37] constructed a composite model with an attention mechanism to predict sales. Linear and nonlinear features were captured using the prophet and GRU models with an attention mechanism. Guo et al. [38] propose a neural network model, to improve the MLP neural network using a deep learning training mechanism, and established a model based on the multilayer perceptron neural network algorithm, which provides a feasible method for industrial logistics demand forecasting. Cai et al. [39] proposed a multimodal data-based approach that combines spatial feature fusion and grouping strategies to construct a neural network prediction model for electronic goods demand. Gao et al. [40] proposed a supply chain network combining a neural network goods demand prediction method with a particle swarm optimization (PSO) algorithm, based on an analysis of a traditional supply chain and a modern supply chain model. This model aims to solve the problems of communication barriers, poor information flow, and supply-demand imbalance among enterprises.

3. Preliminary

3.1 Notation

Given $n(n>1)$time series for features and a time series for targets, We use (1) to represent the feature vectors with time step t, and x is daily demand data.

$$X_{feed\_en} = \{x_1, x_2, ..., x_t\} \tag{1}$$

We use (2) to represent the target vector where y is daily demand data.

$$Y_{feed\_de} = \{y_1, y_2, ..., y_t\} \tag{2}$$

We use (3) to represent the set of the output of the encoder and use (4) to represent the future values of the target series, where τ is the time step to be predicted.

$$x_t' = (\alpha_t^1 x_t^1, \alpha_t^2 x_t^2, ..., \alpha_t^n x_t^n)^T \tag{3}$$

$$Y' = \{y_{t+1}, y_{t+2}, ..., y_{t+\tau}\} \in R^{\tau} \tag{4}$$

3.2 Problem Statement

Given feature sequence as shown in (1) and target sequence as shown in (2), where samples of features and labels are daily demand data. We aim to find a mapping and predict the future values over the next τ time steps as shown in (5).

$$y'_{t+1}, y'_{t+2}, ..., y'_{t+\tau} = f(y_1, y_2, ..y_T, x_1, x_2, ...x_T) \tag{5}$$

4. MAGRU MODEL

In this section, we introduce Multi-layer Attention with Gated Recurrent Unit (MAGRU) and its detailed implementation. The overall framework of our proposed model is shown in **Fig. 1**.

The main two components are encoder and decoder. The input to the model are tensors containing T time step data, the feature dimension of each step data is N. The output of the model is a vector, and the length of the vector is equal to the prediction task.

The encoder uses an attention mechanism to capture key information about the input sequence. Firstly, the encoder calculates the importance of each sample by the attention layer. Specifically, the attention mechanism first generates a weight for each sample of the input sequence. This process is accomplished by multiplying the representation of the input sequence by a matrix, which is a trainable weight matrix, to produce a score (i.e., an "attention score"). This score is then a good reflection of the importance of each input sample relative to the entire sequence. Next, we normalize these attention scores using a softmax layer. By processing the softmax layer, we ensure that the sum of the attention scores of all samples is 1, so that each score can be interpreted as the relative importance of that sample in the whole sequence. After completing the above steps, these weighted samples are fed into a recurrent neural network (GRU in our model) for further processing. The GRU utilizes these weighted samples to update its hidden state, thus effectively encoding the key information of the input sequence.

The decoder uses an attention mechanism to fuse with the GRU network to make predictions about the target. The decoder has two input values, the value of the encoder output and the historical prediction value. Taking the same approach as encoder, decoder uses an attention layer and later predicted by GRU. Finally, the prediction results are output after two fully connected layers.

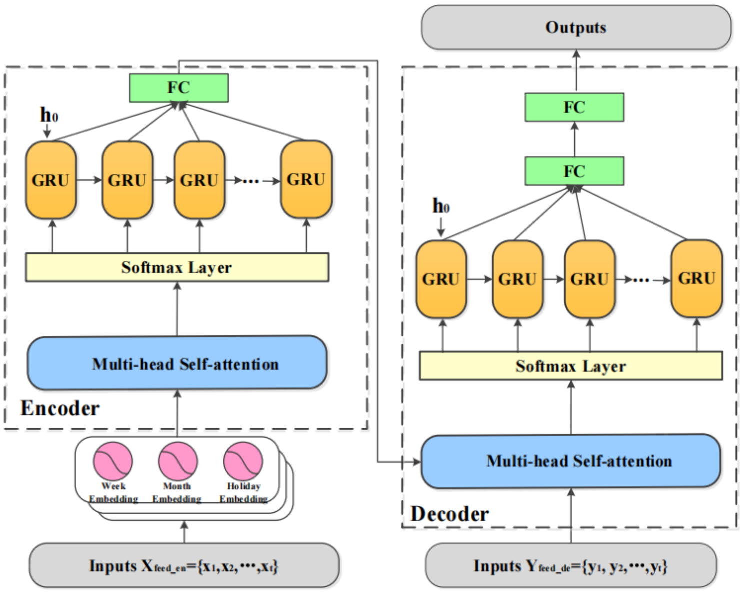

Fig. 1. Multi-layer Attention with Gated Recurrent Unit(MAGRU) module.

4.1 Embedding

The ability to capture long-term independence in forecasting requires global information such as hierarchical timestamps (week, month, and year) and agnostic timestamps (holidays, events). Since demand data is entered into the model in chronological order, it leads to neglecting the properties of time itself, which in turn leads to potential degradation of forecasting performance. For these reasons, we embed the original data before it is fed into the model.

The results after embedding are embedded as features in the original data so that the prediction can take into account the trend around a time period, and unknowable time stamps such as holidays can be obtained and used for prediction.

4.2 Encoder

The demand for a commodity is influenced by various factors. There are *k* time series in the encoder stage, but the importance of these k series for our future prediction is not the same. It is necessary to calculate its weight through the attention mechanism. The detailed process of the encoder attention stage is shown in **Fig. 2**.

An encoder is essentially an RNN. For the time series prediction problem, given an input sequence, the encoder is used to learn the mapping from *x* to h_t as shown in (6), where *t* represents the time step.

$$h_t = f(h_{t-1}, x_t) \tag{6}$$

Where:

h_t is the hidden layer state of the encoder at time *t*, which depends on the current input data x_t and the hidden layer state h_{t-1} at the previous time.

f is a mapping relation, that can be used in RNN, LSTM, or GRU, and our proposed model uses GRU as f to capture the temporal dependencies.

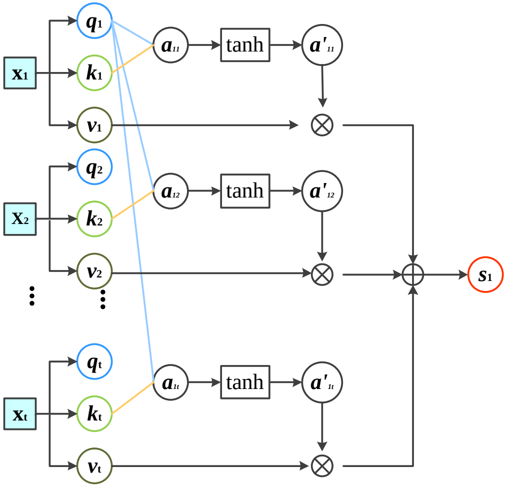

Fig. 2. The details of the encoder. Input is the data after embedding

Each GRU cell is controlled by a reset gate and an update gate with the following update algorithm:

$$R_t = \sigma(X_t W_{xr} + h_{t-1} W_{hr} + b_r) \tag{7}$$

$$Z_t = \sigma(X_t W_{xz} + h_{t-1} W_{hz} + b_z) \tag{8}$$

Where:

h is the number of hidden cells.

$x_t \in R^{n \times d}$ is a batch of inputs at a given time step *t*, with the number of samples *n* and the number of inputs *d*.

$h_{t\text{-}1}$ is the hidden state of the previous step.

R_t is the reset gate and Z_t is the update gate.

W_{xr},W_{hr},W_{xz},W_{hz} are the weight parameters to be trained, b_r, b_z are the bias parameters.

σ is the sigmoid function that controls the value domain at [0,1].

The hidden state at time *t* is:

$$h'_t = \tanh(X_t W_{x,h} + (R_t h_{t-1}) W_{h,h} + b_h) \tag{9}$$

$$h_t = (1 - Z_t) n_t + Z_t h_{t-1} \tag{10}$$

GRU presents a certain network memory over time, which can overcome the problem of RNN gradient vanishing. Reset gates help to capture short-term dependencies in the time series, and update gates help to capture long-term dependencies in the time series.

Given a sequence of *k*, we build an attention mechanism using the hidden state and the current GRU cell state, as shown in (11):

$$s_t^k = v_l^T \tanh(W_l(h_{t-1}; s_{t-1} + U_l x_{i,k} + b_l)) \tag{11}$$

In the above equation v_l, W_l and U_l are the parameters to be trained. For simplicity, we will bias the value b_l can be omitted.

$$\alpha_j^t = \frac{\exp(s_{t,j})}{\sum_{k=1}^{T} \exp(s_{t,k})} \tag{12}$$

α is the weight of the *j*th input sequence at moment *t*. The results derived from the attention mechanism are transformed into probability values by the softmax, so each data batch is weighted by its importance. A batch of samples is obtained, and the individual batch samples form a sequence of inputs to the GRU. With these weights, we can encode the timing in order of importance, and the encoder can selectively focus on some input sequences without having to treat all of them the same. So the output after the encoder is defined as (13):

$$x'_t = (\alpha_t^1 x_t^1, \alpha_t^2 x_t^2, ..., \alpha_t^n x_t^n)^T \tag{13}$$

Algorithm 1 The learning algorithm of the encoder

```
Input: A time step sequence $x$
Output: The result after the encoder $x_{encoder}$
Input the data set $x$ of feature n and length $T$.
Initialize the attention weight value $s_0$ and the hidden state $h_0$ of the GRU.
set $i = 0$
while $i < timestep$ do
    Compute the attention weights of each element $s_t^k$ by (11)
    Compute the probability by the softmax layer $\alpha_j^t$ by (12)
    Compute $x'_t$ by (13)
    Update GRU hidden layer status $h_t = f_1(h_{t\text{-}1}, x_t)$
    Compute $x_{encoder} = h_t$
    $i = i+1$
end while
return $x'$, $x_{encoder}$
```

4.3 Decoder

Encoding all the input sequence information into a fixed-length vector will result in missing data, and too-long sequences can cause poor characterization and prediction performance. The decoder was used in this study to solve these problems. We add the attention mechanism here to capture key sequences considering global data, to avoid performance degradation due to long sequences. We use GRU to decode the input information to take advantage of the correlation of the data in the time domain. The Decoder phase is shown in detail in **Fig. 3**.

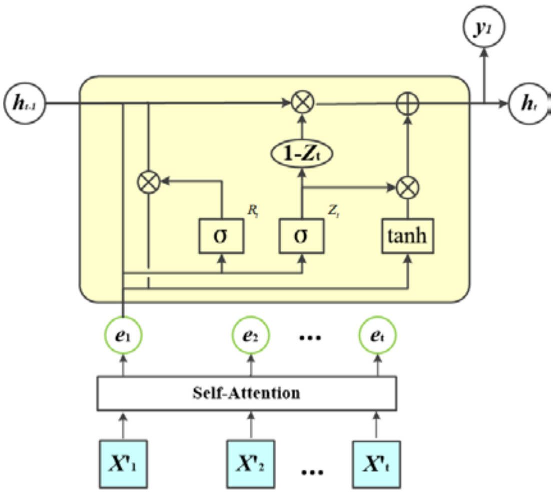

Fig. 3. The details of the decoder. Input is the data after Embedding. The input is Encoder data, and the output is the predicted target.

We use a temporal attention mechanism that automatically selects the state of the encoder output in the all-time series. Then calculate the encoder output state score through the attention mechanism in the decoder stage. Thus, the more important hidden states are selected as the data to be input to the GRU network.

$$e_t^i = v_d^T \tanh(W_d(d_{t-1}; s'_{t-1}) + U_d h_i + b_t) \quad (14)$$

Where *v*, *W* and *U* are all parameters to be trained.

$$\beta_t^i = \frac{\exp(e_t^i)}{\sum_{j=1}^{T} \exp(e_t^j)} \quad (15)$$

Where *e* is the calculated attention weight. By weighting *b* with the hidden state *h*, we can obtain the vector *c*, as shown in (16).

$$c_t = \sum_{i=1}^{T} \beta_t^i h_i \quad (16)$$

We use the c_t vector as an input to the GRU to obtain the hidden state d_t at the moment *t*, as shown in (17).

$$d_t = f(d_{t-1}, c_t) \quad (17)$$

f is a GRU cell, and our prediction target can be obtained in (18):

$$Y'_t = v_y^T(W_y(d_T; c_T) + b_w) + b_v \quad (18)$$

Where (d_T; e_T) is a series of two vectors, *W* and *v* are the model parameters, and b_w, b_v are the bias values. This linear model obtains the final prediction target.

$$y_T, y_{T+1}, \ldots y_{T+\tau} = v_y^T (W_y (d_T ; c_T) + b_w) + b_v \tag{19}$$

Algorithm 2 The learning algorithm of the decoder

```
Input: The historical predicted value $y$, the output of the Encoder layer $x$
Output: Predicted value $Y'$
Initialize the attention vector value $c_0$ and the state $d_0$ of the decoder.
set $i = 0$
while $i < timestep$ do
    Compute the attention weights of encoder $e_t^i$ by (14)
    Compute the probability by the softmax layer $\beta_j^t$ by (15)
    Compute $c_t$ by (16)
    Update $c_n$, $d_n$ by GRU
    Compute $Y'$ by two fully connected layer
    $i = i+1$
end while
return $Y'$
```

5. Experiments

5.1 Data selection and processing

This study selects the data from the 2016 Ali Tianchi demand forecasting and storage planning Competition (Links to datasets). The dataset contains historical data for a total of 5778 commodities from October 10, 2014 to December 27, 2015. The prediction target is the real demand of commodities.

The prediction accuracy of time series is mainly influenced by the internal trends of predicted target and other features. Internal trends include cyclical trends and seasonal trends. Internal features are often implied in data fluctuations and need to be mined by the model. External features are features derived from some field of data. Different external features have different degrees of influence on the model. This study selects some features that are important to the problem through feature engineering and removes the features that are not very relevant to the problem.

In the experiments, we filter the external features to achieve the dimension reduction of the data. On the one hand, do this can accelerate the training, and on the other hand, it can increase the robustness of the model. Because of the wide variety of all commodities and the different demand patterns for each commodity, the experiment selects the trend of individual commodities among them as the data set. Correlation analysis of other features including the predicted target leads to the following heat map. The heat map is shown in detail in **Fig. 4**. The horizontal and vertical coordinates of **Fig. 4** are the feature fields of this dataset.

For the selection of features, this study is based on three reasons: Firstly, the correlation between features and target variables is positive; secondly, the number of features is as small as possible, and the correlation between features is low; thirdly, the features have well explanation for the target. Based on these reasons and the correlation of external features to the prediction target, the experiment selects seven features, “num_gmv”, “amt_gmv”, “qty_gmv,amt_alipay”, “num_alipay”, “pv_ipv”, and “pv_uv”, as features.

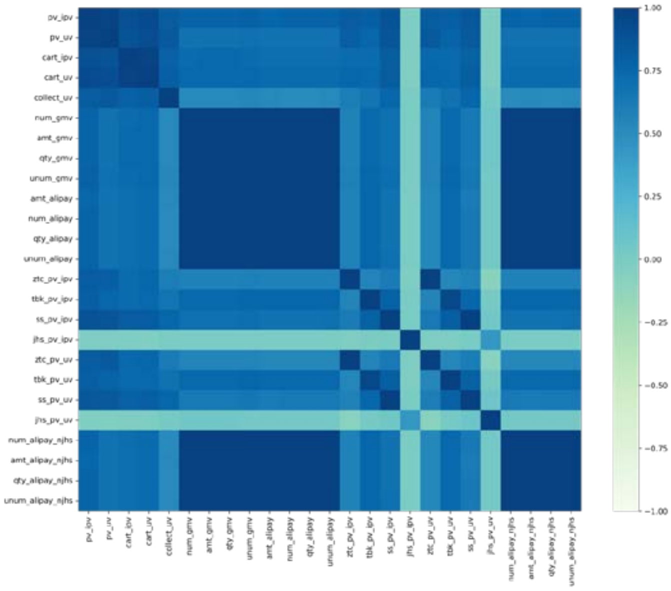

Fig. 4. The heat map of features correlation coefficient

To display the data characteristics of the demand more directly for this commodity, the data mentioned above were classified in terms of daily demand statistics **Fig. 5**. The horizontal axis of **Fig. 5** is the date, and the vertical axis indicates the demand, indicating obvious fluctuations in demand data. Since the dataset is derived from e-commerce data, some dates have extreme values due to merchant promotions. Individual extreme values can affect the model training, so the extreme values were removed from the data preprocessing. **Fig. 5** shows the results obtained by filling in the already excluded extremes and using the latter day's data.

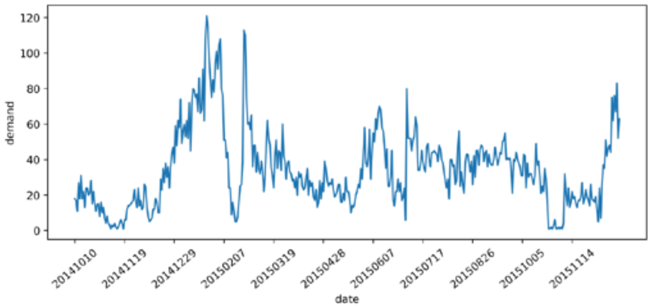

Fig. 5. Time series trend of daily demand from 2014 to 2015

Fig. 5 shows a certain seasonal pattern in the data. The demand is higher in autumn and winter than in summer, and it also shows certain random characteristics on a seasonal basis. However, data periodicity is not obvious, which is one of the difficulties in forecasting logistics and storage demand. The demand for a certain commodity will show a certain cyclicality in the short term. Still, in macro time, it does not necessarily maintain a stable trend due to the competition and substitution of similar products, thus creating a challenge for our forecast.

To make it easier to converge to the optimal solution correctly during the model training process and make the prediction results more accurate, in this study, we normalize the data by Min-Max normalization method compresses the values range to [0, 1], as in (20):

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{20}$$

Where x' is the normalized value, x is the original data value, max(x) and min(x) are the maximum and minimum values in the dataset, respectively.

In addition, to more comprehensively evaluate the performance of the model proposed in this study, we also use the dataset provided by the 2021 AliCloud Infrastructure Supply Chain Competition (Links to datasets). It contains the daily demand for cloud service resources from December 31, 2018 to June 7, 2021. The field is similar to the “2016 demand forecast & storage planning” dataset and is not repeated here. The forecast target is the virtual resource demand.

5.2 Predictive evaluation index

The following 4 indicators are canonically used to indicate forecast accuracy.

1)The root means square error, which is calculated as (21):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left|x_i - \tilde{x}_i\right|^2} \tag{21}$$

2)The absolute square error, which is calculated as (22)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|x_i - \tilde{x}_i\right| \tag{22}$$

3)The Mean Absolute Percentage Error is calculated as (23):

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\tilde{x}_i - x_i}{x_i}\right| \tag{23}$$

4)The symmetric Mean Absolute Percentage Error, which is calculated as (24):

$$SMAPE = \frac{100\%}{n}\sum_{i=1}^{n}\frac{\left|\tilde{x}_i - x_i\right|}{(\tilde{x}_i + x_i)/2} \tag{24}$$

Where n is the number of simples, x_i is the original data, and $\tilde{x}_i$ is the prediction data.

When the original value is zero, MAPE may be an error because the denominator is zero, which may happen in demand forecasting. Therefore, this paper selects MAE, RMSE, and SMAPE as metrics.

5.3 Experimental setting

The experimental environment of this study consisted of Windows10, 64-bit operating system, (Intel)G5400, CPU @3.70GHz, and RAM of 24 GB. We generated a CPU version of the PyTorch framework, which was used to build the neural network model. Meanwhile, Fbprophet and XGboost libraries were adopted for creating the Prophet and XGboost models, respectively.

In experiments, 80% and 20% of the time series data were used as training and testing sets, respectively. The multi-layer attention mechanism with the GRU model was then initialized. The batch size is 32, and the number of GRU hidden units in the model is 15. The length of the predicted sequence is 14. we use the back-propagation algorithm to train all models. During the training process, we use SGD and the Adam optimizer to minimize the mean squared error(MSE) between the original demand date and the prediction data.

$$L(\theta) = \left\|y' - y\right\|_2^2 \tag{25}$$

5.4 Experimental results

The prediction results of the MAGRU models are illustrated in **Fig. 6**.

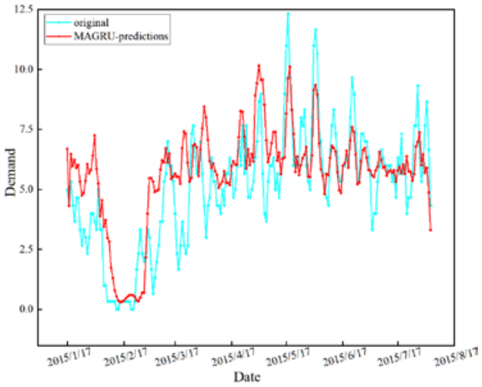

Fig. 6. Forecast results of MAGRU. The horizontal axis is the date for the forecast set, and the vertical axis is the real demand of a given forecast date.

The red line in **Fig. 6** indicates the prediction results of the model, and the blue line indicates the real demand data. It can be seen from the experimental results that the predicted value is consistent with the fluctuation of the sample value. MAGRU can better fit the trend of commodity demand. The difference between the predicted and real data is smaller in **Fig. 6**. It shows the prediction results of the model trained at a prediction length of 14 days. The figure shows that the difference between the true value and the predicted value is large on some dates. The predicted values are very close to the true values in the early stage, while the prediction performance starts to decrease as the prediction length increases. This is because the information recorded in the vector after encoder is limited and cannot store the information of many time steps, so the performance will drop significantly when the sequence length increases. Therefore, within each segment of the prediction sequence, the model can show better prediction performance at the starting position, but as the prediction task length increases, the prediction accuracy starts to decrease, this is shown in **Fig. 6**, where the predicted values for some dates differ significantly from the true values.

5.5 Experimental comparison

To further validate the performance of MAGRU, CNN [15], RNN [3], LSTM [4], GRU [5], and DARNN [19] have been experimentally selected as the baseline for comparison with MAGRU, and already three existing prediction models have been compared with MAGRU. The following **Fig. 7** shows the experimental results. From **Fig. 7**, we can find that the various types of prediction methods can predict the future trend, but there is a difference in the performance.

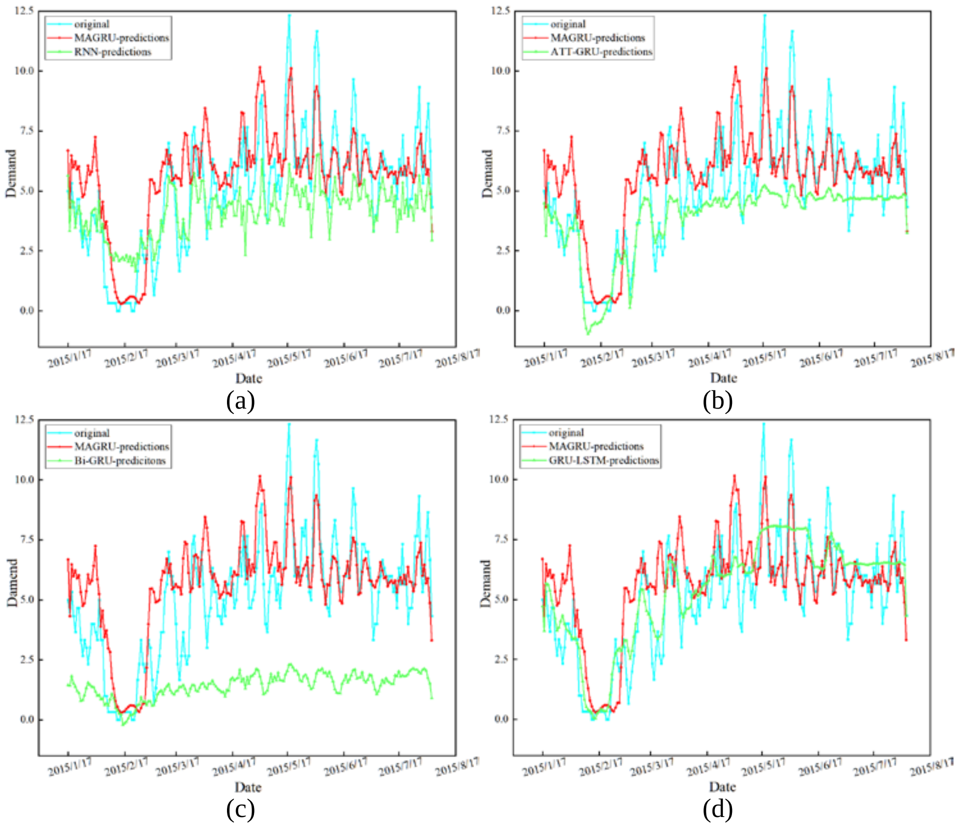

Fig. 7. MAGRU Forecast Curves compared with RNN Model, ATT-GRU Model, Bi-GRU Model, and GRU-LSTM Model

As shown in **Fig. 7(a)**, the predicted trend of the RNN model is close to the true trend, but the gap between the predicted and true values is relatively large. As shown in **Fig. 7(b)** and **Fig. 7(c)**, the prediction effect of the ATT-GRU model and the GRU-LSTM model is poorer, and the predicted trend is roughly similar to the true trend, but the predicted value is different from the true value, and the effect of the prediction model decreases as the length of the predicted time series increases. This indicates that the neural network can only have some accuracy in short time series prediction, but its performance is not as good as the deep learning model to predict future fluctuation changes. As shown in **Fig. 7(d)**, the GRU-LSTM model can predict the trend changes and fluctuations of future data in the early period. However, compared with MAGRU, the predicted and actual values show large differences with the increase of time series length.

The MAGRU model proposed in this study resulted in improved prediction accuracy. The difference between predicted and true values is small. The periodic trend of data is not obvious, but our model can better fit its fluctuation. The MAGRU model better solves the problem that the performance of the RNN model deterioration with time, and the statistical learning model cannot adapt well to the frequent fluctuation of data and can better provide the trend of future demand changes. The model's predictions are more accurate than both statistical methods of time series forecasting models and neural network time series forecasting methods.

We add a commodity demand dataset to validate the model generalization to provide a more comprehensive study of model performance. We selected another commodity demand data from the data provided by the Alibaba Tianchi Competition in 2016. This study labeled the dataset used above as Commodity1 and the newly added commodity dataset as Commodity2.

In this study, we used three different prediction series lengths. Specifically, it consists of 7, 14, and 21 days, the time step setting is 14 days. In order to verify the performance of the model at different series length, the following experiments were performed on the data to compare the prediction results of different models at different prediction lengths. We compared the model to a baseline model and three existing correlation models on the same dataset:

- CNN [15]: CNNs are deep learning algorithms that are particularly powerful for analysis of images, recognizing spatial hierarchies in data.
- RNN [3]: A type of neural network that excel in processing sequences of data for applications like language modeling and speech recognition.
- LSTM [4]: A special kind of RNN, capable of learning long-term dependencies in sequence data.
- GRU [5]: A variant of RNNs that aim to solve the vanishing gradient problem and are efficient for sequence prediction problems.
- DARNN [19]: The model utilizes a convolutional layer, three DAR modules, a parameterized rectified linear unit (PReLU) function, and global maximum pooling (GMP) to extract a depth representation from the input signal, and inputs it into the RUL prediction subnetwork to obtain predictions.
- GRU-LSTM [41]: An Attention-Based GRU-LSTM Model Using Gated Recursive Units (GRUs) with Long Short-Term Memory (LSTM).
- Bi-GRU [42]: An attention-based Bi-GRU network using time-series features for survival prediction.
- ATT-GRU [43]: A Short-Term Load Forecasting Model Using Attention-Based GRU to Pay More Attention to Key Variables and Improve Prediction Performance for Long Input Sequences

On datasets Commodity1 and Commodity2 we used batch sizes, epochs and learning rates of 32, 2000, 0.00001 and 32, 1000, 0.0001. The **Table 1** below shows the best performance of each approach under different parameter settings.

Table 1. Performance comparison for the demand prediction task (2016 dataset).
(the best results are in bold)

Predict length	**7days**			**14days**			**21days**		
	MAE	**RMSE**	**SMAPE**	**MAE**	**RMSE**	**SMAPE**	**MAE**	**RMSE**	**SMAPE**
Commodity1									
CNN	4.00	4.74	73.81%	3.31	4.15	67.94%	3.47	4.06	73.79%
RNN	2.47	3.00	56.00%	2.26	2.62	48.45%	2.08	2.41	45.82%
LSTM	4.89	5.24	126.34%	4.89	5.29	154.89%	4.28	4.75	142.14%
GRU	3.96	4.40	168.89%	3.83	4.30	99.55%	3.04	3.69	79.25%
DARNN	3.59	4.23	102.96%	4.23	4.59	114.41%	4.35	4.74	114.88%
GRU-LSTM	3.91	4.31	111.51%	4.19	4.64	114.93%	3.75	4.22	111.28%
Bi-GRU	3.45	4.02	90.97%	3.06	3.55	72.35%	2.57	3.09	60.86%
ATT-GRU	2.69	3.40	62.32%	2.68	3.18	58.47%	2.18	2.74	48.84%
MAGRU	**1.46**	**1.83**	**27.51%**	**1.76**	**2.02**	**34.41%**	**1.73**	**1.97**	**36.03%**
Commodity2									
CNN	8.64	10.19	56.77%	7.42	9.04	57.47%	7.01	8.39	66.41%
RNN	4.42	6.47	22.33%	7.00	8.89	32.45%	7.91	10.10	35.14%
LSTM	11.08	12.75	76.27%	10.83	12.21	62.32%	10.68	12.37	56.28%
GRU	9.60	11.03	59.83%	11.26	12.49	61.44%	11.64	13.16	59.85%
DARNN	8.71	9.86	97.76%	7.50	8.50	92.13%	9.00	10.10	100.02%
GRU-LSTM	**3.26**	4.58	**15.54%**	5.62	7.38	24.69%	6.52	8.73	27.39%
Bi-GRU	6.44	8.61	36.14%	7.82	9.55	38.28%	8.06	10.13	36.88%
ATT-GRU	9.46	10.47	58.04%	11.95	13.08	65.89%	12.75	14.16	67.17%
MAGRU	3.46	**4.29**	16.58%	**5.19**	**6.64**	**22.50%**	**6.06**	**7.94**	**25.10%**

Table 1 shows that for the Commodity1 dataset, when the prediction length is 7 days, the MAE, RMSE and SMAPE achieved the best results, which decreased by 40.89%, 39.00% and 50.87% compared with the best model. The MAE, RMSE and SMAPE were decreased by 22.12%, 22.90% and 28.98% for the prediction length of 14 days compared to the best model. The MAE, RMSE, and SMAPE were decreased by 16.83%, 18.26%, and 21.37% for the prediction length of 21 days compared to the best model.

On the other hand, the dataset we used does not have obvious periodicity, so the method based on RNN takes longer time dependence into account and has a better effect than the machine learning method.

For the Commodity 2 dataset, in most cases MAGRU achieved the best results in MAE, RMSE and RMSE, and the RMSE decreased by 6.33% to the best model. When the prediction length is 14, MAE, RMSE, and SMAPE decrease by 7.65%, 10.03%, and 8.87% compared with the best model. When the prediction length is 21, MAE, RMSE, and SMAPE decrease by 7.06%, 9.05%, and 8.36% compared with the next second model. The improvement for the Commodity2 dataset was smaller than that for Commodity1 because the data of Commodity2 were more distributed and had smaller values.

We also used the data provided by the AliCloud Infrastructure Supply Chain Competition 2021 for our experiments. Similarly, we selected two types of commodities in the dataset as experimental data, labeled unit1 and unit2, respectively, to examine the performance of the model under different length sequence tasks. On datasets Unit1 and Unit2 we used batch sizes, epochs and learning rates of 32, 1000, 0.0005 and 32, 2000, 0.0001. The experimental results are shown in **Table 2**.

Table 2 Performance comparison for the demand prediction task (2021 dataset).
(the best results are in bold)

Predict length	**7days**			**14days**			**21days**		
	MAE	**RMSE**	**SMAPE**	**MAE**	**RMSE**	**SMAPE**	**MAE**	**RMSE**	**SMAPE**
Unit1									
CNN	66.02	66.04	45.33%	68.24	68.30	46.49%	69.53	69.65	47.14%
RNN	96.65	96.65	60.57%	98.41	98.43	61.33%	100.14	100.18	62.07%
LSTM	123.67	123.92	84.95%	123.16	123.30	83.34%	124.17	124.26	83.27%
GRU	120.16	120.17	81.32%	120.20	120.21	80.40%	121.06	121.07	80.27%
DARNN	60.01	60.03	39.19%	60.82	60.94	39.59%	64.79	65.35	41.54%
GRU-LSTM	43.20	43.21	23.19%	44.86	44.91	23.96%	46.54	46.63	24.73%
Bi-GRU	83.75	83.77	50.46%	83.94	83.95	50.07%	84.86	84.88	50.23%
ATT-GRU	125.47	125.47	86.44%	127.05	127.06	86.94%	128.68	128.71	87.50%
MAGRU	**42.32**	**42.33**	**22.67%**	**43.37**	**43.39**	**23.07%**	**44.77**	**44.83**	**23.68%**
Unit2									
CNN	26.59	26.59	151.95%	27.09	27.11	153.45%	27.75	27.78	154.54%
RNN	15.82	16.78	74.06%	13.48	14.24	59.10%	12.97	13.53	54.93%
LSTM	26.76	26.95	155.60%	20.68	21.70	108.98%	18.56	19.55	92.17%
GRU	25.06	25.24	138.98%	21.78	22.13	112.46%	20.50	20.83	101.36%
DARNN	27.07	27.08	149.75%	27.80	27.82	150.72%	30.03	30.24	153.21%
GRU-LSTM	7.84	**7.86**	**29.10%**	7.81	7.82	29.00%	8.07	8.09	29.81%
Bi-GRU	19.99	20.19	97.40%	16.91	17.34	77.78%	14.80	15.43	65.28%
ATT-GRU	22.43	22.45	114.42%	21.92	21.94	110.47%	21.94	21.95	109.18%
MAGRU	**7.68**	8.30	29.35%	**6.40**	**6.90**	**23.74%**	**6.24**	**6.59**	**22.69%**

Table 2 shows the performance of the baseline on the 2021 dataset and the performance of MAGRU.

In Unit1, the MAGRU model exhibited significant improvements over the best-performing model. For prediction sequence lengths of 7, enhancements were observed as follows: MAE improved by 2.04%, RMSE by 2.04%, and SMAPE by 2.24%. Similarly, for sequence lengths of 14, the improvements were 3.32% in MAE, 3.38% in RMSE, and 3.71% in SMAPE. When applied to a prediction task with a sequence length of 21, the MAGRU model showed improvements of 3.80% in MAE, 3.86% in RMSE, and 4.25% in SMAPE compared to the best model. It shown that MAGRU is most effective for longer prediction tasks, and its long-term prediction advantage is more pronounced.

The traditional time series model, prophet its poor predictive performance because it models only temporal correlation and ignores the impact of external features on the prediction task. RNN, LSTM and GRU still have limitations for long sequence prediction tasks and has poor performance.

In Unit2, MAGRU achieved the best results for each prediction task length. The MAE improved by 2.04%, 18.05% and 22.68% from the best model, the RMSE improved by 11.76% and 18.54% from the best model when the prediction length is 14 and 21, and SMAPE improved by 18.14% and 23.88% from the best model when the prediction length is 14 and 21. Upon analyzing the performance of various models on the Unit2 dataset, it is observed that the variance in their predictive efficacy is significantly greater compared to other datasets. This discrepancy primarily stems from the substantial differences in sample feature values, data distribution, and noise levels in the Unit2 dataset relative to others. These variations significantly impact the predictive capabilities of the models, underscoring the importance of adapting to specific dataset characteristics. This observation highlights the necessity of a thorough understanding and analysis of dataset features when applying machine learning models, emphasizing their critical role in achieving accurate predictions.

MAGRU captures long-sequence dependencies using an attention mechanism combined with external features and GRU networks. For long-term serial prediction tasks, MAGRU is better than other models. The comparison shows that MAGRU has a high generalization capability while ensuring high accuracy.

This study proposed MAGRU model outperforms other models in forecasting, and the overall accuracy is better than that of a single model. Therefore, MAGRU is suitable for logistics demand forecasting in realistic scenarios and can replace a single model.

5.6 Ablation Study

To further evaluate the effectiveness of the individual components in the MAGRU, we conduct ablation studies on the Commodity1 dataset.

We have named the variants of MAGRU as follows:

MAGRU-NE: To validate the attention mechanism at the input, we remove it from the Encoder section.

MAGRU-ND: We simply remove the attention mechanism in the Decoder stage. We provide the results of the comparison of MA-GRU and its variants in **Fig. 8**.

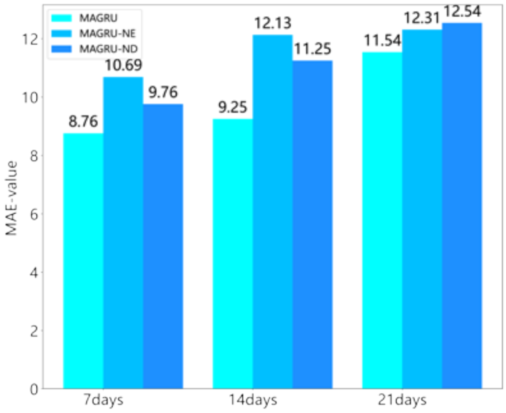

Fig. 8. Components analysis on the Commodity1 dataset

Under the same experimental settings as MAGRU, we perform experiments on its two variants, MAGRU-NE and MAGRU-ND, at multiple prediction length. The above **Fig. 8** shows the average prediction results of the model at different prediction length.

By comparison, we studied that the MAE of MA-GRU prediction results is significantly smaller than MAGRU-NE and MAGRU-ND. It can be considered that the attention mechanism in the encoder stage and the decoder stage is an important factor affecting the effect of the model, and adding multiple layers of attention mechanism to the model can effectively improve the model effect. The reason is that the attention mechanism can effectively capture key information about the entire sequence.

By the multi-layer attention mechanism, the model can extract information that is difficult to express in features. The GRU can receive the output of the attention mechanism and connect the information output from the previous unit. MAGRU extracts the time-domain correlation of the data, and the model obtains high-value information to improve prediction accuracy.

5.7 visual analysis

To gain a deeper understanding of the performance of MAGRU models in demand forecasting, this study analyzes in detail the loss dynamics of the model on the task of commodity demand forecasting. We recorded the model's loss variations during the training and testing phases and observed the loss values for seven consecutive days of forecasting performed after the model completed training. The following graphs (**Fig. 9** and **Fig. 10**) demonstrate the model's loss performance at various stages, providing visual evidence for assessing its forecasting ability.

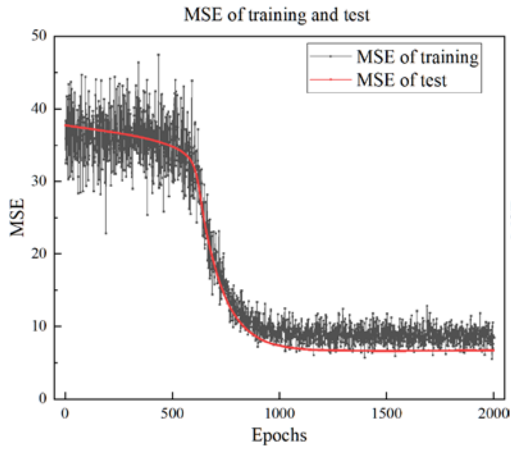

Fig. 9. MSE trends for training and testing

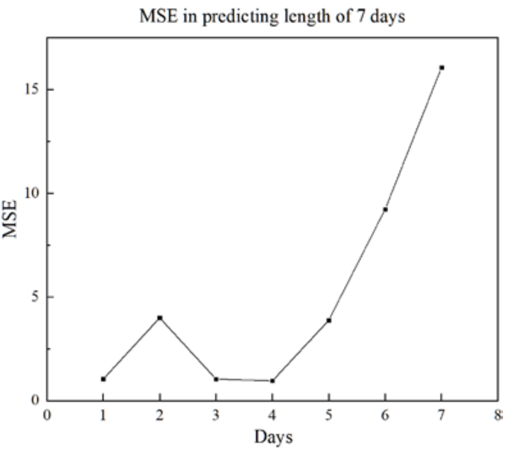

Fig. 10. MSE of the trained model predicting demand for the next 7 days

As shown in **Fig. 9**, as the number of iterations (Epochs) increases, the model's MSE on the training set and the loss on the test set show a significant decreasing trend and start to converge after 1000 epochs. Further, we performed predictions on the trained model for 7 consecutive days, and the loss values for each day are shown in **Fig. 10**. The results show that the model performs better in the early stages of short-term forecasting, but the value of the loss rises gradually as the length of the forecast increases. This trend suggests that although the model is able to capture certain time series features, its accuracy still decreases slightly when making long-term forecasts, which emphasizes the need to improve the model in future work in order to increase its stability and accuracy in long-term forecasting.

In summary, the GRU model has demonstrated its effectiveness on the task of commodity demand forecasting, especially in short-term forecasting. However, the fluctuating performance of the model on long-term forecasting suggests that we should explore hyperparameter tuning, introduce more sophisticated regularization techniques, or improve the loss function to mitigate overfitting and enhance the generalization ability of the model in subsequent studies.

6. Conclusion

In this study, a multi-layer attention mechanism with the GRU model is constructed using historical data of commodity demand in real warehouses. The attention mechanism can capture key information of the input sequence. We propose the embedding approach of timestamp and data integration that can further explore the factors implied in the demand data to improve the model's accuracy. The experimental results show that MAGRU takes precedence over state-of-the-art baseline models of the same type for long series prediction problems. In addition, the model has a simple structure and low training cost, which can provide an efficient solution for realistic warehouse demand forecasting.

This study has important implications for logistics operators to significantly reduce operating costs and enhance the customer experience. Integrating attention mechanisms and deep learning methods is a hot issue. However, it is important to recognize certain limitations of our study that deserve consideration in future work.

First, our model does not consider spatial factors, such as the geographic location and storage capacity of individual warehouses, which can have a significant impact on demand patterns. Second, we have not explored the coordination and interaction between multiple warehouses. In real logistics scenarios, inventory and replenishment strategies among multiple warehouses can affect demand patterns. Therefore, future research should focus on incorporating spatial factors and exploring the issue of multi-warehouse coordination to provide a more comprehensive solution for warehouse demand forecasting and to more accurately reflect complex logistics environments. Finally, although our experimental results show that MAGRU outperforms the baseline model in long series forecasting, the generalization performance of the model may vary across warehouses and demand patterns. Further validation and experiments are needed to determine its applicability in different environments.

In the future, we will explore the improvement of MAGRU in prediction involving spatial factors and apply it to a wider range of fields, such as sales prediction and traffic flow prediction.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China (71961028), the Key Research and Development Program of Gansu(22YF7GA171), the University Industry Support Program of Gansu(2023QB-115), the Innovation Fund for Science and Technology-based Small and Medium Enterprises(23CXGA0136) and the Scientific Research Project of the Lanzhou Science and Technology Program (2018-01-58).

References

[1] A. W. Bohannon, V. J. Lawhern, N. R. Waytowich, and R. V. Balan, "The Autoregressive Linear Mixture Model: A Time-Series Model for an Instantaneous Mixture of Network Processes," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4481-4496, Jul. 2020. Article (CrossRef Link)

[2] D. van der Meer, G. R. C. Mouli, G. M. E. Mouli, L. R. Elizondo, and P. Bauer, "Energy management system with PV power forecast to optimally charge EVs at the workplace," *IEEE transactions on industrial informatics*, vol. 14, no. 1, pp. 311-320, Jan. 2018. Article (CrossRef Link)

[3] Z. C. Lipton, D. Kale, and R. Wetzel, "Directly modeling missing data in sequences with rnns: Improved classification of clinical time series," in *Proc. of the 1st Machine Learning for Healthcare Conference* pp. 253-270, 2016. Article (CrossRef Link)

[4] Z. C. Lipton, D. C. Kale, and R. C. Wetzel, "Phenotyping of clinical time series with LSTM recurrent neural networks," *arXiv*, 2015. Article (CrossRef Link)

[5] A. Amin, L. Grunske, and A. Colman, "An approach to software reliability prediction based on time series modeling," *Journal of Systems and Software*, vol. 86, no. 7, pp. 1923-1932, Jul. 2013. Article (CrossRef SSSLink)

[6] Y. Yu, "A Study of Stock Market Predictability Based on Financial Time Series Models," *Mobile Information Systems*, Aug. 2022. Article (CrossRef Link)

[7] Y. Gao, M. T. Semiromi, and C. Merz, "Efficacy of statistical algorithms in imputing missing data of streamflow discharge imparted with variegated variances and seasonalities," *Environmental Earth Sciences*, vol. 82, no. 20, pp. 476, Sep. 2023. Article (CrossRef Link)

[8] Z. Xia, S. Xue, L. Wu, J. Sun, Y. Chen, and R. Zhang, "ForeXGBoost: passenger car sales prediction based on XGBoost," *Distributed and Parallel Databases*, vol. 38, no. 3, pp. 713-738, May. 2020. Article (CrossRef Link)

[9] J. Henzel, Ł. Wróbel, M. Fice, and M. Sikora, "Energy consumption forecasting for the digital-twin model of the building," *Energies*, vol. 15, no. 12, pp. 4318, Jun. 2022. Article (CrossRef Link)

[10] D. N. Wang, L. Lang, and D. Zhao, "Corporate finance risk prediction based on LightGBM," *Information Sciences*, vol. 602, pp. 259-268, Jul. 2022. Article (CrossRef Link)

[11] H. C. Du, L. Lv, A. Guo, and H. L. Wang, "AutoEncoder and LightGBM for Credit Card Fraud Detection Problems," *Symmetry*, vol. 15, no. 4, pp. 870, Apr. 2023. Article (CrossRef Link)

[12] J. Ke, H. Zheng, H. Yang, and X. M. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 591-608, Dec. 2017. Article (CrossRef Link)

[13] Z. Masood, R. Gantassi, and Y. Choi, "A Multi-Step Time-Series Clustering-Based Seq2Seq LSTM Learning for a Single Household Electricity Load Forecasting," *Energies*, vol. 15, no. 7, pp. 2623, Apr. 2022. Article (CrossRef Link)

[14] M. Liu, A. Zeng, Q. Lai, and Q. Xu, "T-WaveNet: Tree-Structured Wavelet Neural Network for Sensor-Based Time Series Analysis," *arXiv*, Dec. 2020. Article (CrossRef Link)

[15] T. Li, Y. Zhang, and T. Wang, "SRPM–CNN: a combined model based on slide relative position matrix and CNN for time series classification," *Complex & Intelligent Systems*, vol. 7, no. 3, pp. 1619-1631, Feb. 2021. Article (CrossRef Link)

[16] S. Mehtab, and J. Sen, "Analysis and forecasting of financial time series using CNN and LSTM-based deep learning models," *Advances in Distributed Computing and Machine Learning*, pp. 405-423, Nov. 2022. Article (CrossRef Link)

[17] J. Park, M. R. Babaei, S. A. Munoz, A. N. Venkat, and J. D. Hedengren, "Simultaneous multistep transformer architecture for model predictive control," *Computers & Chemical Engineering*, vol. 178, Oct. 2023. Article (CrossRef Link)

[18] X. Li, Z. Liu and Z. Huang, "Attention-Based Radar PRI Modulation Recognition With Recurrent Neural Networks," *IEEE Access*, vol. 8, pp. 57426-57436, Mar. 2020. Article (CrossRef Link)

[19] F. Zeng, Y. Li, Y. Jiang, and G. Song, "A deep attention residual neural network-based remaining useful life prediction of machinery," *Measurement*, vol. 181, pp. 109642, Aug. 2021. Article (CrossRef Link)

[20] N. Zucchet, S. Kobayashi, Y. Akram, J. von Oswald, M. Larcher, A. Steger, and J. Sacramento, "Gated recurrent neural networks discover attention," *arXiv*, Sep. 2023. Article (CrossRef Link)

[21] W. Ye, H. Kuang, J. Li, X. Lai, and H. Qu, "A parking occupancy prediction method incorporating time series decomposition and temporal pattern attention mechanism," *IET Intelligent Transport Systems*, vol. 18(1), pp. 58-71, Oct. 2023. Article (CrossRef Link)

[22] C. Liu, Y. Liu, Y. Yan, and J. Wang, "An intrusion detection model with hierarchical attention mechanism," *IEEE Access*, vol. 8, pp. 67542-67554, Mar. 2020. Article (CrossRef Link)

[23] Y. Wu, and W. Li, "Aspect-level sentiment classification based on location and hybrid multi attention mechanism," *Appl Intell*, vol. 52, no. 10, pp. 11539-11554, Jan. 2022. Article (CrossRef Link)

[24] S. Du, T. Li, Y. Yang, and S. J. Horng, "Multivariate time series forecasting via attention-based encoder–decoder framework," *Neurocomputing*, vol. 388, pp. 269-279, May. 2020. Article (CrossRef Link)

[25] P. Kavianpour, M. Kavianpour, E. Jahani, and A. Ramezani, "A cnn-bilstm model with attention mechanism for earthquake prediction," *The Journal of Supercomputing*, vol. 79, no. 17, pp. 19194-19226, May. 2023. Article (CrossRef Link)

[26] Y. Cao, C. Liu, Z. Huang, Y. Sheng, and Y. Ju, "Skeleton-based action recognition with temporal action graph and temporal adaptive graph convolution structure," *Multimedia Tools and Applications*, vol. 80, no. 19, pp. 29139-29162, Jun. 2021. Article (CrossRef Link)

[27] B. Diallo, J. Hu, T. Li, G. A. Khan, X. Liang, and Y. Zhao, "Deep embedding clustering based on contractive autoencoder," *Neurocomputing*, vol. 433, pp. 96-107, Apr. 2021. Article (CrossRef Link)

[28] G. A. Khan, J. Hu, T. Li, B. Diallo, and H. Wang, "Multi-view Clustering for Multiple Manifold Learning via Concept Factorization," *Digital Signal Processing*, vol. 140, Aug. 2023. Article (CrossRef Link)

[29] B. Diallo, J. Hu, T. Li, G. A. Khan, X. Liang, and H. Wang, "Auto-attention Mechanism for Multi-view Deep Embedding Clustering," *Pattern Recognition*, vol. 143, Nov. 2023. Article (CrossRef Link)

[30] H. Liao, J. Hu, T. Li, S. Du, B. Peng, "Deep linear graph attention model for attributed graph clustering," *Knowledge-Based Systems*, vol. 246, Jun. 2022. Article (CrossRef Link)

[31] J. T. Fite, G. D. Taylor, J. S. Usher, J. R. English, and J. N. Roberts, "Forecasting freight demand using economic indices," *International Journal of Physical Distribution & Logistics Management*, vol. 32(4), pp. 299-308, May. 2002. Article (CrossRef Link)

[32] J. J. Bergman, J. S. Noble, R. G. McGarvey, and R. L. Bradley, "A Bayesian approach to demand forecasting for new equipment programs," *Robotics and Computer-Integrated Manufacturing*, vol. 47, pp. 17-21, Oct. 2017. Article (CrossRef Link)

[33] K. Bhuwalka, E. Choi, E. A. Moore, R. Roth, R. E. Kirchain, and E. A. Olivetti, "A hierarchical Bayesian regression model that reduces uncertainty in material demand predictions," *Journal of Industrial Ecology*, vol. 27, no. 1, pp. 43-55, Feb. 2023. Article (CrossRef Link)

[34] T. Yang, N. Yang, and C. Zhu, "A forecasting model for feed grain demand based on combined dynamic model," *Computational Intelligence and Neuroscience*, Sep. 2016. Article (CrossRef Link)

[35] J. Huber, A. Gossmann, and H. Stuckenschmidt, "Cluster-based hierarchical demand forecasting for perishable goods," *Expert systems with applications*, vol. 76, pp. 140-151, Jun. 2017. Article (CrossRef Link)

[36] Q. Q. He, C. Wu, and Y. W. Si, "LSTM with Particle Swam Optimization for Sales Forecasting," *Electronic Commerce Research and Applications*, vol. 51, pp. 101118, 2022. Article (CrossRef Link)

[37] Y. Li, Y. Yang, K. Zhu, and J. Zhang, "Clothing Sale Forecasting by a Composite GRU–Prophet Model with an Attention Mechanism," *IEEE Transactions on Industrial Informatics*, vol. 17 no. 12, pp. 8335-8344, Feb. 2021. Article (CrossRef Link)

[38] H. Guo, C. Guo, B. Xu, Y. Xia, and F. Sun, "MLP neural network-based regional logistics demand prediction," *Neural Computing and Applications*, vol. 33 no. 9, pp. 3939-3952, 2021. Article (CrossRef Link)

[39] W. Cai, Y. Song, and Z. Wei, "Multimodal data guided spatial feature fusion and grouping strategy for E-commerce commodity demand forecasting," *Mobile Information Systems*, vol. 2021, pp. 1-14, Jun. 2021. Article (CrossRef Link)

[40] Q. Gao, H. Xu, and A. Li, "The analysis of commodity demand predication in supply chain network based on particle swarm optimization algorithm," *Journal of Computational and Applied Mathematics*, vol. 400, pp. 113760, Jan. 2022. Article (CrossRef Link)

[41] H. S. Munir, S. Ren, M. Mustafa, C. N. Siddique, and S. Qayyum, "Attention based GRU-LSTM for software defect prediction," *Plos one*, vol. 16, no. 3, pp. e0247444, Mar. 2021. Article (CrossRef Link)

[42] Z. Yang, Y. Tian, T. Zhou, Y. Zhu, P. Zhang, J. Chen, and J. Li, "Time-series deep survival prediction for hemodialysis patients using an attention-based Bi-GRU network," *Computer Methods and Programs in Biomedicine*, vol. 212, pp. 106458, Nov. 2021. Article (CrossRef Link)

[43] S. Jung, J. Moon, S. Park, and E. Hwang, "An attention-based multilayer GRU model for multistep-ahead short-term load forecasting," *Sensors*, vol. 21, no. 5, pp. 1639, Feb. 2021. Article (CrossRef Link)

Ran Tian, Ph. D., associate professor. His current research interests include deep reinforcement learning, cloud service platform, supply chain intelligence decision making, etc.

Bo Wang, M.E. candidate. is a Master's degree candidate at the School of Computer Science and Technology, Northwest Normal University. His current research interests include attention mechanism, recurrent neural networks, etc.

Chu Wang, M.E. His current research interests include spatial-temporal data mining, graph neural network, etc.