Expert Systems With Applications 238 (2024) 122160

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Review

# Multi-Time Scale Aware Host Task Preferred Learning for WEEE return prediction

Yujiang Wu [a], Min Gao [a,*], Ruiqi Liu [a], Jie Zeng [a], Quanwu Zhao [b], Jinyong Gao [c], Jia Zhang [a]

[a] School of Big Data and Software Engineering, Chongqing University, Chongqing 401331, China
[b] School of Economics and Business Administration, Chongqing University, Chongqing 400044, China
[c] Aibo Green Reverse Supply Chain Co., Ltd., Shenzhen 518000, China

ARTICLE INFO

*Keywords:*
Reverse logistics return prediction
Waste electrical and electronic
Multi-task learning
Preferential multi-task learning

ABSTRACT

Recently, with the improvement of per-capita income, the number of waste electronic and electrical equipment (WEEE) has increased significantly. The WEEE return prediction is an essential part of reverse logistics (RL) due to its helpfulness in decision-making. The traditional prediction methods usually learn from the historical data of merely a single type of WEEE. However, the prediction tasks of different types of WEEE are relevant to some extent, the lack of considering their relationships in prediction leads to sub-optimal performance. To this end, we propose a multi-task learning model, Multi-Time Scale Aware Host Task Preferred Learning model (MAHOP), to predict return volume by learning from multiple types of WEEE. The work is non-trivial due to the challenges: (1) Collaborative extraction of multi-time scale features and multi-task common features from different types of WEEE, (2) fair prediction for every type of WEEE, and (3) Proper usage of common different time scale features. To tackle these challenges, we first construct a multi-task learning framework with different towers to learn three common time-scale features from the time series data of all types of WEEE. Besides, we propose a polling host-task learning strategy and a host-preferred loss. Moreover, we design a more sharing tower to make the model not overly dependent on some specific time series features. We apply MAHOP to a WEEE recycling enterprise and conduct extensive experiments to demonstrate that MAHOP outperforms baseline models, with improved performance and acceptable hyper-parameter sensitivity. To be more specific, the average prediction error for the fridge, air conditioner, washing machine, and television is about 8% lower than that of the suboptimal model.

## 1. Introduction

Nowadays, as the use of electronic equipment becomes more and more prevailing, the problems and challenges of how to deal with the waste of electronic and electrical equipment (WEEE) begin into public view. Considering that WEEE is a non-homogeneous and complex in terms of materials and components and many of the materials are highly toxic, we need the methods to process them appropriately. The traditional recycling methods always have high time cost and expense costs, and also, they are not fully optimized in various stages of the industry chain, which may lead to some social issues, such as inconvenience for people who have to go to designated recycling points to dispose of their waste. As a result, the reverse logistics (RL) emerges as the times require (Lehtinen & Poikela, 2006). As the central part of the research in WEEE RL domains (Islam & Huda, 2018), decision-making highly depends on the prediction of return volume.

The RL return prediction belongs to time series prediction. Many researchers have made a lot of breakthroughs in this domain. There are two main types of solutions in this field, traditional models, which contain statistical models as well as machine learning models, and deep learning methods.

The former includes regression models based on various types of time series data features (Kelle & Silver, 1989), such as Bayesian-based (Toktay et al., 2000) and Autoregressive Integrated Moving Average (ARIMA) (Wang et al., 2012) based models. Although these methods performed well in some aspects, they are not good at processing nonlinear data (Qian & Gao, 2017; Zhang et al., 2023). The latter use deep neural networks, such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997), DeepAR (The combination of RNN and AR) (Salinas et al., 2020), and Temporal Convolutional Networks (TCN) (Bai et al., 2018) to

* Corresponding author.
*E-mail addresses:* wuyujiang@stu.cqu.edu.cn (Y. Wu), gaomin@cqu.edu.cn (M. Gao), liuruiqi@stu.cqu.edu.cn (R. Liu), zengjie@cqu.edu.cn (J. Zeng), zhaoquanwumx@cqu.edu.cn (Q. Zhao), gjy@boolv.com (J. Gao), jiazhang@cqu.edu.cn (J. Zhang).

https://doi.org/10.1016/j.eswa.2023.122160
Received 19 March 2023; Received in revised form 23 September 2023; Accepted 12 October 2023
Available online 17 October 2023
0957-4174/

capture the nonlinear data features. Moreover, Zhang et al. (2023) first considered the multi-time scale features of WEEE in the RL prediction field and put forward a multi-timescale attention network (MULAN).

These methods only use the historical data of each single WEEE to predict its future recovery without considering the correlation of different WEEE prediction tasks. However, the tasks of different types of WEEE prediction highly correlate with each other; when dealing with related tasks, the models learning multiple tasks collaboratively are superior to those only learning for a single task (Standley et al., 2020; Zhang & Yang, 2021) because the usage of relative tasks will learn more precise common features by auxiliary data. Moreover, more types of electronic and electrical equipment are included in WEEE (Directive, 2012) that provide more related auxiliary data. To this end, we propose a multi-task learning model, Multi-time Scale Aware Host Task Preferred Learning (MAHOP), for WEEE RL prediction. Our work is non-trivial due to the following challenges: (1) How to extract multi-time scale features and multi-task common features collaboratively? Most of the previous multi-task learning networks do not take time scale features into consideration and are not suitable for time series data prediction. (2) How to achieve fair prediction for all tasks? Existing multi-task learning models focus more on efficiency and overall performance, usually leading to lower accuracy of a few tasks. (3) How to use the multiple time scale features probably to avoid the host task paying more attention to some special time series features?

For the first challenge specifically, we propose a framework to learn common multi-time scale features, a.k.a. monthly periods, annual trends, and recent closeness, from time series data of different WEEE. In the feature extraction phase of the framework, we design three tower modules to deal with three time-scale features. In the prediction phase of the framework, each task selectively extracts information from the different towers because the time series of different WEEE may have variant dependencies among different time features.

For the second challenge, we design a polling host-task learning strategy and a host-preferred Loss to achieve a fair prediction for all types of WEEE. We define the host task as the target task and the guest task as the task to assist in predicting the host task. It ensures that the host task keeps the highest speed to learn, and at the same time, all the guest tasks can find a balance among each other when they are trained together. Every task can be a host task once and a guest task many times (when another task is a host task).

For the third challenge, we find that the model is sensitive to the hyper-parameters for the balance of multiple time series features; thus, we introduce one more sharing tower containing all time series data, which can make the model not overly depend on some specific time series features. Our improved MAHOP can decrease sensitivity and increase the generalization capability. To distinguish between the proposed model with and without the additional sharing tower, we name the model without the tower MAHOP/st.

The MAHOP model can fully use common multi-time scale features and focus on every host task. Although there have already been some asymmetry multi-task learning models such as deep asymmetric multitask feature learning (Lee et al., 2018) and temporal probabilistic asymmetric multi-task learning (Tuan et al., 2019), they still make all the tasks trained at the same time, rather than the principles of our model, training different tasks separately and exchanging the role of each task every time. We show the difference in the characteristics among STL (single-task learning, i.e., MULAN), MTL (multi-task learning, i.e., Multi-gate Mixture-of-Experts (MMOE)), normal asymmetry MTL, and our MAHOP, in Fig. 1 (two tasks learning for instance).

To summarize, the contributions of our work are as follows:

(1) **Our model first makes use of the relationships among various WEEE historical data for the prediction task.** We design a multi-time scale aware multi-task learning framework for the WEEE prediction.
(2) **Our model extracts multi-time scale features and multi-task common features collaboratively.** We create three towers to deal with the different time sequences of data, and different WEEE prediction tasks can extract these features selectively.
(3) **We make the first exploration of distinguishing host and guest tasks in the learning for WEEE RL prediction domain.** We design a Host-preferred Loss to ensure that the host task keeps the highest speed to learn and that all the guest tasks can find a balance when they are trained together.
(4) **We design a sharing tower to help the model avoid overly depending on some specific time series features.** The sharing tower can leverage full sequence data to get common time series features that can improve the model's generalization.

## 2. Related works

### 2.1. Traditional methods in the RL prediction field

Plenty of methods have been put forward to improve the effectiveness of the prediction model in the RL field. Researchers first solved this problem by applying various statistical and machine learning methods. Kelle and Silver (1989) first put forward the regression equations to predict. After that, Toktay et al. (2000) developed a Bayesian-based forecasting model, which depended on the assumption that RL recovery models conformed to a binomial probability distribution. To further increase the precision of the prediction model, Yang and Williams (2009) created regression equation-based forecasting models. The most recent machine learning method to deal with this was called Autoregressive Integrated Moving Average (ARIMA) (Wang et al., 2012). Most of the traditional methods could solve some prediction problems very well, but Siami-Namini et al. (2018) got a conclusion through the contrast experiment that their effect did not compare with deep learning models.

### 2.2. Deep learning models for time series prediction

In addition to the traditional models, many deep learning methods also were put into use in the time series analysis field. The earliest solutions are recurrent neural networks (RNNs) and Long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997), introducing the gate mechanism based on RNNs. Apart from LSTM, Salinas et al. (2020) did another improvement to RNN, combining RNN with AR, which was called DeepAR. Although the cycle structures always perform well in the time series field, some researchers put forward the other method based on the convolutional neural network, which is called temporal convolutional networks (TCN) (Bai et al., 2018). Recently, some people creatively employed the temporal attention mechanisms to LSTM for better effect on time period prediction tasks (Hu & Zheng, 2020). In addition, many studies also used transformer structures for related predictions of time series (Shen & Wang, 2022; Zhou et al., 2021), which could efficiently handles extreme long input sequences. Moreover, there was research that designs a multi-variable time series prediction model that incorporates transformer structures, Crossformer (Zhang & Yan, 2022), which could handle the dependency among different variables and had important implications for our task of predicting multiple categories of WEEE simultaneously. The latest research achievement in the RL field was a multi-timescale attention network (Zhang et al., 2023), which first considered the different timescale features in the specific RL field.

### 2.3. MTL methods

However, all of the previous methods did not make use of the similarity of the multi-time scale features among different electronic equipment, which contradicted the spirit of the new standard of WEEE (Directive, 2012).

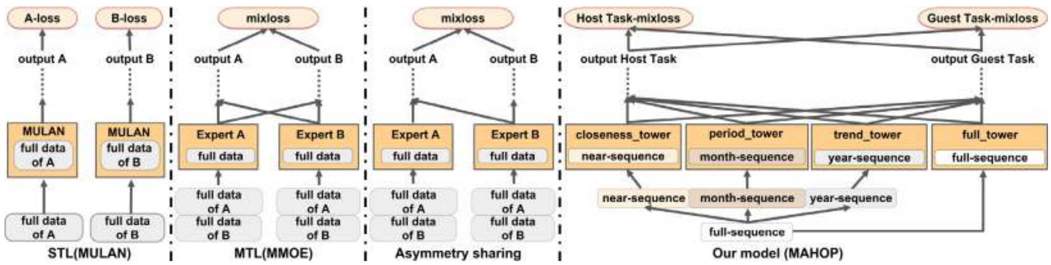


**Fig. 1.** The difference among structures of classical prediction models and our MAHOP (its input consists of six parts. Horizontally, it is composed of closeness-sequence, month-sequence, and year-sequence; while vertically, it is composed of two different kinds of WEEE).

To study the similarity features among different tasks, a lot of multi-task learning (MTL) structures can be applied to deal with it. The earliest method was called hard parameter sharing (Caruana, 1993), followed by the Deep Mixture of Experts (Eigen et al., 2013), a deeper network as an improvement of the Mixture of Experts (MOE) (Jacobs et al., 1991). The latest work to refine MOE, Multi-gate Mixture-of-Experts (MMOE), was made by Ma et al. (2018), which introduced the multi-gate mechanism. After that, Tang et al. (2020) put forward Customized Gate Control (CGC) as well as Progressive Layered Extraction (PLE) to make further improvements. Some researchers also migrated the methods commonly used in the field of recommendation systems and computer vision to the field of time series prediction. Mahmoud et al. (2020) proposed the MTL model which fused Gated Recurrent Units (GRU) and Convolutional Neural Networks (CNN). After that, Warrier et al. (2022) put forward the MTL methods containing LSTM networks. In addition to the normal MTL models, many researchers also tried to propose some asymmetry MTL models, such as Deep Asymmetric Multitask Feature Learning (Deep-AMTFL) (Lee et al., 2018) and Temporal Probabilistic Asymmetric Multi-Task Learning (TP-AMTL) (Tuan et al., 2019).

Although there are so many MTL models to deal with the multi-task learning problems, none of them can directly be applied to the prediction task of different types of electronic equipment in the RL field due to the two reasons as follows. For one thing, most of the previous MTL networks (Ma et al., 2018; Tang et al., 2020) concentrate on the Computational Vision and Recommendation System field, and they cannot be directly put into use in the time series. It is because that they do not make use of the time features in the RL prediction field. Even for MTL models that apply to the field of time series, they have not taken into account the multi-time scale characteristics of RL sequences. For another thing, the final purpose of almost all the previous MTL models is reducing the time of training, which means that they place more emphasis on overall performance and efficiency, rather than improving the prediction accuracy of each individual task. With such a design purpose, different WEEE predictions in multi-task learning will inevitably lead to the problem of gradient conflict between tasks, making it impossible to simultaneously improve the prediction accuracy of various WEEE, which is not in line with our ultimate goal in the RL prediction field. As a result, we will propose our new model in this paper to solve these problems.

### 2.4. Overall novelty

Compared to the aforementioned methods, our approach has the following innovative features to address the challenges faced by those methods. (1) We make use of multi-time scale features when doing the multi-task learning and create three towers to deal with the different time sequences of data. (2) We propose the Host-preferred Loss to distinguish host and guest tasks in order to increase the effect of all of the prediction tasks. (3) We design a sharing tower to help the model better evenly depend on each specific time series feature.

## 3. Preliminaries

### 3.1. Method applicability

MTL has always been lacking in effective application in the field of time series. The main reason is that the core of MTL is to process many relative tasks at the same time and each task has different outputs to achieve different purposes. However, for most work in the time series field, we always expect to predict one specific indicator in the future. And even if many types of data need to be forecast, the most popular and common way is to predict them separately or to package all indicators into a high-dimensional vector, instead of considering the prediction of each indicator as one task of MTL.

For the specific task in this paper, predicting the data of different WEEE, however, is very suitable for the idea of MTL. We find that for different electronic equipment, their data series are similar to each other but not totally the same, so making them train together will be beneficial for each prediction task. Moreover, we have different indicators to learn, the return of each WEEE in the future, and at the same time, the indicators have relationships with each other, which means we cannot consider them as completely independent tasks or package them into a vector.

As a result, we do the work to adjust classical MTL methods to some models that can fit our tasks well in this paper. And the most important problem to solve is how to make sure that our models behave better for each single WEEE prediction task, which is introduced in detail in Section 4.

### 3.2. Notation definition

We define the notations as follows to describe the problems as well as our models more clearly (For clarity and conciseness, we will use HTK to represent the host task, and GTK to represent the guest task in the following formulations):

- **(X)-Y-sequence:** We stipulate $X \in \{FD, AC, WM, TV, FAW, FAWT\}$ and $Y \in \{year, month, near, full\}$, which represent the time series data of fridge, air conditioner, washing machine, television, first 3 WEEE, and all WEEE in trend window, period window, closeness window and the conjunction of these 3 windows (Zhang et al., 2023) respectively.
- **X-loss:** We stipulate $X \in \{FD, AC, WM, TV, HTK, GTK\}$, which represents the loss of each WEEE prediction task (including the prediction task of the host task and guest tasks) before calculating them together to get the final mixloss.
- **X-mixloss:** We stipulate $X \in \{FD, AC, WM, TV, HTK\}$, which represents the mixloss of each WEEE prediction task (including the prediction task of the host task). Each time we want to concentrate on the task of one specific X, we will calculate X-loss and the loss of the rest guest tasks together, to get X-mixloss.

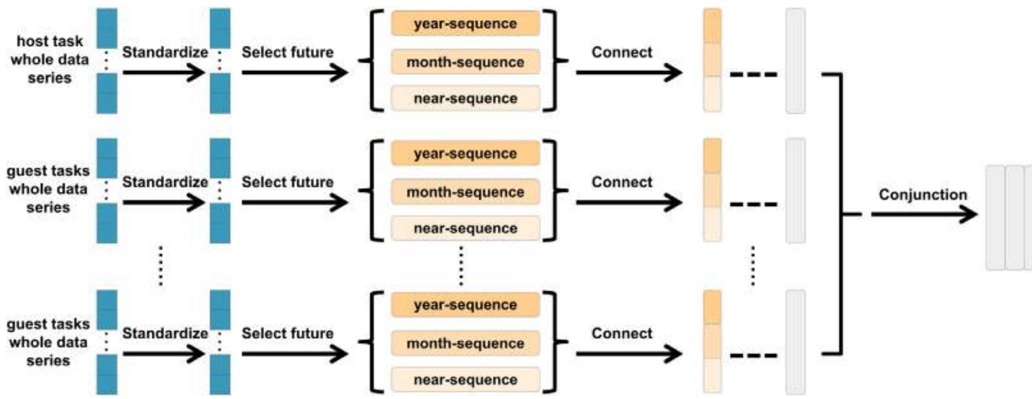


**Fig. 2.** The pre-processing of original data.

### 3.3. Guest tasks selecting and data pre-processing

According to Fig. 5 in Section 5, we can figure out that the time series of FD, AC, and WM all show an upward and then downward trend, while the time series of Tv shows an upward then steady trend, which means the multi-time scale features of FD, AC, WM are much more similar to each other compared with TV. On the other hand, the first three kinds of WEEE data still have some similarities with TV, which means that the TV return prediction task can learn something when it is training with the rest three tasks together. As a result, when we choose TV to be the host task, the guest tasks will be the rest of WEEE. and when we choose another WEEE to be the host task, the guest tasks will be the rest of WEEE except TV.

Before we input the data chosen from the datasets, we are supposed to do some pre-processing of the original data. Initially, we will standardize the data of each WEEE separately. Secondly, we will select the year-sequence, month-sequence as well as near-sequence of the host task from the whole data series, and then we connect them to the full-sequence. After that, we will repeat the same process for the rest guest tasks. Eventually, we will conjunction all full-series into a high-dimensional vector. The complete steps of the data pre-processing introduced previously are shown in Fig. 2.

## 4. Methodology

In this section, we first introduce the detailed structures of our two models, original MAHOP/st and improved MAHOP, in 4.1 and 4.2. After that, we explained the principle and implementation of the polling host-task learning strategy and the Host-preferred Loss algorithm in 4.3. Eventually, we use 4.4 to show the specific algorithm process of MAHOP.

### 4.1. Multi-time scale aware host task preferred learning model without sharing-tower

#### 4.1.1. Overview of the framework

The whole structure of MAHOP/st is shown in Fig. 3, and some details of the model are also in it.

We can find that our model MAHOP/st uses some ideas and structures of the Multi-gate Mixture-of-Experts model in general. The input of the model is a high-dimensional vector, and different dimensions represent the number of different WEEE. Firstly, the input will be sliced into 3 time series, near-sequence, month-sequence, and year-sequence. Then they will transfer to closeness_tower, period_tower, and trend_tower separately. After that, the outputs of each tower are fused through different sockets, and the fused results are decoded by their respective decoders. Eventually, we will get the output and corresponding loss from each decoder, and calculate all the losses to the final mixloss.

As we can see from Fig. 3, considering the difference between the time series data of TV and the other 3 WEEE, the model to predict TV data is slightly different from other WEEE prediction models. For one thing, the input of the TV prediction model is FAWT-full-sequence, while others are FAW-full-sequence. For another, when we choose TV to be the host task, we need 4 sockets and 4 decoders to calculate the loss of all WEEE and fuse them to the TV-mixloss. However, for the other 3 WEEE, they only need 3 sockets and 3 decoders and TV-loss is not included in the final mixloss.

#### 4.1.2. Details of tower

In the tower layer, the input is the form of a high-dimensional vector, combined with the time series of the host task and guest tasks, so the model learns the mutual characteristics of all WEEE. The different towers only represent that the model will learn from three different features in this layer: annual synchronization, monthly synchronization, and recent trends.

Different towers have similar networks, but they are not identical. It is because the structure of each tower is an LSTM network as shown in Fig. 3, while the LSTM network of different towers has various amounts of repeating modules, which is due to the length of respective input data. In other words, the number of repeating modules in closeness_tower is consistent with the length of near-sequence, and the number of repeating modules in period_tower is consistent with the length of month-sequence, and the same goes for trend_tower. The calculation process of LSTM is shown below:

$$\begin{bmatrix} f_j \\ i_j \\ o_j \end{bmatrix} = \sigma\left(\begin{bmatrix} W_f \\ W_i \\ W_o \end{bmatrix}\left[h_{j-1}, x_j\right] + \begin{bmatrix} b_f \\ b_i \\ b_o \end{bmatrix}\right), \tag{1}$$

$$\tilde{C}_j = \tanh\left(W_C\left[h_{j-1}, x_j\right] + b_C\right), \tag{2}$$

$$C_j = f_j * C_{j-1} + i_j * \tilde{C}_j, \tag{3}$$

$$h_j = o_j * \tanh\left(C_j\right). \tag{4}$$

As we can see, the $x_j$ represents all WEEE return data of each day. Apart from $h_0$, which is a vector randomly initialized at the beginning, other variables are only intermediate variables. Since the dimensions of each vector passed into the socket must be the same, we have to make sure that the outputs of all towers have the same shape. As a result, we only choose the last $h_j$, which is also known as $y_0$, as the output, while we discard the other $h_j$.

#### 4.1.3. Structure of socket

Different WEEE should share the parameter learning of all towers so that the three towers can fully represent the characteristics of annual synchronization, monthly synchronization, and recent trends. However,

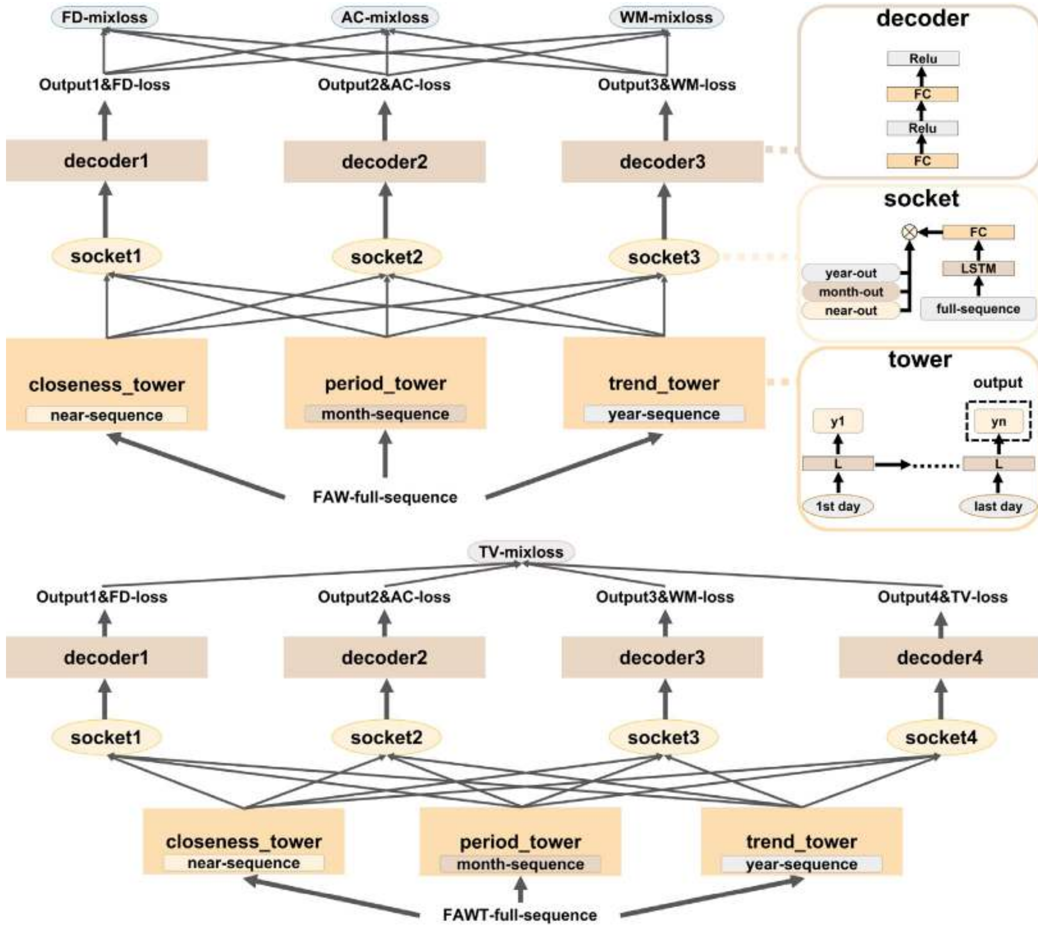


**Fig. 3.** The overall architecture of MAHOP/st.

through sockets, the network of different WEEE can control the attention weight of previous features (annual synchronization, monthly synchronization, and recent trend) separately, mainly because different WEEE have different dependencies on each feature.

Firstly, we input the full-sequence to the LSTM, and then we use the Full Connect Neural Network (FC) to change the corresponding dimension of the LSTM output into 3, which is the same as the number of towers. And then we fusion the output of three towers by multiplying them with the output of the previous FC. The whole process is shown below:

$$x_{LSTM\text{-}out} = LSTM\left(x_{full\_sequence}\right), \tag{5}$$

$$x_{FC\text{-}out} = W_{FC}x_{LSTM\text{-}out} + b_{FC}, \tag{6}$$

$$x_{socket\text{-}out} = x_{FC\text{-}out}\begin{bmatrix} y_{closeness\_tower} \\ y_{period\_tower} \\ y_{trend\_tower} \end{bmatrix}. \tag{7}$$

The $x_{full\_sequence}$ is the input of the whole model, and it will change as the forecast date changes, so it can serve as a query. When $x_{full\_sequence}$ go through the LSTM and FC, we will get the $x_{FC\text{-}out}$, which contains the information about the attention weight of each tower. Then we will use the $x_{FC\text{-}out}$ to control the fusion of $y_{closeness\_tower}$, $y_{period\_tower}$ and $y_{trend\_tower}$, which represent the outputs of three towers separately.

#### *4.1.4. Decoder network*

Decoder is mainly composed of a simple linear layer, a relu layer, and a softmax layer, which is unique for each WEEE prediction task without sharing information. The main purpose of the decoder is to make each task learn the personality feature of the time series, and finally obtain the prediction values of each WEEE respectively.

More details of the decoder we can see from the below calculations: (The $x_{socket\text{-}out}$ flows through two FC and Relu layers, and finally becomes $y_{Output}$)

$$x_{Relu\text{-}in} = FC\left(x_{socket\text{-}out}\right), \tag{8}$$

$$x_{Relu\text{-}out} = \max\left(0, x_{Relu\text{-}in}\right), \tag{9}$$

$$y_{Output} = Relu\left(FC\left(x_{Relu\text{-}out}\right)\right). \tag{10}$$

### *4.2. Multi-time scale aware host task preferred learning model*

As for the original MAHOP/st model, when the model is training, if the parameters are not in the most appropriate range, it is very likely that some of WEEE prediction tasks will pay too much attention to some special time series features, monthly synchronization, annual synchronization or recent trend, and almost ignore others, leading to the bad performance. At the same time, due to the special processing of loss, we have made the MAHOP/st model can distinguish the host task from guest tasks. As a result, compared with undifferentiated MTL, our network structure seems to have gone to the opposite extreme: excessive dependence on personality features, lack of effective extraction of common features, and a tendency to return to STL.

All of the problems mentioned above will lead to the too-high sensitivity and too-slow training speed of our MAHOP/st model. As

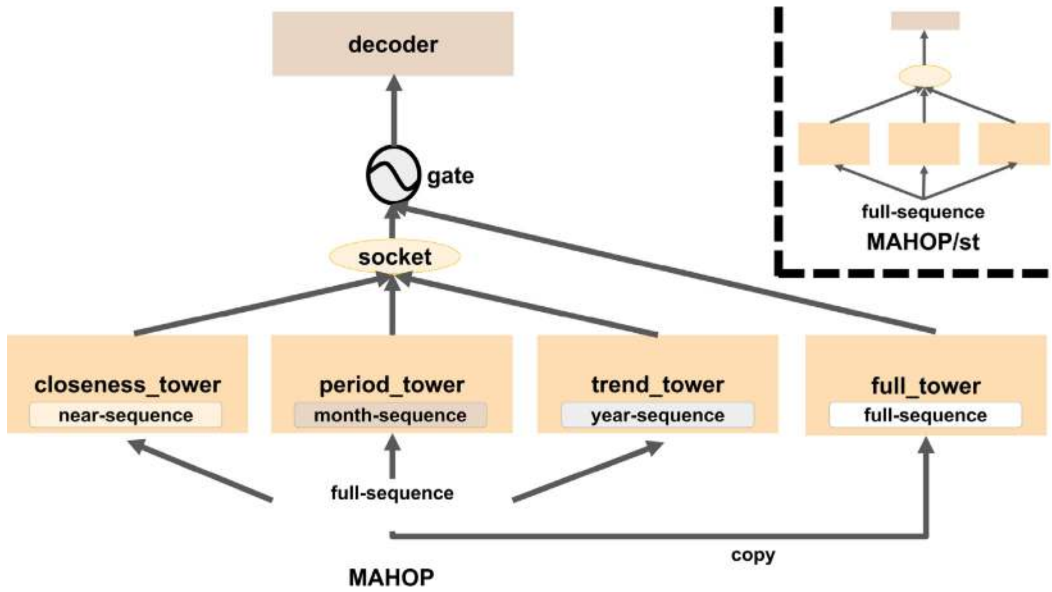


**Fig. 4.** The unique structure of MAHOP compared with MAHOP/st.

a result, we imitate the update process from Multi-gate Mixture-of-Experts to Customized Gate Control and propose our improving model MAHOP.

On account of the similarity between original MAHOP/st and MAHOP, we only show the unique structure of MAHOP in Fig. 4.

As we can see in Fig. 4, based on the MAHOP/st model, MAHOP introduces the full_tower with full-sequence as the input data, containing all the information of three multi-time scale features: annual synchronization, monthly synchronization, and recent trend. The important thing we are supposed to recognize is that the output of full_tower is not directly integrated with the outputs of other towers at the same layer through sockets. Instead, we achieve the fusion purpose through the skip connection, which is established through the gate structure, similar to the socket but only has an FC layer to form the query, and the input of FC is the output of the previous socket instead of the full_sequence. The specific structure of the gate is shown in the following formula:

$$x_{query} = FC\left(x_{socket\text{-}out}\right), \tag{11}$$

$$x_{gate\text{-}out} = x_{query}\begin{bmatrix} x_{socket\text{-}out} \\ y_{full\_tower} \end{bmatrix}. \tag{12}$$

We can get to know from the name that $y_{full\_tower}$ refers to the output of full_tower, and $x_{gate\text{-}out}$ refers to the output of the gate. $x_{query}$ represents the proportion extracted from $y_{full\_tower}$ and $x_{socket\text{-}out}$ by gate structure.

### 4.3. Host-preferred loss and polling host-task learning strategy

In the process of MTL loss summation, we need to make sure that the loss value of guest tasks only plays an auxiliary role and serves to better improve the prediction accuracy of the host task. In addition, we are supposed to achieve fair prediction for all tasks to improve the accuracy of all WEEE prediction tasks.

To achieve that, we first design the Host-preferred Loss (HPL) to meet 2 requirements. For one thing, the loss value of guest tasks should not exceed the loss value of the host task. For another, the loss value of different guest tasks should be in a state of dynamic balance, which means that if one of the losses of guest tasks is far beyond than others, it should attenuate at a minimum speed. Moreover, we put forward the polling host-task learning strategy, which can make sure that each task can be the host task once and the guest task many times (while another task is the host task).

More details of Host-preferred Loss (HPL) are shown below. Firstly, we will specify 2 variables, $X\text{-}Y\text{-}weight$ as well as $TV\text{-}Y\text{-}weight$. We stipulate $X \in \{FD, AC, WM\}$, $Y \in \{FD, AC, WM\}$, and $X \neq Y$, and the formulation set is as follows:

$$X\text{-}Y\text{-}weight = \min\left(1, \frac{X\text{-}loss}{sum\,(GTKs\text{-}loss)}\right) \times \frac{remain\text{-}loss}{sum\,(GTKs\text{-}loss)}, \tag{13}$$

$$TV\text{-}Y\text{-}weight = \min\left(1, \frac{TV\text{-}loss}{sum\,(GTKs\text{-}loss)}\right) \times \frac{sum\,(remain\text{-}loss)}{2 \times sum\,(GTKs\text{-}loss)}. \tag{14}$$

From the above formulations, GTKs-loss represents the loss of separately each guest task, while remain-loss represents the loss of separately each guest task except Y-loss. Finally, we get all the mixloss as follows:

$$X\text{-}mixloss = X\text{-}loss + sum\,(X\text{-}Y\text{-}weight \times Y\text{-}loss), \tag{15}$$

$$TV\text{-}mixloss = TV\text{-}loss + sum\,(TV\text{-}Y\text{-}weight \times Y\text{-}loss). \tag{16}$$

### 4.4. Algorithm process

**Algorithm 1** MAHOP Algorithm

**Input:** temporal features series of the host task (HTK) $R_{HTK}$;
temporal features series of guest tasks (GTKs) $R_{GTKs}$;
**Output:** return HTK-mixloss;
1: Stack $HTK_{full\text{-}sequence}$ and $GTKs_{full\text{-}sequence}$ to $full\text{-}sequence$;
2: Define time windows: $cw$, $pw$, $tw$;
3: Calculate tower output: $near_{LSTM\text{-}out}$, $month_{LSTM\text{-}out}$, $year_{LSTM\text{-}out}$, $full_{LSTM\text{-}out}$;
4: Get HTK fusion result from socket:

$$HTK_{socket\text{-}out} = socket\left(full\text{-}sequence \quad \begin{bmatrix} near_{LSTM\text{-}out} \\ month_{LSTM\text{-}out} \\ year_{LSTM\text{-}out} \end{bmatrix}\right);$$

5: Calculate HTK gate output:

$$HTK_{gate\text{-}out} = gate\left(full_{LSTM\text{-}out} \quad HTK_{socket\text{-}out}\right);$$

6: Decode HTK gate-out to output and get corresponding loss: $HTK\text{-}loss$;
7: **while** not all GTKs are through calculation **do**

**Table 1**
Datasets description.

| | Training set | Validation set | Test set |
|---|---|---|---|
| Data length | 273 | 30 | 31 |
| Start date | 2019–02–01 | 2019–11-01 | 2019–12–01 |
| End date | 2019–10–31 | 2019–11-30 | 2019–12–31 |
| Time sequence | year_len:7 month_len:7 near_len:15 | | |

Get GTKn fusion result from socket:

$$GTKn_{socket\text{-}out} = socket_n\left( full\text{-}sequence \quad \left[ \begin{array}{c} near_{LSTM\text{-}out} \\ month_{LSTM\text{-}out} \\ year_{LSTM\text{-}out} \end{array} \right] \right);$$

Get GTKn gate output:

$$GTKn_{gate\text{-}out} = gate\left( full_{LSTM\text{-}out} \quad GTKn_{socket\text{-}out} \right);$$

Decode GTKn gate-out to output and get corresponding loss: $GTKn\text{-}loss$;

**end while**

Calculate HTK-mixloss:

$$HTK\text{-}mixloss = HPL\left( HTK\text{-}loss \quad GTK1\text{-}loss \cdots \quad \cdots GTKn\text{-}loss \right);$$

**return** HTK-mixloss;

## 5. Experiment setup

We use the data provided by Aibo green, a commercial recycling company, which contains the daily returns of 4 kinds of electric appliances within 2 years. We apply our model and the previous MTL as well as single task learning (STL) methods to these specific RL datasets and prove that our model has a better effect compared with others. We will show the details of the datasets, the information on the contrast models, the measurement of all the methods, and the process of the experiment as follows.

### *5.1. Datasets*

The original data are presented in the form of orders placed by users of Aibo green, which are then reformed by us to change into returning amounts of different types of WEEE. All the RL data we can get from Aibo green includes 5 types of WEEE, which are respectively fridges (FD), air conditioners (AC), washing machines (WM), televisions (TV), and other machines (OM). Unfortunately, the data of OM completely do not have monthly periodicity and annual periodicity features. Moreover, it is not similar to other WEEE time series, which means that any method of MTL is not suitable to study the common features between OM and other WEEE. As a result, we only use the data of the first 4 kinds of electronic machines and show them in Fig. 5.

As we can see in Fig. 5, the data of all types of WEEE have monthly periodicity, annual periodicity, and near trend features. In addition, their time series of them are similar to each other, so putting them together to train the model will be beneficial to the MTL task. We can find that the 2018 and 2019 return data have almost the same trend in Fig. 5, but if we want to use this annual periodicity, we have to put the return data of 2018 in the model to help predict the 2019 return data, which means that the number of total samples can be up to the length of one year. Moreover, we also find that the return data of four types of WEEE in the first month is irregularly and randomly fluctuating, so the data of the first month is discarded. Finally, we divide the rest data into 3 parts: training set at the rate of 81%, validation set at the rate of 9.5%, and test set at the rate of 9.5%. More information on the 3 parts of data is shown in Table 1.

### *5.2. Baselines*

We include other 7 contrast models besides our MAHOP/st and MAHOP models. The first model is the STL model which is proposed by Zhang et al. (2023) and has been proven to work so well in this specific WEEE prediction task, while the next 3 models are classical methods in the MTL field, and the rest are some improved models proposed by us to make the comparison.

- MULAN (Zhang et al., 2023): This model is the first one to introduce multi-timescale windows and attention fusion. As a result, it will capture the temporal dependence for all the WEEE about the year period, month period, and near trend.
- MMOE (Ma et al., 2018): After the same four copies of input data containing annual, monthly, and short-term trend features, they are respectively transferred to four experts (LSTM networks) for information fusion, and differentiated features are extracted through four sockets, and also, each socket serves four experts at the same time.
- CGC (Tang et al., 2020): Based on the MMOE model, the CGC adds a set of identical input data as well as an identical expert, and each socket can only fuse information of its unique single expert and the newly added expert.
- Asymmetry-MMOE (A-MMOE) (Tang et al., 2020): Based on the MMOE model, the four categories of electrical appliances are trained four times respectively. The socket of each task to be predicted fuses all expert information, but the socket of other categories of electrical appliances can only use their expert information.
- Lossfixed-MMOE (L-MMOE): Based on the MMOE model, the four categories of electrical appliances were trained four times. In the process of loss summation, the loss value coefficient of each task to be predicted was 1, and the loss value coefficient of other tasks was 0.1.
- MAHOP/st (lossfixed): This model adopts the MAHOP/st framework, but replaces the HPL algorithm to calculate the loss to the simple loss-fixed method which has been introduced in L-MMOE.
- MAHOP/st (gradnorm): This model adopts the MAHOP/st framework, but replaces the HPL algorithm to calculate the loss to gradnorm, which can make each loss adaptively adjust according to the current weight.

### *5.3. Measurement*

To calculate the effect of different models, we use 2 evaluations, which are respectively Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). In addition, we also introduce Average Epoch Number (AEN) to represent the efficiency of each model (We use $epoch_i$ to represent the number of epochs of the $i$th WEEE training). When it comes to sensitivity, we define two kinds of methods to measure it, which can better describe the influence of the change of parameters for the RL prediction task. The first one is called the Border of Worst Performance and Best Performance (B-WP-BP), and the other is called the Average Deviation of Performance (ADP). The following formulas show the calculation process of the above indicators:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \hat{x}_i\right)^2}, \tag{17}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|x_i - \hat{x}_i\right|, \tag{18}$$

$$AEN = \frac{1}{n}\sum_{i=1}^{n} epoch_i, \tag{19}$$

$$B - WP - BP = x_{max} - x_{min} \qquad x \in \{RMSE, \mathrm{MAE}\}, \tag{20}$$

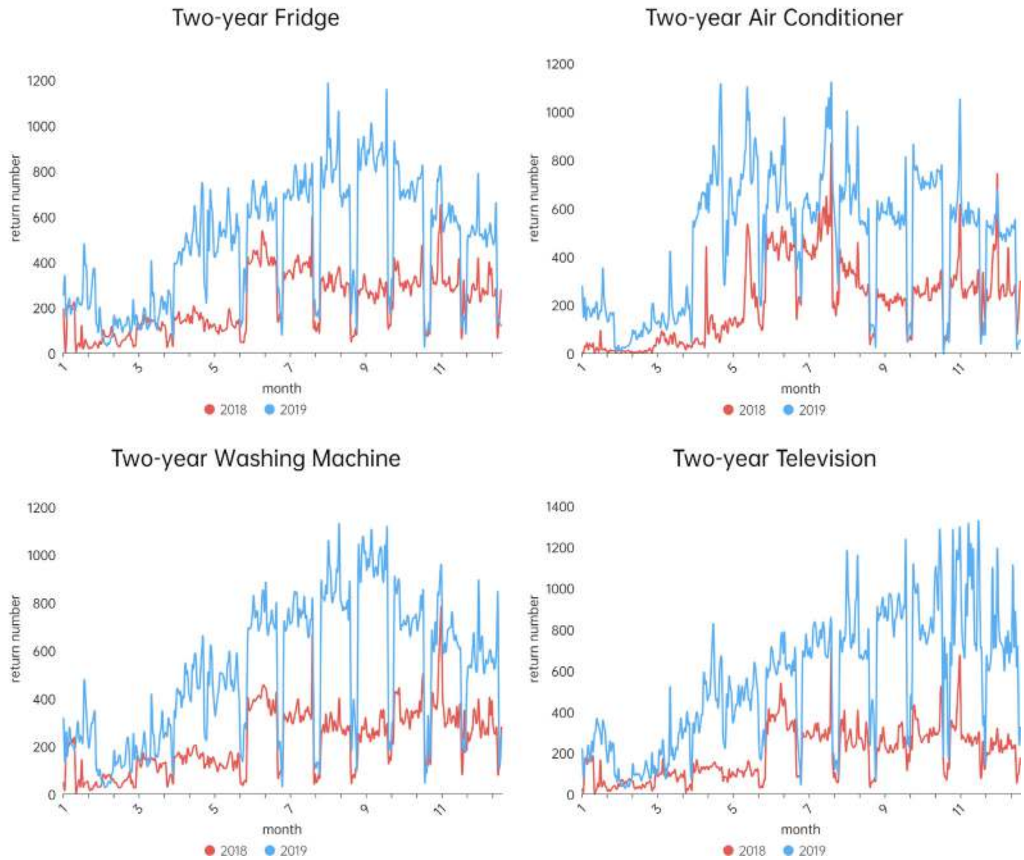


**Fig. 5.** The return data of 4 types of WEEE within 2 years.

$$ADP = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\left|x_i - x_{avg}\right|} \qquad x \in \{RMSE, \mathrm{MAE}\}. \tag{21}$$

We can find that B-WP-BP represents the range of model performance changed with parameters, while the ADP shows the average index of model sensitivity. The design of ADP can achieve two requirements as follows:

1. When we alter parameters, the greater changes in model effect, the higher sensitivity.
2. It can minimize the influence of outlier points on the final model sensitivity, which is also the reason why we introduce the square root calculation.

### 5.4. Process

We set up five groups of controlled experiments in total. The first four experiments are designed to verify and compare the accuracy and efficiency of different models, while the last experiment is about sensitivity, in order to show the change of the model sensitivity after we adjust and improve our model from MAHOP/st to MAHOP.

## 6. Results and analyses

We compare the performance and efficiency among different models in the WEEE RL prediction field and also figure out the strengths and weaknesses of our MAHOP/st model due to the experimental result. Moreover, we analyze the sensitivity of both our original model MAHOP/st and our improved model MAHOP, in order to demonstrate that the adjustment of our initial model indeed solves some weaknesses.

### 6.1. Overall performance comparison

For the first 4 groups of contrast experiments, we compare the models which are separately classical MTL models with MULAN, some present asymmetry MTL methods with MULAN and our MAHOP/st, variants of the MAHOP/st model without HPL with MULAN and our MAHOP/st, as well as MAHOP/st model with MAHOP model. The result of the comparison is shown in Fig. 6. Both RMSE and MAE of TV are far beyond other WEEE so it is not clear to display all four WEEE data in one picture at the same time. As a result, we just show FD, AC, and WM in Fig. 6, and all WEEE indicators, including TV, are shown in Table 2.

Firstly, we compare the MULAN with the current popular multitask model for experimental results. We find that both the MMOE model and the CGC model have an obvious feature: they greatly shorten the time required (equal to AEN) to train to the stable level, but the indicators of RMSE and MAE are far greater than MULAN. As we all know, the MTL model itself is good at making full use of the commonalities among different tasks. For specific tasks researched in this paper, the data of FD, AC, WM, and TV all have a certain degree of monthly synchronization, annual synchronization, and recent trend features. However, the dependence of these four WEEE on 3 time features is not completely the same. Therefore, simple MTL can indeed learn the common features of all household appliances in a short period, but due to the limited ability to learn the special feature of each WEEE, it will result in poor training results.

Secondly, for all the asymmetry MTL models, the learning effect and accuracy rate are far better than those of the previous undifferentiated MTL method, but correspondingly, their AEN exceeds the MULAN model. The performance of A-MMOE and L-MMOE models only has little improvement compared with MULAN in some WEEE prediction tasks, while for other WEEE, the prediction effect is even worse. By contrast, the improved MAHOP/st model in this paper not only has a

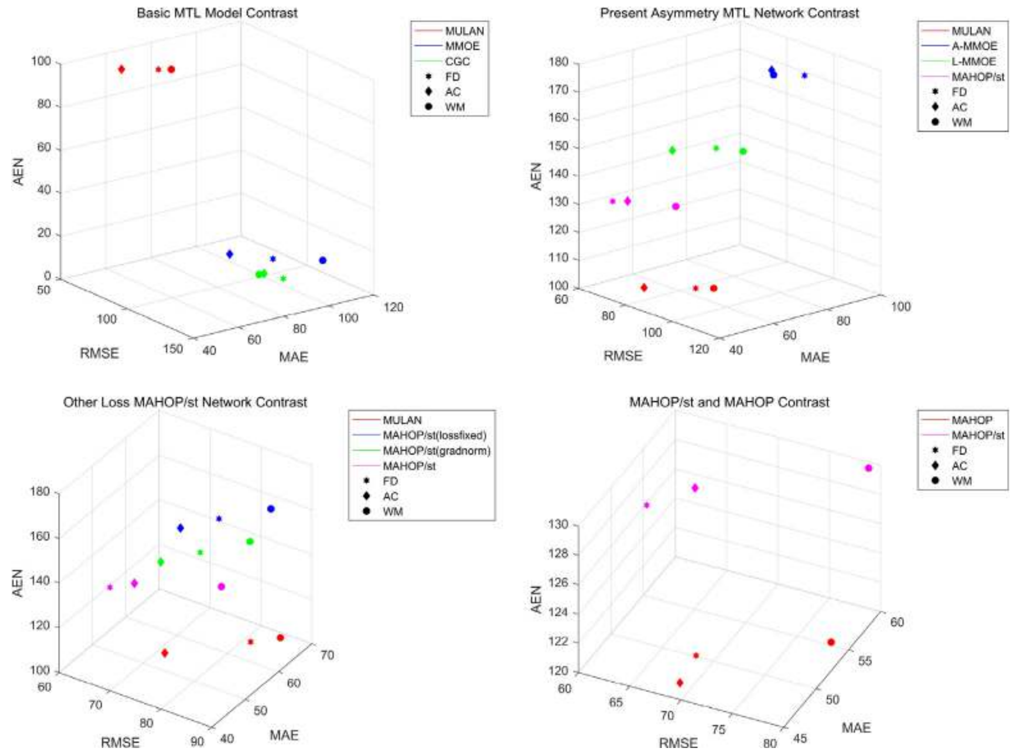


**Fig. 6.** Comparison results of the efficiency and the prediction effect of different methods.

much better prediction effect than MULAN for all WEEE but also is the most efficient one of the three preferential MTL models in terms of AEN indicators.

After analyzing the third picture, we get the conclusion that no matter what kind of loss calculation method is adopted, as long as we use the MAHOP/st framework and the loss algorithm is not equal to the original method (adding directly), the accuracy of them will exceed the MULAN model. However, it is still obvious that when HPL is used, the time for model training is shortened, and the AEN index decreases, which represents that the efficiency of the MAHOP/st model is improved. From another aspect, compared with the rest loss calculation methods, the MAE and RMSE indicators have been greatly improved.

Finally, the last result shows us that MAHOP performs better for WM prediction, while MAHOP/st performs better for FD prediction, which means when it comes to the model effect, MAHOP and MAHOP/st each have their own merits. On the other hand, we find that the AEN indicator for MAHOP is 10 epochs less than MAHOP/st, regarded as a small improvement in efficiency, and the most important superiority of MAHOP compared with MAHOP/st, the acceptable sensitivity, will be introduced in Section 6.3.

Upon closely examining the data from MTL models, STL models, and Ablation models presented in Fig. 6, distinct patterns emerge. As we transition from MTL models through STL models and onto Ablation models, there is a discernible trade-off: while efficiency diminishes, prediction accuracy sees consistent enhancement. Intriguingly, within the Ablation models category, MAHOP stands out, demonstrating an unmatched balance of prediction accuracy and operational efficiency.

Several insights can be gleaned from these observations:

- **Efficiency vs. Accuracy in MTL and STL models:** MTL models seem to emphasize operational efficiency. However, potential gradient conflicts across tasks might compromise their predictive prowess, leading STL models to outperform them in terms of accuracy.
- **The Role of GTKs in Ablation Models:** Ablation models, designed to predict for each household appliance distinctly, leverage GTKs to bolster the primary task key (HTK) in predictive performance. The integration of these GTKs requires additional computational resources, elucidating the trade-off between accuracy and efficiency. Furthermore, the comparative analysis among the ablation models underscores the prowess of our HPL algorithm. It not only amplifies predictive accuracy but also streamlines the learning process.

In the realm of real-world implications, our model stands as a valuable tool for WEEE prediction, facilitating: (1) Resource Allocation for Recycling Companies: The heightened accuracy empowers recycling enterprises to optimize their personnel deployment. This includes determining the number and qualifications of personnel for specific regions. (2) Operational Efficiency for Dismantling Companies: The dismantling entities can effectively allocate their resources, translating into cost savings and enhanced operational efficiency.

### 6.2. Strengths of MAHOP

The strengths of our model compared with others are described as follows:

(1) Our model is the only one to introduce the multi-time scale features to MTL. For other MTL models, the inputs, as well as the network structures for different experts, are the same, while for MAHOP, we input the year series data, month series data, and near trend data separately in 3 different experts, and the LSTM structures of the 3 experts are also different due to the length of data.

(2) We use the special loss processes, the HPL as well as the polling host-task learning strategy, to stress the importance of the host task and make the training speed of each task can adjust by itself. Most of the loss calculation methods do not have these two functions, except for two algorithms mentioned previously in this paper, lossfixed and gradnorm. But even for these two

**Table 2**
The detailed Comparison of each type WEEE among different methods.

| | | FD | AC | WM | TV |
|---|---|---|---|---|---|
| MULAN | AEN | 100 | 100 | 100 | 100 |
| | MAE | 64.92 | 54.41 | 68.66 | 171.41 |
| | RMSE | 81.33 | 71.36 | 84.72 | 216.84 |
| MMOE | AEN | <u>15</u> | <u>15</u> | <u>15</u> | <u>15</u> |
| | MAE | 93.27 | 83.88 | 106.67 | 186.45 |
| | RMSE | 119.47 | 102.81 | 134.3 | 234.26 |
| CGC | AEN | **10** | **10** | **10** | **10** |
| | MAE | 90.51 | 88.26 | 86.27 | 186.32 |
| | RMSE | 131.88 | 121.29 | 120.96 | 229.52 |
| A-MMOE | AEN | 180 | 180 | 180 | 180 |
| | MAE | 81.15 | 77.28 | 74.94 | 167.02 |
| | RMSE | 109.14 | 99.56 | 103.07 | 205.56 |
| L-MMOE | AEN | 150 | 150 | 150 | 150 |
| | MAE | 69.31 | 58.65 | 73.01 | 167.35 |
| | RMSE | 85.08 | 78.58 | 92.24 | 215.38 |
| MAHOP/st (lossfixed) | AEN | 140 | 140 | 140 | 140 |
| | MAE | 57.13 | 51.32 | 64.25 | 166.24 |
| | RMSE | 80.22 | 76.47 | 85.76 | 200.09 |
| MAHOP/st (gradnorm) | AEN | 135 | 135 | 135 | 135 |
| | MAE | 59.28 | 53.39 | 66.56 | 171.25 |
| | RMSE | 75.13 | 71.24 | 80.12 | 203.47 |
| MAHOP/st (HPL) | AEN | 130 | 130 | 130 | 130 |
| | MAE | **49.12** | <u>52.33</u> | <u>59.35</u> | **162.47** |
| | RMSE | **64.06** | **66.75** | <u>79.24</u> | <u>195.09</u> |
| MAHOP | AEN | 120 | 120 | 120 | 120 |
| | MAE | <u>50.12</u> | **46.79** | **55.32** | <u>165.05</u> |
| | RMSE | <u>68.25</u> | <u>68.77</u> | **78.12** | **193.46** |

methods, they can only achieve one of the two purposes, because the lossfixed algorithm has fixed parameters for each loss, without dynamic adjustment, and the gradnorm method cannot distinguish the host task from guest tasks.

(3) The performance of our model is the best, and far beyond any other model mentioned in this paper for the WEEE RL prediction task. MAHOP/st as well as MAHOP both behave better than other models for all types of WEEE, including FD, AC, WM, and TV.

(4) We avoid the host task paying too much attention to some specific time series features, so the sensitivity of our model can be controlled in an acceptable range. The requirement of the parameters of our original MAHOP/st, such as the input data length (IDL) and learning rate (LR), are very strict, which means that any tiny change to the parameters will lead to significant deterioration of the model prediction effect due to the pretty high sensitivity. By contrast, after adding one more sharing tower, the sensitivity of our final MAHOP is greatly decreased to a pretty low level.

### *6.3. Sensitivity analysis*

We compared the sensitivity of the two models in four kinds of WEEE with LR changing from 0.005 to 0.02 (increased in steps of 0.001), and IDL changing from 23 (year_len:5, month_len:5, near_len:13) to 38 (year_len:10, month_len:10, near_len:18) (increased in steps of 1 for each sequence). The indicators of TV prediction performance, RMSE, and MAE fluctuate most significantly and the model itself is also the most unstable when it is applied to the task of TV prediction. We only show the comparison between MAHOP/st and MAHOP. To present the difference between the two models more obviously, we only select TV for sensitivity display in Fig. 7, by comparing the trend of RMSE and MAE with the change of parameters LR and IDL. In addition, the results of all WEEE sensitivity comparisons with the two models are shown in Table 3.

**Table 3**
The sensitivity comparison of all WEEE.

| | | | | FD | AD | WM | TV |
|---|---|---|---|---|---|---|---|
| MAHOP/st | B-WP-BP | LR | MAE | 5.44 | 7.37 | 8.33 | 18.69 |
| | | | RMSE | 6.32 | 10.02 | 11.49 | 15.94 |
| | | IDL | MAE | 5.2 | 7.15 | 9.28 | 16.62 |
| | | | RMSE | 5.98 | 11.33 | 13.74 | 18.38 |
| | ADP | LR | MAE | 1.11 | 1.08 | 1.32 | 1.74 |
| | | | RMSE | 1.23 | 1.21 | 1.67 | 1.91 |
| | | IDL | MAE | 1.06 | 1.03 | 1.3 | 1.87 |
| | | | RMSE | 1.2 | 1.17 | 1.72 | 2.03 |
| MAHOP | B-WP-BP | LR | MAE | 1.67 | 2.13 | 3.24 | 2.27 |
| | | | RMSE | 3.58 | 7.41 | 6.38 | 5.19 |
| | | IDL | MAE | 2.64 | 3.62 | 4.27 | 4.33 |
| | | | RMSE | 5.28 | 6.88 | 7.35 | 7.12 |
| | ADP | LR | MAE | 0.41 | 0.49 | 0.61 | 0.75 |
| | | | RMSE | 0.66 | 0.83 | 0.88 | 1.15 |
| | | IDL | MAE | 0.58 | 0.71 | 0.96 | 1.03 |
| | | | RMSE | 0.75 | 0.94 | 1.11 | 1.22 |

In Fig. 7, we find that the sensitivity of MAHOP is much lower than MAHOP/st for both LR and IDL, which corresponds to the purpose we design the MAHOP. As a result, after the improvement of our original model, MAHOP will perform better when we apply the model to predict the amount of waste household appliances for recycling. Because it has stronger robustness to help resist the interference of external factors' fluctuations.

## 7. Conclusion

In this paper, we first introduce the background of WEEE and RL, leading to the inference that the prediction of all types of WEEE is a very important part of the RL field. Then we briefly present some classical STL models, which have been proven to be effective for time series prediction tasks. However, through the analysis of the similarities among different WEEE time sequence data, we find that the STL methods cannot perform well in our tasks. As a result, we turn to research the MTL methods, but after investigating carefully, we find that existing MTL models never consider the multi-time scale features. Moreover, these models pay more attention to efficiency and total performance than the prediction accuracy of specific each WEEE, which is the opposite of our task.

In order to improve the effectiveness of all the WEEE prediction tasks at the same time, we create a MAHOP model inspired by the Multi-gate Mixture-of-Experts model, replacing the original experts with the tower structures, which can accept input of various shapes and have networks (LSTM) of different lengths. In addition, we also use the HPL algorithm to make the model have different preferences for the host task as well as the guest tasks, and the polling host-task learning strategy to achieve fair prediction for all tasks. The HPL algorithm along with the polling host-task learning strategy enables the model to focus on improving the prediction accuracy of one kind of WEEE every time and repeat it many times until the performance of prediction tasks of all types of WEEE can be improved. On account of the high sensitivity of our original MAHOP/st model, we add one more sharing tower, and get our final MAHOP, to fix this problem.

Applying our model to solve the prediction tasks in this paper, we design four groups of contrast experiments to compare the efficiency and effect of different models and one more contrast experiment to analyze the sensitivity of MAHOP/st and MAHOP models. Eventually, we get the conclusion that our final model, MAHOP, has a great improvement of effect among all the WEEE prediction tasks, with a little more time cost, compared with the rest of MTL and STL methods and performs well slightly for efficiency and great for sensitivity.

This research contributes a novel model to the domain with noteworthy strengths, but we recognize certain areas for enhancement. Notably: (1) Our model demonstrates a commendable performance,

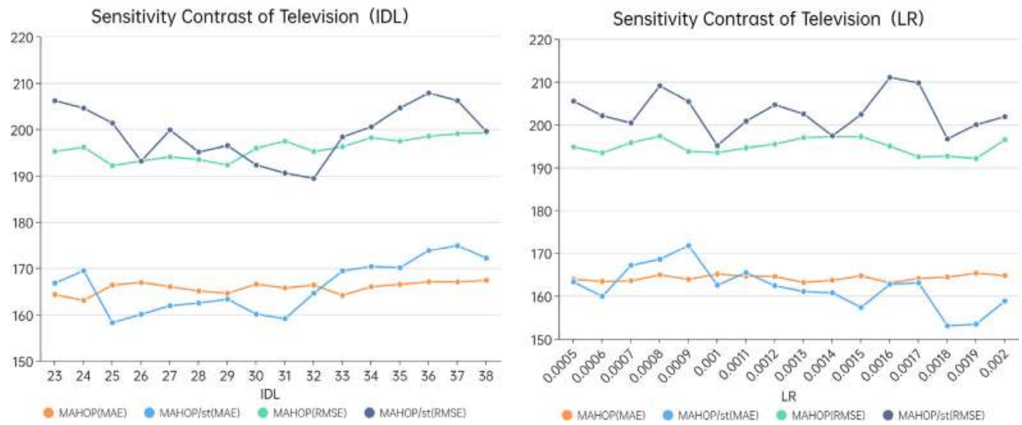


**Fig. 7.** Comparison of sensitivity between MAHOP/st and MAHOP.

surpassing both MTL and STL methodologies in accuracy. However, the efficiency front presents a scope for optimization. (2) While specific household electrical appliances like FD, AD, WM, and TV showcase congruent time series patterns, our model currently does not cater to integrating appliances devoid of such resemblances. This limitation delineates an avenue for further research and enhancement. (3) The dataset's sparsity, particularly in some provinces and specific recycling points, restrains our model's capability to precisely predict recycling volumes at the provincial or individual recycling point level. Future iterations may benefit from the integration of transfer learning to mitigate this limitation.

## CRediT authorship contribution statement

**Yujiang Wu:** Methodology, Investigation, Writing – original draft, Writing – review & editing. **Min Gao:** Conceptualization, Validation, Writing – review & editing. **Ruiqi Liu:** Methodology, Validation, Writing – review & editing. **Jie Zeng:** Validation, Resources, Project administration. **Quanwu Zhao:** Validation, Writing – review & editing, Resources. **Jinyong Gao:** Validation, Resources. **Jia Zhang:** Validation, Resources.

## Declaration of competing interest

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

## Data availability

The data that has been used is confidential.

## Acknowledgments

This study was supported by the National Key Research and Development Program of China (2020YFB1712901).

## References

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271.

Caruana, R. (1993). Multitask learning: A knowledge-based source of inductive bias1. In *Proceedings of the tenth international conference on machine learning* (pp. 41–48). Citeseer.

Directive, E. (2012). Directive 2012/19/EU of the European parliament and of the council of 4 July 2012 on waste electrical and electronic equipment, WEEE. *Official Journal of the European Union L*, *197*, 38–71.

Eigen, D., Ranzato, M., & Sutskever, I. (2013). Learning factored representations in a deep mixture of experts. arXiv preprint arXiv:1312.4314.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hu, J., & Zheng, W. (2020). Multistage attention network for multivariate time series prediction. *Neurocomputing*, *383*, 122–137.

Islam, M. T., & Huda, N. (2018). Reverse logistics and closed-loop supply chain of waste electrical and electronic equipment (WEEE)/E-waste: A comprehensive literature review. *Resources, Conservation and Recycling*, *137*, 48–75.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, *3*(1), 79–87.

Kelle, P., & Silver, E. A. (1989). Forecasting the returns of reusable containers. *Journal of Operations Management*, *8*(1), 17–35.

Lee, H. B., Yang, E., & Hwang, S. J. (2018). Deep asymmetric multi-task feature learning. In *International conference on machine learning* (pp. 2956–2964). PMLR.

Lehtinen, U., & Poikela, K. (2006). Challenges of WEEE on reverse logistics: A case study on a collection network in Finland. In *Proceedings of logistics research network annual conference*: *vol. 2006*, (pp. 6th–8th).

Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., & Chi, E. H. (2018). Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1930–1939).

Mahmoud, R. A., Hajj, H., & Karameh, F. N. (2020). A systematic approach to multi-task learning from time-series data. *Applied Soft Computing*, *96*, Article 106586.

Qian, X. Y., & Gao, S. (2017). Financial series prediction: Comparison between precision of time series models and machine learning methods. (pp. 1–9). arXiv preprint arXiv:1706.00948.

Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, *36*(3), 1181–1191.

Shen, L., & Wang, Y. (2022). TCCT: Tightly-coupled convolutional transformer on time series forecasting. *Neurocomputing*, *480*, 131–145.

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018). A comparison of ARIMA and LSTM in forecasting time series. In *2018 17th IEEE international conference on machine learning and applications* (pp. 1394–1401). IEEE.

Standley, T., Zamir, A., Chen, D., Guibas, L., Malik, J., & Savarese, S. (2020). Which tasks should be learned together in multi-task learning? In *International conference on machine learning* (pp. 9120–9132). PMLR.

Tang, H., Liu, J., Zhao, M., & Gong, X. (2020). Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Fourteenth ACM conference on recommender systems* (pp. 269–278).

Toktay, L. B., Wein, L. M., & Zenios, S. A. (2000). Inventory management of remanufacturable products. *Management Science*, *46*(11), 1412–1426.

Tuan, N. A., Jeong, H., Yang, E., & Hwang, S. (2019). Temporal probabilistic asymmetric multi-task learning.

Wang, L., Zhang, L. J., Chen, Y., Wang, W., Xiao, W., & Wang, S. (2012). Reverse logistics predicting model and its application. In *2012 IEEE eighth world congress on services* (pp. 399–406). IEEE.

Warrier, S., Rutter, E. M., & Flores, K. B. (2022). Multitask neural networks for predicting bladder pressure with time series data. *Biomedical Signal Processing and Control*, *72*, Article 103298.

Yang, Y., & Williams, E. (2009). Logistic model-based forecast of sales and generation of obsolete computers in the US. *Technological Forecasting and Social Change*, *76*(8), 1105–1114.

Zhang, J., Gao, M., Zhao, L., Hu, J., Gao, J., Deng, M., Wan, C., & Yang, L. (2023). Multi-time scale attention network for WEEE reverse logistics return prediction. *Expert Systems with Applications*, *211*, Article 118610.

Zhang, Y., & Yan, J. (2022). Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*.

Zhang, Y., & Yang, Q. (2021). A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 35*, (no. 12), (pp. 11106–11115).