DT-MUSA: Dual Transfer Driven Multi-source Domain Adaptation for WEEE Reverse Logistics Return Prediction

Ruiqi Liu[1,2], Min Gao[1,2(✉)], Yujiang Wu[1,2], Jie Zeng[2], Jia Zhang[2], and Jinyong Gao[3]

[1] Key Laboratory of Dependable Service Computing in Cyber Physical Society (Chongqing University), Ministry of Education, Chongqing 401331, China
{liuruiqi,wuyujiang}@stu.cqu.edu.cn
[2] School of Big Data and Software Engineering, Chongqing University, Chongqing 401331, China
{gaomin,zengjie,jiazhang}@cqu.edu.cn
[3] Aibo Green Reverse Supply Chain Co., Ltd., Shenzhen 518000, China
gjy@boolv.com

Abstract. Reverse logistics (RL) return prediction for Waste Electrical and Electronic Equipment (WEEE) has gained attention due to its potential to improve operational efficiency in the recycling industry. However, in data-scarce regions, commonly used deep learning models perform poorly. Existing multi-source cross-domain transfer learning models can partially overcome data scarcity by using historical data from multiple sources. However, these models aggregate multi-source domain data into a single-source domain in transfer, ignoring the differences in time series features among source domains. Additionally, the lack of historical data in the target domain makes fine-tuning the prediction model inoperative. To address these issues, we propose Dual Transfer Driven Multi-Source domain Adaptation (DT-MUSA) for WEEE RL return prediction. DT-MUSA includes a dual transfer model that combines sample transfer and model transfer and a basic prediction model MUCAN (Multi-time Scale CNN-Attention Network). It employs a multi-task learning to aggregate predictors from multiple regions and avoids negative transfer learning. The dual transfer model enables fine-tuning of the base model MUCAN by generating long-term time series data through sample transfer. We applied DT-MUSA to real cases of an RL recycling company and conducted extensive experiments. The results show that DT-MUSA outperforms baseline prediction models significantly.

Keywords: waste electrical and electronic equipment · reverse logistics return predction · dual transfer learning · multi-task learning

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2024
Published by Springer Nature Switzerland AG 2024. All Rights Reserved
H. Gao et al. (Eds.): CollaborateCom 2023, LNICST 563, pp. 365–384, 2024.
https://doi.org/10.1007/978-3-031-54531-3_20

1 Introduction

As reported by the United Nations University, only about 20% of the 44.7 million tons of WEEE generated worldwide each year undergo proper recycling and treatment [13,29]. If not properly treated promptly, the leaching of hazardous substances from a large volume of WEEE can pose significant risks to the environment and human health [7,30,32]. The use of RL return data prediction can enhance the efficiency of WEEE RL by supporting transportation scheduling, labor and material scheduling, and production planning for reverse recycling efforts [4,14,25]. Therefore, the RL return prediction of WEEE has received widespread attention [11,16].

[15,38] earlier investigated RL return prediction based on regression equations. A Bayesian-based prediction model developed by [34] assumed that RL flows obey a binomial probability distribution. [23] modeled RL return prediction by analytical moving averages, exponential smoothing, and causal analysis. These studies considered the time-series relationship of the data and achieved good results under certain applicable conditions. However, the models are extremely dependent on feature selection, and different modeling is required for different application scenarios, leading to difficulties in applying to complex scenarios. To be able to build an RL prediction model with generalization ability, [40] recently proposed a deep learning-based multi-time scale attention network (MULAN). By dividing the closeness window, period window, and trend window in the historical data as model inputs, this approach is able to capture various features at different time scales of the series. As a result, there is a significant improvement in the accuracy of the prediction. However, the prediction accuracy of MULAN is severely degraded in some scenarios lacking long-term history data (see Sect. 5.1), because historical data are scarce and trend window inputs are not available.

To improve model performance in historical data lacking scenarios, researchers proposed transfer learning that learns knowledge from selected sufficient data related to the target domain with sparse samples [5,12,26]. However, in RL prediction scenarios, the distribution of source and target domain data may differ significantly due to geographical and temporal differences, leading to severe negative transfer [28,39]. Many recent studies have tried to employ multi-source data adaptive transfer learning in the expectation that pre-trained networks can extract richer sets of common features in multiple source domains and overcome the differences in data distribution from source to target domains [6,10]. These methods can be used to predict RL return data in the data-lacking scenario, but there are still two challenges.

The first challenge is how to utilize multi-source data effectively to mitigate negative transfer. In a multi-source knowledge transfer learning task, the distribution of data in each source domain varies, and therefore, their contributions to the target domain task should be adjusted accordingly, so a key issue is how to adaptively aggregate the source domain predictors [3,33]. Traditional studies generally assign weights to source domains or select source domains subjectively and directly by domain similarity [1,9,17]. However, existing two-stage learning methods (involving domain selection and model transfer) lack adaptive algorithms capable of accurately quantifying the similarity between the source and target domains in relation to the assigned weights.

The second challenge is the lack of long-term time series data leading to suboptimal fine-tuning of model transfer, especially when the base prediction model for transfer needs to extract the features from multi-time scale windows. That is because trend data cannot be used in this case, and the multi-time window input structure does not have sufficient windows of data, leading to suboptimal fine-tuning of the model transfer eg. MULAN [40].

To this end, we propose Dual Transfer Driven Multi-source Domain Adaptation (DT-MUSA). In DT-MUSA, (1) we first propose a multi-source domain adaptation scheme based on multi-task learning to avoid the possible negative transfer effects caused by subjective source domain predictors aggregation. We compute the similarity between the data in the source and target domains and prioritize the source domains for labeling. After that, we use a multi-task learning strategy to perform association learning on the data from source domains and dynamically integrate the prediction loss of each source domain during pre-training. The model can adaptively aggregate the source domain predictors during multi-task association learning to minimize negative transfer by learning the mutual base model parameters in the shared layer of all tasks. (2) We propose a dual transfer model (model and sample transfer) and a base prediction model MUCAN (Multi-time Scale CNN-Attention Network) to tackle the suboptimal fine-tuning of the model transfer due to long-term time series data scarcity. First, we employ a sample transfer strategy to generate long-term time series data. Then, we build a specialized multi-scale time-series feature extraction network, referred to as MUCAN, which is built upon the convolutional attention module and utilizes a multi-time scale input window structure. This serves as the foundation for our dual transfer prediction model. Finally, we use the pre-trained network of the source domain set for model transfer, initialize the shared layer parameters of MUCAN in the target domain, and then fine-tune the shared layer parameters of the model to better adapt to the data distribution present in the target domain.

In summary, our main contributions are as follows:

- We propose a novel approach to multi-source domain adaptation that utilizes multi-task learning to acquire mutual knowledge from various source domains. As far as we know, this is the first study to employ multi-task learning to mitigate possible negative transfer effects in the field of RL time series prediction.
- We propose a dual transfer model, where sample transfer is used to generate long-term time series of the target domain, and model transfer is utilized to effectively transfer mutual knowledge obtained from multiple source domains in multi-source domain adaptation. In addition, we propose a base prediction model MUCAN for model transfer, which relies on the convolutional attention module to obtain the degree of influence of different time-scale encodings on prediction and has better results than other network structures in encoding fusion.

– To evaluate the effectiveness of our model DT-MUSA, we apply it to a real-world case involving an enterprise specializing in RL returns. Through extensive experiments and ablation experiments, we analyze the benefits of utilizing multi-task learning for multi-source knowledge fusion and utilizing the dual transfer model.

2 Related Works

To overcome the distributional disparity between source and target domain data in transfer learning, and enhance the transfer effect. In the last decade, many research results on domain adaptation in transfer learning have been published. Shallow domain adaptation methods are typically used to establish a connection between the source and target domains by either learning invariant features or estimating the importance of instances from the source domain [20]. For example, [22] used a modified Transfer Naive Bayes (TNB) as a prediction model. They employed a similarity measure based on ranges to allocate weights to source domain instances, and subsequently trained the prediction model using these weighted instances. [24] proposed TCA+ to reduce the differences in the distribution of features that make the source and target domains.

With the rising prevalence of deep neural networks, there has been an increasing interest in investigating deep domain adaptation techniques. This type of method utilizes an adaptive module embedded in the deep architecture to minimize differences between the source and target domains. [19,35] proposed the DDC (Deep Domain Confusion) method and DAN (Deep Adaptation Network), respectively, which diminish the dissimilarity between source and target domains by introducing an adaptive adaptation layer or an additional domain confusion loss. Later, [21] extended DAN by proposing Joint Adaptation Network (JAN), which further considers the joint probability distribution of features and labels. [28] introduce a novel deep Transfer Learning based on Transformer (TLT) model that utilizes a recurrent fine-tuning transfer learning approach during the pre-training phase of knowledge transfer. The purpose is to prevent deep learning models from overfitting the source data and reduce domain gap between the source and target domains in transfer learning tasks.

Multi-source Domain Adaptation (MDA) is a powerful extension to Domain Adaptation (DA) that can collect labeled data from multiple sources with different distributions. With the success of DA methods and the widespread use of multi-source data, MDA has gained growing attention from both academia and industry. [17] proposed TPTL based on TCA+ to automatically select the two source items that have the closest match with the target domain distribution. After that, the two prediction models are constructed separately, and their predictions are combined to improve the prediction performance further. A Multi-Source Adaptive Network (MSAN) based on multiple GAN architectures [2] can effectively learn the bidirectional transfer between the source and target domains, thus reducing the distribution differences. A joint feature space is also introduced to guide the multi-level consistency constraint of all transformations to maintain consistent domain patterns during the adaptive process and simultaneously empower the recognition of unlabeled target samples. To address the cross-project defect prediction task, [1] proposed 3SW-MSTL, which did exploratory work in three directions, namely, the number of source domains, source domain instance weights, and multi-source data utilization scheme using conditional distribution information.

These ideas still have room for improvement in the RL return prediction task. When the data of the target domain is sparse, the data sparsity problem can be alleviated by using data from multiple source domains. However, the above model ignores mutual time series features among source domains in the design of multi-source domain predictors aggregation, which can be affected by negative transfer resulting in suboptimal performance. To this end, we propose DT-MUSA to solve the RL return prediction task. The model DT-MUSA will be introduced in the following section.

3 Methodology

3.1 Overview

Faced with the task of predicting RL return in the target domain, we are faced with the following challenges: (1) Difficulty in the aggregation of multi-source domain predictors leads to a negative transfer effect. Leveraging knowledge from multiple sources can improve prediction accuracy. However, minimizing knowledge conflicts and distribution differences between multiple source domains and the target domain presents a significant challenge in domain adaptation, which can adversely affect the transfer effect. What kind of source domain aggregation strategy can we adopt to avoid negative transfer effectively? (2) Long-term data scarcity leads to suboptimal fine-tuning of model transfer. The multi-time-window input structure is crucial to fully extract the regular features of the RL return time series. However, the data scarcity in the target domain leads to the lack of trend data input to the model, which makes the transfer ineffective. How can we improve the neural network structure or refine the model inputs to improve the model transfer effect?

To address the above challenges, we propose DT-MUSA as shown in Fig. 1, which consists of a multi-source domain adaptation module multi-task learning based, and a dual transfer module based on the feature extraction network MUCAN.

- Multi-source domain adaptation algorithm based on multi-task learning.
  - Possible ways to select the appropriate source domain for domain adaptation: The source domains are given priority by estimating the similarity between the data distribution of each source domain and that of the target domain. After that, the model is pre-trained on the source domains within the priority threshold to overcome the distribution differences between source and target domains.
  - Multi-task learning based multi-source predictor aggregation: Pre-training is performed in the source domain set using multi-task learning, with knowledge shared among source domains. The model adaptively

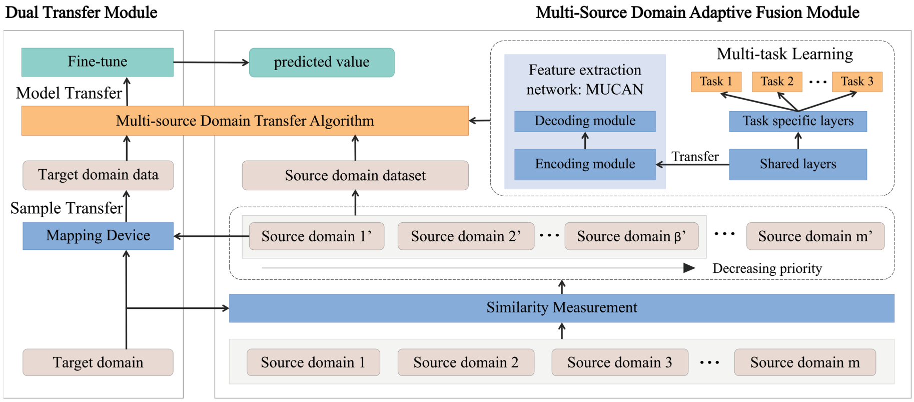

Fig. 1. Overall architecture of the DT-MUSA. Given m source domains with abundant historical data and current target domain historical data for prediction of future RL in the target domain, the DT-MUSA consists of a multi-source domain adaptation module based on multi-task learning and a dual transfer module based on feature extraction network MUCAN.

adjusts the weight distribution among the tasks to continuously learn the public knowledge, which is updated and retained within the shared layer to increase the amount of positive transfer knowledge.

– Dual transfer model based on feature extraction network MUCAN.
  - The base model MUCAN for transfer learning: A multi-scale time-series feature extraction network MUCAN is built specifically for RL regression prediction by combining a convolutional attention module and a multi-time scale window structure. MUCAN can accommodate the efficient transfer of source domain knowledge and the complete extraction of target domain features in model transfer.
  - Dual transfer algorithm combining sample transfer and model transfer: The data in the target domain are set as the direction of sample transfer, and the mapper is trained to fill the missing trend window data in the target domain. Model transfer with fine-tuned parameters is subsequently employed to complete the transfer of public knowledge and enhance the model's generalization ability.

3.2 Possible Ways to Select the Appropriate Source Domain for Domain Adaptation

The data features learned in the source domain can be effectively transferred to the target domain when the sufficient distributional similarity between the source and target domain data is satisfied. Otherwise, the source domain will transfer more knowledge carrying negative effects to the target domain, which will bring some adverse effects on transfer learning. Therefore, we introduce the domain similarity estimation algorithm to initially screen the source domains with high similarity to the target to avoid negative transfer.

We introduced the Jensen-Shannon (JS) divergence based on the Kullback-Leibler (KL) divergence as a measure of domain similarity estimation. The smaller the JS divergence of the source and target domains, the greater the similarity. The similarity is expressed as

$$KL\left(p||q\right) = \sum p\, log\frac{p}{q}, \tag{1}$$

$$JS(P_x||P_y) = \frac{1}{2}KL(P_x||\frac{P_x+P_y}{2}) + \frac{1}{2}KL(P_y||\frac{P_x+P_y}{2}). \tag{2}$$

After calculating the similarity between the distribution of the target domain P_y and that of the source domain P_x, we assign all source domains within the source domain set X priority rank $f_{x_i}(f_{x_i} \leq |X|, f_{x_i} \in Z^+)$. The higher the similarity between the source and the target domains, the smaller the value of the corresponding source domain priority rank, when $f_{x_i} = 1$ indicates that x_i is the source domain with the highest similarity. A reasonable parameter priority threshold β is also determined, i.e., the subsequent pre-training of the model is performed on the source domain $x_i(f_{x_i} \leq \beta, x_i \in X)$ only. We consider the data distribution of source domains with priority rank below the threshold to be highly similar to the target domain and use them as alternate source domains, thus enhancing the proportion of positive transfer in the transfer learning process.

3.3 Multi-task Learning Based Multi-source Predictor Aggregation

To make the model fully extract and aggregate the public knowledge of each source domain when pre-training in the source domain set, we define pre-training on different source domains as different tasks and use hard parameter sharing to associate these tasks. By sharing knowledge, model components complement each other and enhance the effectiveness of mining time series public features.

Figure 2 illustrates the architecture of the pre-training model utilizing multi-task learning. Based on MUCAN (we will cover this in detail in Sect. 3.4), we will slightly change the model structure by using multiple decoders in parallel to replace the original MLP network used for single-task decoding to obtain Muti-MUCAN. Each decoder acts as a private module for different pre-training tasks to independently decode the fusion codes extracted from the shared layer. In the face of multiple decoding outputs of multiple tasks, the output results of different tasks are calculated to obtain different *loss* magnitudes, and the task with larger *loss* may dominate the model optimization direction. To overcome this problem, we constitute the total *loss* by calculating the weighted *loss* of different tasks as follows:

$$L(t) = \sum\nolimits_{i=1}^{\beta} w_i(t)L_i(t) \tag{3}$$

where t denotes the current training step number, and w_i denotes the weight of different task *losses*. For w_i, we are not sure that we can manually set the weights quite appropriately, so it is wiser to choose a method that can dynamically adjust

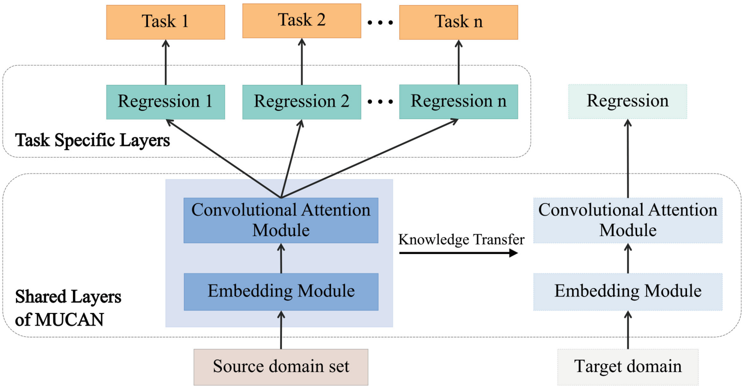

Fig. 2. A pre-trained model utilizing multi-task learning.

the weights according to the learning effect of different tasks. Dynamic Weight Average [18] reflects the learning difficulty by considering the rate of change of each task's *loss* and thus dynamically calculates the weights of the tasks.

$$w_i(t) := \frac{S\ \exp(r_i(t-1)/T)}{\sum_{i=1}^{\beta} \exp(r_i(t-1)/T)}, \quad r_i(t-1) = \frac{L_i(t-1)}{L_i(t-2)} \tag{4}$$

$r_i(\cdot)$ denotes the relative rate of decline of *loss*, S is used to limit the range of variation of weights to satisfy $\sum_i w_i(t) = S$, and T denotes the degree of relaxation between individual tasks, if the larger the value, the more the weights of each task tend to be equal.

3.4 The Base Model MUCAN for Transfer Learning

To solve the prediction task for regions with sparse samples, we introduced a convolutional attention module based on the structure of multi-time scale window inputs [40] and built a feature extraction network MUCAN for the subsequent transfer task. The extraction of features at different time scales of the series is achieved to capture the dependence of WEEE at multiple time scales.

First, we label the closeness data window, period data window, and trend data window in the historical data as cw, pw, and tw, respectively, and the size of each window is c_{len}, p_{len}, and t_{len}. Based on the current time step s, the size of each window in the historical data is defined as follows.

– Closeness window: $cw = [s - c_{len}, s)$
– Period window: $pw = [s - 30 - \frac{p_{len}}{2}, s - 30 + \frac{p_{len}}{2})$
– Trend window: $tw = [s - 365 - \frac{t_{len}}{2}, s - 365 + \frac{t_{len}}{2})$

We extract cw, pw, and tw from the historical data as inputs to capture the temporal dependence of sequences at multiple time scales. Then the three input features are encoded by the corresponding LSTM modules to mine the correlation between the sequence adjacent data within each window to generate three coded sequences with the same dimension c^{enc}, p^{enc}, and t^{enc}. We use the convolutional attention module to weigh the three coded sequences for fusion to obtain the fusion coding m^{enc}. Finally, the fusion coding is decoded by a multilayer neural network as a regression layer to obtain the final prediction. The structure of our designed network model is shown in Fig. 3.

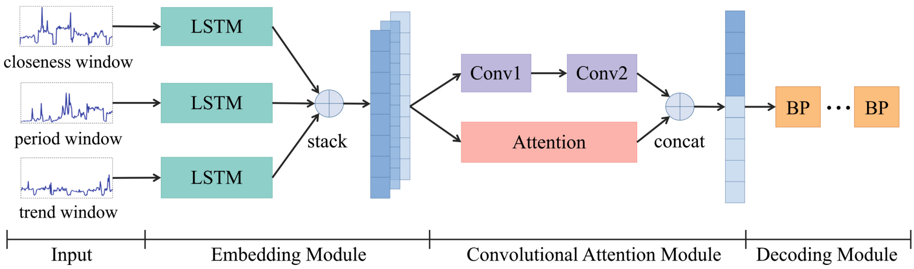

Fig. 3. Structure of feature extraction network MUCAN for model transfer.

The characteristics and design motivation of the MUCAN extraction layer are described as follows:

First is the LSTM-based coding module. Due to the unique gating unit and memory unit of LSTM, it avoids the gradient disappearance problem of RNN in the training process of long sequences and is good at extracting long-term time series features. We adopt LSTM as the encoding module for the input window to extract the characteristic laws of sequences in trend, period, and closeness, respectively. Our model encodes the window input data x at different time scales by LSTM to obtain dimensionally consistent encoding sequence x^{enc}.

$$x^{enc} = LSTM(x) \tag{5}$$

We generate 3 coding sequences: c^{enc}, p^{enc}, and t^{enc}, after inputting the window data cw, pw, and tw into their respective LSTM modules.

Second is the convolutional attention module. In exploring the three dependencies of sequences, we consider the mining of sequence trendiness and periodic regularity features to act as a supporting role. To focus the model's attention more on the closeness window encoding sequences c^{enc}, we use an additive attention mechanism [36] as part of the fusion encoding. Based on the prediction task of the time series, we define the encoding of the closeness window c^{enc} as a query vector q and set the key values $K = \{c^{enc}, p^{enc}, t^{enc}\}$. The scoring function for the additive attention mechanism is as follows:

$$s_i = s(k_i, q) = \varepsilon^T tanh(W_k k_i + W_q q) \tag{6}$$

where W_k, W_q, and ε are all learnable parameters. The attentional scoring s_i is obtained by this formula, which is subsequently softmax to obtain the attention distribution α_i. At this point we weight the coded input $V = [c^{enc}, p^{enc}, t^{enc}]$ with the attention distribution coefficients α_i to obtain the output m^{att}:

$$\alpha_i = softmax(s_i) = \frac{\exp(s_i)}{\sum_{j=1}^{len(K)} s_j} \tag{7}$$

$$m^{att} = \sum_{i=1}^{3} \alpha_i v_i \ , \ v_i \in V \tag{8}$$

The annual trend, monthly period, and closeness law features on which the WEEE data depend do not change easily. Considering the effectiveness of the convolution operation in extracting spatio-temporal features, we use the convolution module as another part of the fusion coding, where the convolutional kernel size of $conv1$ and $conv2$ is 3, the padding of $conv1$ is 1, and the padding of $conv2$ is 0. The convolutional coding m^{conv} is obtained as follows:

$$m^{conv} = conv1(conv2(stack(v_1, v_2, v_3))) \tag{9}$$

After deriving the attentional and convolutional encoding, the two are concatenated together to obtain the fusion encoding m^{enc}.

$$m^{enc} = concat(m^{att}, m^{conv}) \tag{10}$$

Finally, the decoding module. We use a multilayer neural network (MLP) to decode the fusion code m^{enc} and output the predicted values for the next t days. In addition, a dropout layer is added to the multilayer neural network to avoid the overfitting phenomenon.

3.5 Dual Transfer Algorithm Combining Sample Transfer and Model Transfer

Since the target domain with missing historical data generally only provides order data within the last three months, applying the model will result in missing trend window input tw for the feature extraction network MUCAN. This means that it will be difficult for MUCAN to model the trend, and the dependence of the series on the trend time scale will not be explored.

To solve this problem, we first select the source domain $x_1(f_{x_1} = 1)$ with the highest priority in the source domain set and define the trending data domains used in the source domain x and the target domain y as x^{tre} and y^{tre}, respectively. We learn the mapping rules from the sample data in the source domain x_1 to the sample data in the target domain y by training a mapper (see Fig. 4), and subsequently use x^{tre} as the input to the mapper to generate the missing trend data in the target domain y^{tre}. To simplify the complexity of the overall model, we use LSTM-MLP to learn the mapping rules for sample transfer.

After completing the trend window of the target domain, to minimize the loss of public knowledge in the transfer learning process between the source and target domains as much as possible, we set a lower learning rate for the part of the convolutional attention layer and the part of the LSTM layer in the

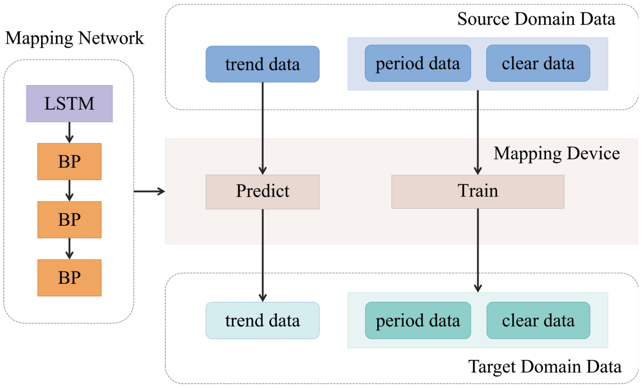

Fig. 4. Mapper structure for sample transfer.

network structure to achieve the effect of parameter fine-tuning while adapting the parameters of the regression layer to the distribution of the target domain data by retraining.

4 Experiment Setup

To assess the model performance, we conducted several experiments, including overall performance comparison, ablation experiments, and sensitivity analysis. In this section, we will provide details on the datasets utilized in the experiments, the selection of baseline methods, evaluation metrics, and other relevant experimental details.

4.1 Dataset

The commercial company Able Green provided the dataset used to test the model's predictive performance. From the company's WEEE recycling service order data for the last two years, two types of time-series data are available from 28 Chinese provinces: air conditioners (AC) and washing machines (WM). Some provinces are late to start recycling services and are missing long-term data, and our experiments will focus on predicting two types of data for AC and WM in such provinces. Detailed statistical data is shown in Table 1, where we distinguish the provinces with scarce data from those with sufficient data according to the earliest correct order time in each province and divide them into the training, validation, and test sets according to 7:1.5:1.5.

4.2 Baselines

– ARIMA [31]: A method combines an autoregressive (AR) model with a moving average (MA) model and a differential preprocessing step of the series to smooth the series.

Table 1. The statistics of datasets

Domain	Types	Provinces covered	Period begin	Period end
Source	AC	25	1/1/2018	12/31/2019
	WM	25	1/1/2018	12/31/2019
Target	AC	3	8/30/2019	12/31/2019
	WM	3	8/30/2019	12/31/2019

- LSTM [8]: A special type of RNN with special gated memory units that are good at extracting long-term time series features.
- Autoformer [37]: Based on Transformer, a model that enables efficient connection at the sequence level for better information aggregation.
- Informer [41]: Based on Transformer, a sparse attention mechanism is incorporated to reduce the network complexity.
- MULAN [40]: The model introduces multi-time scale windows and attention-based alignment fusion, which can capture the temporal dependence of sequences at multiple timescales.

4.3 Evaluation Metrics

To evaluate the performance of DT-MUSA and other models, we need to use appropriate evaluation metrics to measure the prediction accuracy of the corresponding algorithms. MAE and RMSE are the evaluation metrics employed in this study to assess the model's prediction performance. Lower values of these metrics indicate better performance by the model. The MAE and RMSE can be calculated as follows:

$$MAE = \frac{1}{t}\sum_{i=1}^{t}|pre_i - tru_i|, \tag{11}$$

$$RMSE = \sqrt{\frac{1}{t}\sum_{i=1}^{t}(pre_i - tru_i)^2} \tag{12}$$

where t denotes the step size that the model will predict, and pre_i and tru_i denote the predicted and true values of the day, respectively.

4.4 Experimental Details

By grid search, we set the three input window lengths c_{len}, p_{len}, and t_{len} of the feature extraction network MUCAN to 15, 10, and 20, respectively, and determined $\beta = 3$, $lr = 0.001$ and $epoch = 50$ rounds in the source domain and

$lr = 0.01$ and $epoch = 100$ rounds in the target domain. The best-performing model on the validation set was retained, and all RMSEs and MAEs were calculated on the test set. All experiments were performed in the PyTorch framework [27].

5 Results and Analyses

5.1 Predictive Performance Comparison

To demonstrate the superiority of DT-MUSA in solving the task of predicting RL return data in regions with sparse samples, we compare the results with other commonly used prediction models as shown in Table 2 and 3.

Table 2. Overal performance comparison (MAE) on the AC and WM datasets.

MAE	Jiangsu Province		Chongqing City		Peking City	
	AC	WM	AC	WM	AC	WM
ARIMA	25.659	16.888	1.360	4.103	0.652	1.032
LSTM	30.344	25.231	4.311	6.364	1.366	3.215
Informer	27.926	18.480	3.379	7.099	1.182	1.863
Autoformer	24.264	19.826	2.852	3.976	1.366	3.215
MULAN	20.386	<u>8.854</u>	1.838	3.637	1.508	1.505
MUCAN	<u>9.373</u>	12.586	<u>0.736</u>	<u>1.675</u>	<u>0.307</u>	<u>0.719</u>
MUCAN-sbs*	5.432	8.268	0.589	1.826	0.238	0.682
MUCAN-rt*	17.283	15.383	1.116	2.926	0.422	0.823
DT-MUSA*	**2.775**	**3.471**	**0.233**	**1.436**	**0.098**	**0.310**

Notes: The models with * in Tables 2 and 3 indicate the transfer learning models, while the rest indicate the non-transfer models. Bold in Table 2 indicates the minimum value of MAE obtained for all models involved in the comparison in the corresponding data set; the sliding line represents the minimum value of MAE obtained for the non-transfer learning models involved in the comparison.

Under the condition of no transfer, the models are highly susceptible to underfitting because the short-term data are difficult to meet the training requirements of most deep learning models. The comparison of the transfer-free methods in Tables 2 and 3 shows that various classical prediction models perform unsatisfactorily and are poorly adapted to the prediction task in regions with sparse samples. One of the statistical methods, ARIMA, achieves a relatively good result but is still objectively less than ideal. The feature extraction

Table 3. Overal performance comparison (RMSE) on the AC and WM datasets.

RMSE	Jiangsu Province		Chongqing City		Peking City	
	AC	WM	AC	WM	AC	WM
ARIMA	26.939	17.307	1.445	4.232	0.782	1.155
LSTM	27.753	21.755	3.571	2.939	1.573	3.879
Informer	30.278	20.288	2.962	3.092	1.279	2.379
Autoformer	28.738	20.896	3.725	5.527	0.942	2.072
MULAN	25.393	<u>11.665</u>	2.476	7.379	0.805	<u>0.644</u>
MUCAN	<u>10.981</u>	15.869	<u>0.861</u>	<u>2.333</u>	<u>0.356</u>	0.929
MUCAN-sbs*	8.327	9.236	0.627	1.923	0.188	0.572
MUCAN-rt*	20.238	13.378	1.630	4.132	0.283	0.592
DT-MUSA*	**4.801**	**4.610**	**0.327**	**1.553**	**0.112**	**0.418**

NotesNotes: Bold in Table 3 indicates the lowest RMSE value achieved across all models involved in the comparison in the corresponding data set; the sliding line represents the lowest value of RMSE obtained for the non-transfer learning models involved in the comparison.

network MUCAN achieves excellent results in the target domain by capturing the correlation patterns of the series on various time scales. However, due to the lack of sample data, the dependencies of sequences on various time scales cannot be explored more fully, resulting in the inability to predict some inflection points in the prediction task accurately.

When a large amount of source domain data is available for transfer, we compare DT-MUSA with some simple source domain selection strategies. As shown in Tables 2 and 3 for the comparison of transfer methods, DT-MUSA performs significantly better than single source domain transfer as well as random source domain transfer for the target domain prediction task. This demonstrates that a simple source domain selection strategy may introduce unsuitable source domain data, leading to negative transfer. Compared with MUCAN without transfer, DT-MUSA also has significant performance improvement. This proves that DT-MUSA not only overcomes the distribution difference between the source domain data and the target domain data to a certain extent but also solves the negative impact due to the missing data of the target domain. Overall, DT-MUSA performs adaptive aggregation of predictors from multiple source domains, effectively reducing negative transfer.

5.2 Strength and Weakness

Comparing the baselines in the table, our advantages are as follows:

– Pre-training in the source domain set by multi-task learning aggregates the common knowledge of multiple source domains to transfer to the target domain, which avoids negative transfer to a certain extent.

- For WEEE return data, we build a feature extraction network MUCAN based on the convolutional attention module, which fully explores the dependencies of sequences at multiple time scales and helps to obtain more accurate prediction results in transfer learning.
- Sample transfer is used to complement the trend window data input of MUCAN on the target domain to expand the range of explorable time-scale categories and enhance the effect of model transfer.

In terms of model shortcomings: our model initially screens the source domains by estimating the similarity between the data of each source domain as a whole and the data of the target domain and thus divides the priorities to initially screen the source domains in the expectation of reducing the data distribution differences from the source to the target domains. However, the instance adaptive approach has its limitations in the end, and the distribution differences between the source and target domains will not be completely eliminated. Suppose the deep feature adaptive approach is further explored, and the embedding of an adaptive module to encode the mapping of source domain data is considered. In that case, reducing the negative transfer phenomenon leads to an improvement in model prediction accuracy.

5.3 Ablation Experiments

In our experiments, (1) we introduced a model transfer approach to pre-train the model with shared parameters in the multi-source domain, followed by fine-tuning in the target domain; (2) we transferred data from the rich data domain to the poor data domain by mapping the model, which complements the trend window data input of MUCAN; (3) we built a feature extraction network MUCAN based on the Convolutional Attention Module for weighted fusing data from different time scales for decoding output. We designed the following ablation experiments to demonstrate the usefulness of each of the above three components.

- Validate the effect of fine-tuning.
  - DT-MUSA-ft: Remove the fine-tuning in the target domain, and apply the pre-trained model directly to the target domain.
- Validate the effect of sample transfer.
  - DT-MUSA-st: The closeness window data of the target domain is used to fill the trend window data of the target domain.
- Verify the effect of Convolutional Attention Module on data fusion at different time scales
  - DT-MUSA-at: Remove the Attention Module and use CNN-LSTM for training.
  - DT-MUSA-ac: Remove the Attention and CNN Modules, and use LSTM for training.

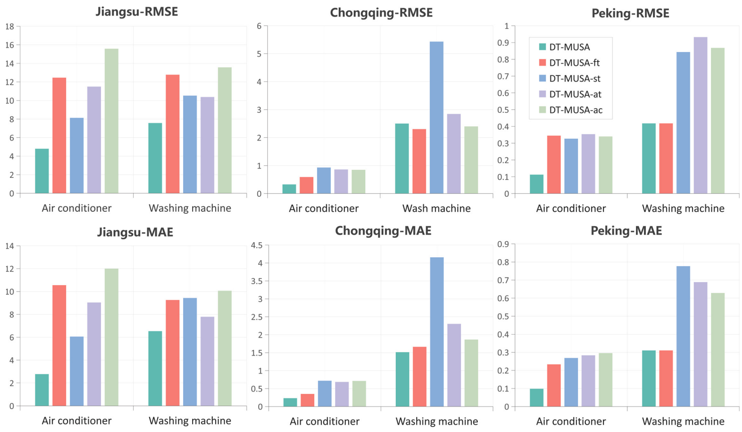

Fig. 5. The effect of each part on model performance.

From the experimental results in Fig. 5, it can be seen that these three components play an extremely important role in the excellent overall model results. (1) The model transfer based on the pre-training fine-tuning approach contributes significantly to the overall model performance. While the pre-training model is effective at extracting complex temporal feature information in data-rich domains, it can only provide a rough reflection of the trend characteristics of the data when carrying the public knowledge from the source domain. Without fine-tuning, it cannot make accurate predictions due to its neglect of the distributional differences between the target and source domain data. (2) Sample transfer based on a mapper trained on a rich set of samples is essential in this model. Previously, MULAN [40] has demonstrated that the window design based on trending scales can effectively improve prediction accuracy. Due to the lack of trend window data in the target domain, we complemented the trend window data input in the poor data domain by the LSTM-MLP based mapping model and achieved good results. (3) The embedding of the convolutional attention module is also shown to effectively improve the model's prediction performance, especially in scenarios with transfer learning, allowing the model to gain more room for improvement.

5.4 Model Sensitivity Analysis

The priority rank associated with the domain similarity size is introduced in the source domain set used for the model transfer. The model is pre-trained for multi-task association learning using only multiple source domains within the priority threshold β. Our model should be robust to moderate changes in β of the source domain set. To this end, we conducted sensitivity experiments to verify that the model performs consistently over various variations in the source domain set priority threshold parameter. When $\beta = 1$, the model degenerates to single source domain transfer, so we make β vary in the integer interval $[2:7]$ and compare the model predictions with the results of the transfer models ($\beta = 3$) in Sect. 5.1.

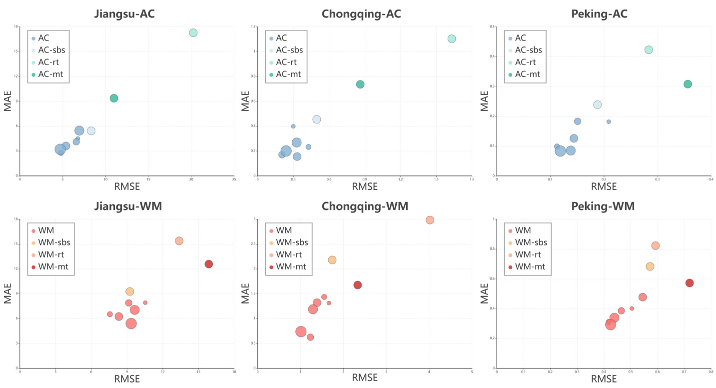

Fig. 6. The effect of changes in the source domain priority threshold β on model performance.

The comparison results are shown in Fig. 6, where (AC/WM)-sbs indicates the experimental results of the best single-source domain transfer on the air conditioner or washing machine dataset; (AC/WM)-rt indicates the random source domain transfer results; (AC/WM)-mt indicates the prediction results of MUCAN without transfer, and the bubble size of (AC/WM) type corresponds to the number of source domains used by DT-MUSA. It is easy to see that DT-MUSA effectively improves the accuracy of the RL prediction task, and the model's prediction performance fluctuates within an acceptable range. And the model generally outperforms other source domain adaptation strategies overall within a certain variation of the priority threshold β. This demonstrates that our model's performance is stable and less affected by variations in the priority threshold parameter.

6 Discussion and Conclusion

In this paper, we have extensively investigated the predictive effectiveness of various excellent models in the presence of sparse RL return data for WEEE. For multi-source domain adaptive models, we point out the challenges of their current application in practical scenarios: inefficient use of multi-source data in multi-source knowledge transfer learning tasks; lack of long-term time-series data leading to suboptimal fine-tuning of model transfer.

We thus propose DT-MUSA to address the above challenges. We first try to use multi-task learning in the pre-training of multiple source domains, which effectively resolves the knowledge conflicts among source domains and dramatically helps the fusion of source domain public knowledge. To overcome the adverse effects of long-term data scarcity, DT-MUSA performs model transfer from multiple source domains to target domains based on MUCAN neural networks and complements the input data of trend windows by training mappers.

DT-MUSA focuses on the WEEE prediction task for certain provinces with late start-up recycling services, where order information is typically only collected for the past three months. We applied the model to a real Able Green recycling business case and conducted a full experiment. The results show that the model significantly improves the prediction accuracy on prediction tasks in the target domain and is robust to critical parameters, showing that our model is practical and feasible in real-world applications.

In this study, the method of similarity estimation to screen the source domain cannot completely overcome the difference in the distribution of source and target domains. Adding an adaptive layer to the network model structure of the source and target domains may improve the transfer effect. Besides, there may be connections between the recycling data of different kinds of WEEE, and using these connections to build a multi-task learning model may achieve better prediction results.

Acknowledgments. This study was supported by the National Key Research and Development Program of China (2020YFB1712901) and the Science and Technology Research Project of Chongqing Education Commission (KJZD-K202204402).

References

1. Bai, J., Jia, J., Capretz, L.F.: A three-stage transfer learning framework for multi-source cross-project software defect prediction. Inf. Softw. Technol. **150**, 106985 (2022)
2. Chen, C., Xie, W., Wen, Y., Huang, Y., Ding, X.: Multiple-source domain adaptation with generative adversarial nets. Knowl.-Based Syst. **199**, 105962 (2020)
3. De Bois, M., El Yacoubi, M.A., Ammi, M.: Adversarial multi-source transfer learning in healthcare: application to glucose prediction for diabetic people. Comput. Methods Programs Biomed. **199**, 105874 (2021)
4. Dev, N.K., Shankar, R., Qaiser, F.H.: Industry 4.0 and circular economy: Operational excellence for sustainable reverse supply chain performance. Resour. Conserv. Recycl. **153**, 104583 (2020)
5. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**(1), 2030–2096 (2016)
6. Garg, S., Kumar, S., Muhuri, P.K.: A novel approach for covid-19 infection forecasting based on multi-source deep transfer learning. Comput. Biol. Med. **149**, 105915 (2022)
7. Govindan, K., Bouzon, M.: From a literature review to a multi-perspective framework for reverse logistics barriers and drivers. J. Clean. Prod. **187**, 318–337 (2018)
8. Graves, A.: Long short-term memory. Supervised sequence labelling with recurrent neural networks, pp. 37–45 (2012)
9. Gu, Q., Dai, Q., Yu, H., Ye, R.: Integrating multi-source transfer learning, active learning and metric learning paradigms for time series prediction. Appl. Soft Comput. **109**, 107583 (2021)

10. Gupta, P., Malhotra, P., Vig, L., Shroff, G.: Transfer learning for clinical time series analysis using recurrent neural networks. arXiv preprint arXiv:1807.01705 (2018)
11. Hasan, M.M., Nekmahmud, M., Yajuan, L., Patwary, M.A.: Green business value chain: a systematic review. Sustain. Prod. Consumption **20**, 326–339 (2019)
12. Huang, B., Xu, T., Li, J., Shen, Z., Chen, Y.: Transfer learning-based discriminative correlation filter for visual tracking. Pattern Recogn. **100**, 107157 (2020)
13. Ismail, H., Hanafiah, M.M.: Evaluation of e-waste management systems in malaysia using life cycle assessment and material flow analysis. J. Clean. Prod. **308**, 127358 (2021)
14. Jabbour, C.J.C., de Sousa Jabbour, A.B.L., Govindan, K., Teixeira, A.A., de Souza Freitas, W.R.: Environmental management and operational performance in automotive companies in brazil: the role of human resource management and lean manufacturing. J. Clean. Prod. **47**, 129–140 (2013)
15. Kelle, P., Silver, E.A.: Forecasting the returns of reusable containers. J. Oper. Manag. **8**(1), 17–35 (1989)
16. Kilic, H.S., Cebeci, U., Ayhan, M.B.: Reverse logistics system design for the waste of electrical and electronic equipment (WEEE) in turkey. Resour. Conserv. Recycl. **95**, 120–132 (2015)
17. Liu, C., Yang, D., Xia, X., Yan, M., Zhang, X.: A two-phase transfer learning model for cross-project defect prediction. Inf. Softw. Technol. **107**, 125–136 (2019)
18. Liu, S., Johns, E., Davison, A.J.: End-to-end multi-task learning with attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1871–1880 (2019)
19. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: International Conference on Machine Learning, pp. 97–105. PMLR (2015)
20. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
21. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: International Conference on Machine Learning, pp. 2208–2217. PMLR (2017)
22. Ma, Y., Luo, G., Zeng, X., Chen, A.: Transfer learning for cross-company software defect prediction. Inf. Softw. Technol. **54**(3), 248–256 (2012)
23. Marx-Gomez, J., Rautenstrauch, C., Nürnberger, A., Kruse, R.: Neuro-fuzzy approach to forecast returns of scrapped products to recycling and remanufacturing. Knowl.-Based Syst. **15**(1–2), 119–128 (2002)
24. Nam, J., Pan, S.J., Kim, S.: Transfer defect learning. In: 2013 35th International Conference on Software Engineering (ICSE), pp. 382–391. IEEE (2013)
25. Ni, Z., Chan, H.K., Tan, Z.: Systematic literature review of reverse logistics for e-waste: overview, analysis, and future research agenda. Int. J. Logist. Res. Appl. **26**, 1–29 (2021)
26. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE Trans. Neural Networks **22**(2), 199–210 (2010)
27. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
28. Peng, L., et al.: TLT: recurrent fine-tuning transfer learning for water quality long-term prediction. Water Res. **225**, 119171 (2022)
29. Rocha, T.B., Penteado, C.S.G.: Life cycle assessment of a small WEEE reverse logistics system: Case study in the campinas area, brazil. J. Clean. Prod. **314**, 128092 (2021)

30. Sepúlveda, A., et al.: A review of the environmental fate and effects of hazardous substances released from electrical and electronic equipments during recycling: Examples from china and India. Environ. Impact Assess. Rev. **30**(1), 28–41 (2010)
31. Siami-Namini, S., Tavakoli, N., Namin, A.S.: A comparison of ARIMA and LSTM in forecasting time series. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1394–1401. IEEE (2018)
32. Song, Q., Li, J.: A review on human health consequences of metals exposure to e-waste in china. Environ. Pollut. **196**, 450–461 (2015)
33. Sun, S., Shi, H., Wu, Y.: A survey of multi-source domain adaptation. Inf. Fusion **24**, 84–92 (2015)
34. Toktay, L.B., Wein, L.M., Zenios, S.A.: Inventory management of remanufacturable products. Manage. Sci. **46**(11), 1412–1426 (2000)
35. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014)
36. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
37. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: decomposition transformers with auto-correlation for long-term series forecasting. Adv. Neural. Inf. Process. Syst. **34**, 22419–22430 (2021)
38. Yang, Y., Williams, E.: Logistic model-based forecast of sales and generation of obsolete computers in the us. Technol. Forecast. Soc. Chang. **76**(8), 1105–1114 (2009)
39. Ye, R., Dai, Q.: A novel transfer learning framework for time series forecasting. Knowl.-Based Syst. **156**, 74–99 (2018)
40. Zhang, J., et al.: Multi-time scale attention network for WEEE reverse logistics return prediction. Expert Syst. Appl. **211**, 118610 (2023)
41. Zhou, H., et al.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 11106–11115 (2021)