Tsinghua Science and Technology
Doi: 10.26599/TST.2026.9010055

Research Article

MSDA4SD: A Framework for Sequential Data Analysis Based on Multi-Source Domain Adaptation

Zujun Liu†(✉), Mengfan Zhao†, Ying Xing, Zan Zhou, Fei Gao and Yiliang Lai

Abstract Sequential data plays a pivotal role in various real-world applications, such as traffic flow and source code, and is characterized by its strong temporal dependencies. Although deep neural networks (DNNs) achieve promising performance in sequential data analysis, they usually require large amounts of labeled data. Multi-source domain adaptation (MSDA) alleviates this issue by transferring knowledge from multiple source domains to the target domain. However, existing MSDA methods often ignore temporal dependencies and adaptive domain relevance in sequential data. To address these limitations, this paper proposes MSDA4SD, a multi-source domain adaptation framework based on long short-term memory (LSTM) networks and a domain-correlation attention-based weighted maximum mean discrepancy (WMMD) mechanism. MSDA4SD captures temporal features while adaptively aligning source and target domain distributions to improve robustness and generalization. Experiments on vulnerability multi-classification and traffic flow prediction datasets demonstrate that MSDA4SD consistently outperforms existing MSDA methods. Specifically, MSDA4SD improves the average Accuracy by 0.3%–0.6% in vulnerability multi-classification tasks, reduces MAE by 10.2%–19.4%, and decreases RMSE by 8.7%–16.9% in traffic flow prediction tasks.

Keywords Sequential Data, Multi-source Domain Adaptation, Vulnerability Multi-classification, Traffic Flow Prediction

• Zujun Liu is with Smart Steps Digital Technology Company Limited, Beijing, 100033, China. E-mail: liuzj@smartsteps.com.
• Mengfan Zhao, Ying Xing are with School of Intelligent Engineering and Automation, Beijing University of Posts and Telecommunications, Beijing, 100876, China. E-mail: zmf@bupt.edu.cn; xingying@bupt.edu.cn.
• Zan Zhou is with the School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing, 100876, China. E-mail: zan.zhou@bupt.edu.cn.
• Fei Gao is with School of Information Science and Technology Tibet University, Lhasa, 850000, China. E-mail: 337679107@qq.com.
• Yiliang Lai is with Assumption University OF THAILAND, Bangkok, 10540, Thailand. E-mail: yilianglai9@gmail.com.
† These authors contributed equally to this work.
Manuscript received: 2026-03-23; revised: 2026-05-18; accepted: 2026-05-21

1 Introduction

Sequential data, comprising a series of elements arranged in a specific order, plays a pivotal role in various fields [1]. There are various types of sequential data, including time series (such as traffic flow data) [2–5] and text series (such as codes) [6], as shown in Figure 1, traffic flow data is numerical data, while codes are symbolic data. Each element in this sequence holds a distinct position, establishing a unique relationship or dependency with its neighboring elements [7]. This characteristic implies that the state or value of one element can significantly influence its successors [8]. The research and analysis of sequential data have attracted great attention, leading to the development of various technologies and models[9–12].

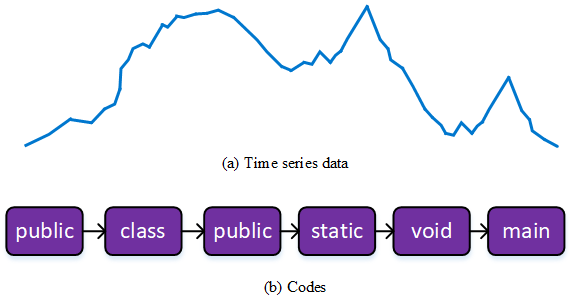

Fig. 1 Schematic diagram of sequential data

Traditional methods for modeling sequential data are difficult to capture the long-term dependencies of sequences. In recent years, deep neural networks (DNNs) have achieved tremendous success. DNN models such as convolutional neural networks (CNN) [13] and recurrent neural networks (RNN) [14] perform well in processing unstructured data, such as images, signals, speech, and text. More importantly, DNN has the ability to process sequential data. DNN can learn to extract features at different levels of abstraction, capture context dependencies and patterns in the data. LSTM can effectively capture temporal dependencies in sequential data and has shown strong performance in sequence modeling tasks [15].

However, although DNNs have shown remarkable performance in sequential data processing tasks, training them

requires a large amount of data, which can be difficult to obtain in some scenarios [16]. In practice, a single dataset may not be sufficient to achieve high accuracy, especially when the distribution of the target data differs significantly from the training data. To address this issue, multi-source domain adaptation (MSDA) has been proposed as an effective method to improve the performance of them by leveraging multiple datasets from different sources [17]. As a branch of domain adaptation [18], MSDA aims to learn a robust model that can effectively generalize across multiple domains even when the data distributions vary among these domains. Despite the effectiveness of existing MSDA methods, most of them are mainly designed for image-based tasks and pay limited attention to the temporal dependencies and contextual relationships in sequential data. Directly applying these methods to sequential scenarios may lead to insufficient feature representation and suboptimal domain alignment performance. In addition, different source domains contribute unequally to the target domain in practical sequential data applications, while conventional MSDA methods usually treat source domains equally.

Therefore, we propose a MSDA framework for sequential data analysis using long short-term memory (LSTM) networks, called MSDA4SD. MSDA4SD utilizes the ability of LSTM networks to capture long-term correlations in sequential data, while using attention-based MSDA method to enhance the robustness of the model to transfer domain and improve its generalization performance. Moreover, to evaluate the performance of MSDA4SD, we conduct experiments on two types of typical sequential data: software code data and traffic flow data. Software code data is often considered as sequential data with context dependencies, where the order of lines of code follows a specific structure and each line depends on the context provided by the preceding lines. On the other hand, traffic flow data can be viewed as a time series data that reflects the changes in traffic flow over time, with each time point being dependent on the traffic conditions at previous time points[19]. By conducting experiments on these two types of datasets, we demonstrate the effectiveness of MSDA4SD in capturing the context dependencies of sequential data, and its robustness to adapt to variations in data distribution across different domains.

2 Related Work

We select two main research directions to discuss, including sequential data analysis and multi-source domain adaptation (MSDA). Existing studies on sequential data mainly focus on spatio-temporal modeling and feature representation learning, while MSDA methods mainly optimize domain alignment and adaptive source-domain weighting.

Table 1 Comparison of representative sequential data analysis studies.

Representative Studies	Time-series	Textual
References[3,4,20–22]	✓	×
References[6,23,24]	×	✓

2.1 Sequential Data Analysis

Existing studies on sequential data mainly focus on spatio temporal dependency modeling and sequential feature representation learning to improve prediction and classification performance in different application scenarios, such as traffic flow prediction and source code analysis. As shown in Table 1, existing sequential data analysis studies are mainly designed for either time series or textual sequential data within specific domains.

2.1.1 Time-series Sequential Data

Time-series sequential data studies mainly focus on temporal dependency modeling and spatio-temporal feature extraction for dynamic prediction tasks. Alsarhan et al. [20] proposed PH-GCN to improve human action recognition through multi-level granularity representation and pair-wise hyper graph convolution, demonstrating the effectiveness of structured feature modeling for sequential scenarios. Ali et al. [3, 4] proposed dynamic multi-graph spatio-temporal learning methods for citywide traffic flow prediction and reviewed advanced deep learning and spatio-temporal feature extraction techniques for intelligent transportation systems. Ali et al. [21] proposed an attention-driven graph convolutional network to improve deadline-constrained virtual machine task allocation through adaptive relational feature aggregation. Cao et al. [22] proposed the Spatiotemporal-aware Trend-Seasonality Decomposition Network (STDN), a dynamic graph framework that disentangles trend and seasonal components to capture complex traffic dynamics.

2.1.2 Textual Sequential Data

Textual sequential data studies mainly focus on sequential semantic representation learning and contextual dependency modeling for source code analysis. Liu et al. [6] proposed TAILOR, which leverages Code Property Graphs (CPGs) combining abstract syntax trees and data flow to explicitly exploit deep graph-structured code features. He et al. [23] demonstrated that employing Bidirectional Long Short-Term Memory (Bi-LSTM) networks to learn sequential features from source code representations can significantly boost vulnerability detection accuracy. Sun et al. [24] proposed a

Table 2 Classification of representative MSDA studies.

Representative Studies	Feature-based	Attention-based
References[26–32]	✓	×
References[33–38]	×	✓

novel fine-grained sequence architecture that jointly learns token and statement representations to extract deep sequential semantics for software security.

Existing sequential data analysis studies mainly focus on either time-series or textual sequential data within specific domains. Cross-domain heterogeneous sequential adaptation remains insufficiently explored.

2.2 MSDA

As shown in Table 2, existing MSDA studies can be mainly categorized into feature-based and attention-based methods [25].

2.2.1 Feature-based MSDA

Zhao *et al.* [26] proposed a new generalization bound to learn domain invariant features. Liu *et al.* [27] aligns at the domain-level through adversarial learning and weights different source instances. Align at the class-level by minimizing the distance between the class prototype and the unlabeled target instance. Nguyen *et al.* [28] shared an encoder between the source domain and the target domain, adopted adversarial learning to train the encoder, and obtained domain experts for each source domain through training. Ren *et al.* [29] mapped each set of source and target domains into a group-specific subspace through adversarial learning with metric constraints, and constructed a series of pseudo-target domains accordingly. Then, the remaining source domain is effectively aligned with the pseudo-target domain in the subspace. Peng *et al.* [30] used BERT and bidirectional LSTM to extract domain invariant features and applied width learning to train classifiers. Taghiyarrenani *et al.* [31] constructed a shared feature space to generalize linear regression over all domains. Lu et al. [32] proposed a multisource domain adaptation method with distribution discrepancy constraints to improve feature discriminability across multiple domains through joint optimization of domain alignment and classification performance. Although these methods achieved promising performance in specific sequential tasks, feature-based methods mainly focused on reducing feature distribution discrepancies between source and target domains, which might weaken domain specific characteristics.

2.2.2 Attention-based MSDA

Dai *et al.* [33] proposed a Weighting Scheme based Unsupervised Domain Adaptation framework (WS-UDA) to obtain good target assumptions by combining source assumptions. Zhao *et al.* [34] simultaneously considers feature distance and feature similarity between the source and target. Zuo *et al.* [25] trained a domain recognition model to calculate the probability that the target image belongs to each source domain. On the basis of domain correlation, moment distance (MD) and classification loss are weighted to concentrate on the source domain with higher similarity. Peng *et al.* [35] dynamically aligns the MD of the feature distribution of the labeled source dataset, transferring knowledge to the unlabeled target dataset. Wang *et al.* [36] proposed a class aware sampling strategy that selects samples from each domain to achieve class level conditional alignment, assigns weights to each sample using sample reweighting, while considering classification reliability and spatial information correlation. Xu *et al.* [37] achieved effective feature transfer by dynamically aligning spatial and temporal feature matrices. Belal et al. [38] proposed an attention-based class-conditioned alignment method for multisource domain adaptation of object detectors by dynamically aligning category-specific features across multiple domains. Although attention-based methods considered the unequal contributions of different source domains, they might lose domain-invariant features and insufficiently model temporal dependencies in heterogeneous sequential data environments [25].

3 Methodology

The principle of MSDA is shown in Figure 2. Existing sequential data studies are mainly designed for either time series or textual sequential data within specific domains. To simultaneously handle heterogeneous sequential data from different domains, an end-to-end MSDA framework, named MSDA4SD, is proposed based on sequential feature modeling and attention-based domain alignment. In this paper, domain and dataset are interchangeable, and each dataset is regarded as a domain.

MSDA4SD includes two phases: training and prediction. In order to make our description precise, the symbols used in this paper are defined below. We assume that a group of M source datasets $s_1, s_2, \ldots, s_M$ and a target dataset T are given. To unify the sample size definitions, let n_{s_j} denote the number of samples in the j-th source dataset s_j, n_t denote the number of samples in the target dataset T, $n_s = \sum_{j=1}^{M} n_{s_j}$ denote the total number of labeled source data, and $n = n_s + n_t$ denote the total number of samples across all datasets. The j-th source dataset is denoted by $\{(X_i^{s_j}, Y_i^{s_j})\}_{i=1}^{n_{s_j}}$, where $X_i^{s_j}$ represents the i-th labeled sample and $Y_i^{s_j}$ represents the corresponding label space. The target dataset T is denoted as $\{X_i^t\}_{i=1}^{n_t}$,

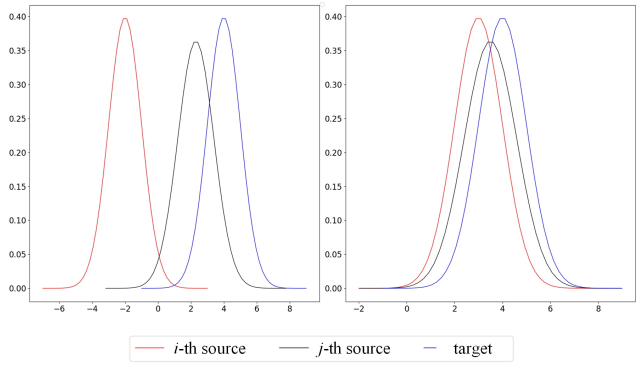

Fig. 2 The distribution of source and target domains before and after MSDA

where X_i^t is the i-th unlabeled sample (its corresponding unseen label space is denoted as Y^t). The target dataset T is divided into a validation set and a testing set in a ratio of 1:2, which are used for training and testing, respectively. In addition, we denote $\{(X_i, d_i)\}_{i=1}^n$ as the collection of all samples across domains with their corresponding domain labels d_i. Note that each source dataset s_j $(j \in [1, M])$ contains the same number of categories as the target dataset T, i.e., Y^t consists of the same number of categories as any source dataset Y^{s_j} $(j \in [1, M])$. Similar to single-source domain adaptation, Y^t is not available during training and is used only for evaluation. As shown in Figure 3, we take two source domains (s_i and s_j) and the target domain (T) as an example. The training phase involves three losses, namely, $\mathcal{L}_{dc}$, $\mathcal{L}_{dis}$, and $\mathcal{L}_c$. The source and target dataset samples are used as inputs to train the model to obtain a trained MSDA model.

Firstly, the domain encoder G and domain discriminator D are used to obtain $\mathcal{L}_{dc}$. The encoders G and E are two-layer bidirectional LSTM networks, and their encoding process is shown in Figure 4. Specifically, the domain encoder (G) extracts domain-correlation information to adaptively measure the relevance between the source and target domains. Concurrently, the embedding encoder (E) focuses on learning domain-invariant sequential feature representations for downstream classification and distribution alignment. D is a common classifier to classify the datasets of samples, and can be used to measure the similarity of data distribution between source dataset and target dataset, and $\mathcal{L}_{dc}$ is calculated by:

$$\mathcal{L}_{dc} = -\frac{1}{n}\sum_{i=1}^{n} d_i^{\top} \ln D\left(\mathrm{G}\left(X_i\right)\right) \tag{1}$$

where d_i is the domain label.

We use domain discriminator D to calculate the domain correlation between a batch of target samples and all source domains, that is, for N target samples, we use D to predict the probability of their domain categories. D outputs a probability matrix $P \in R^{N\times(M+1)}$ to represent the relationship between the target samples and the domain label.

After obtaining $p \in R^{N\times(M+1)}$, the first M terms in p which is related to M source datasets are selected as the domain correlation matrix $P_d \in R^{N\times M}$, and the domain correlation coefficient matrix $w = \{w_1, w_2, \ldots, w_M\}$ can be obtained by softmax:

$$w = softmax\left(P_d\right) \tag{2}$$

In order to make the distribution of a source dataset and the target dataset similar, the data distribution is aligned between the source and target dataset to mitigate the distribution gaps between datasets. In this paper, the maximum mean discrepancy (MMD) [39] is used to measure the similarity between two distributions. $\mathrm{MMD}_k(S, T)$ can measure the distance between the distribution of S and T. The square formula of MMD is defined as:

$$\mathrm{MMD}_k^2(S, T) = \left\| E_S\left[\sigma\left(X^s\right)\right] - E_T\left[\sigma\left(X^t\right)\right]\right\|^2 \tag{3}$$

where the feature mapping function σ is defined as the combination of m positive semi-definite kernels $k_u (u \in [1, m])$. $E[\cdot]$ denotes the mean operation, and the other variables follow the definitions introduced in Section 3.

Traditional MMD treats all source domains equally, which may introduce negative influence from unrelated source domains in heterogeneous multi-source scenarios. Combining the obtained domain correlation coefficient w produced by formula (2), the weighted MMD (WMMD) is proposed and we define the squared formula for WMMD as:

$$\begin{aligned}&\mathrm{WMMD}_k^2(S, T)\\ &= \sum_{j=1}^{M} w_j \left\| E_{X^{s_j}}\left[\sigma\left(X^{s_j}\right)\right] - E_{X^t}\left[\sigma\left(X^t\right)\right]\right\|^2\end{aligned} \tag{4}$$

where w_j depicts the $j-th$ component of domain correlation coefficient matrix w, and the meanings of the other variables are the same as formula (3). Sequential datasets from different domains may contain distinct temporal dependencies and contextual patterns. By assigning larger weights to source domains with higher relevance to the target domain, WMMD can better preserve useful sequential characteristics during distribution alignment while reducing the influence of unrelated temporal patterns. $\mathrm{WMMD}_F^2(S, T)$ is equivalent to $\mathcal{L}_{dis}$ in this paper.

Apart from aligning the target and source distributions, the representation learning on each source dataset is also critical for prediction results. Given M source datasets $s = s_1, s_2, \ldots, s_M$ with the provided labels Y^{s_j} $(j \in [1, M])$ for each dataset s_j. For the task of vulnerability multi-

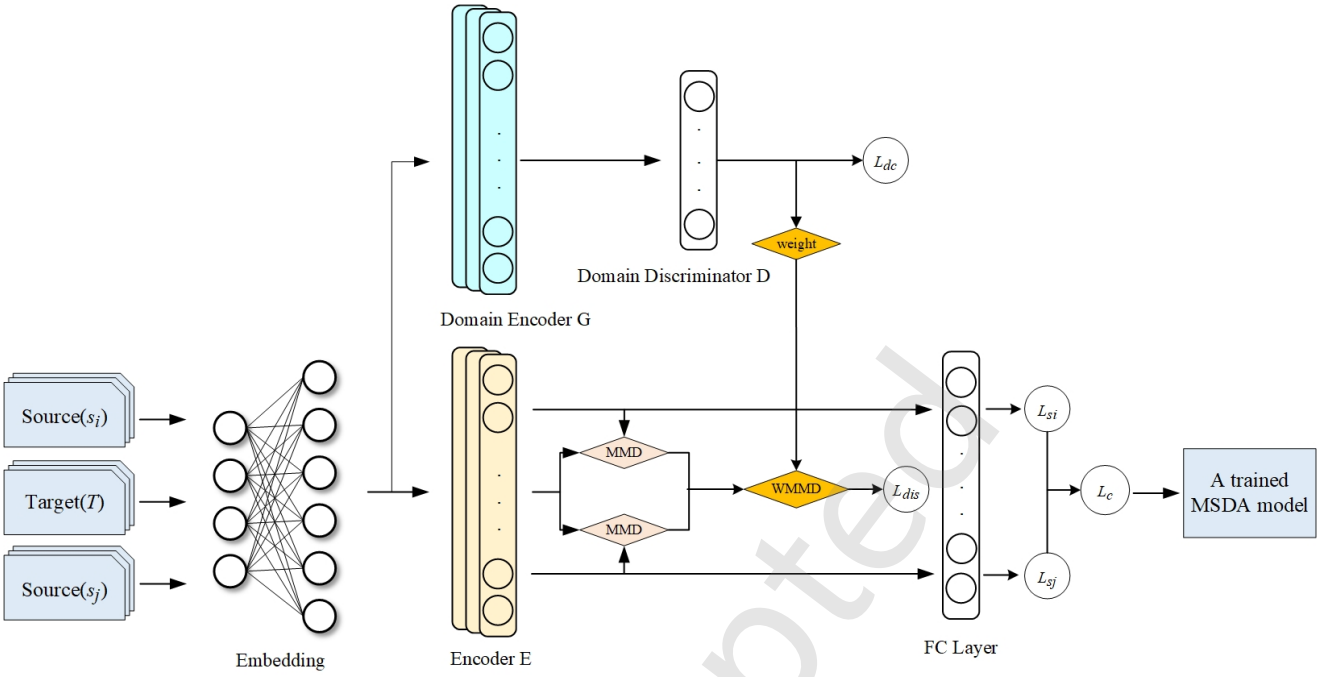

Fig. 3 Overview of MSDA4SD

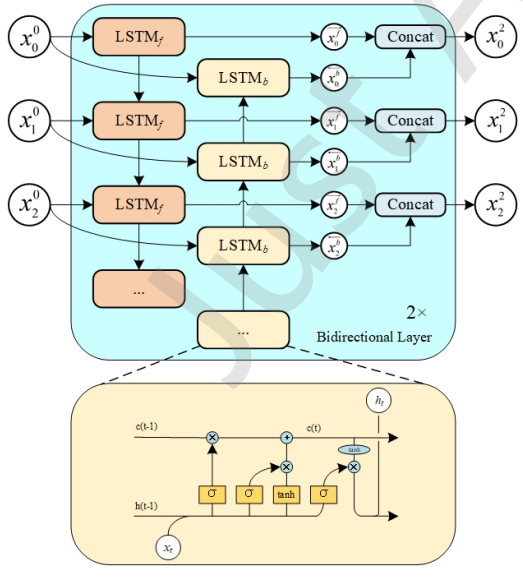

Fig. 4 Overview of MSDA4SD encoding process

classification, we choose the cross-entropy loss, which is calculated using the following formula:

$$\mathcal{L}_{c1} = -\frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{n_{s_j}} \boldsymbol{Y}_i^{\top} \ln FC\left(E\left(X_i^{s_j}\right)\right) \tag{5}$$

For the task of traffic flow prediction, we choose mean square error loss, which is calculated using the following formula:

$$\mathcal{L}_{c2} = -\frac{1}{M}\sum_{j=1}^{M}\frac{\sum_{i=1}^{n_{s_i}}\left(FC\left(E\left(X_i^{S_j}\right)\right) - Y_i\right)^2}{n_{s_i}} \tag{6}$$

To sum up, the overall loss function of MSDA4SD is:

$$\mathcal{L}_{total} = \min_{G,D}\mathcal{L}_{dc} + \gamma\min_{E}\mathcal{L}_{dis} + \min_{E,FC}\mathcal{L}_{c} \tag{7}$$

where γ is the trade-off parameter. $\mathcal{L}_c$ refers to $\mathcal{L}_{c1}$ or $\mathcal{L}_{c2}$.

During the prediction phase, the previously trained model will be used for MSDA on target dataset (T) samples. Specifically, samples are first embedded to obtain feature vectors. Subsequently, the vectors are input into E and the final vector is extracted through E. Then, the final vector is passed through FC layer. For vulnerability multi-classification task, softmax processing is applied to the output of the FC layer to obtain prediction results. For traffic flow prediction task, the output of the FC layer is the prediction result.

The computational complexity of MSDA4SD mainly comes from the Bi-LSTM encoder and the WMMD module. The Bi-LSTM encoder processes sequential inputs with linear complexity with respect to sequence length, while the WMMD module introduces additional alignment computation

across multiple source domains. Although the attention-based weighting mechanism increases training cost, the overall complexity remains manageable for practical sequential data analysis tasks.

4 Experiment Setup

In this chapter, we perform two typical tasks on two types of typical sequential data: classification and prediction. We describe the MSDA datasets we construct, which include vulnerability multi-classification dataset and traffic flow prediction dataset (Section 4.1). Then, we describe our evaluation metrics and implementation details (Section 4.2), and finally introduce the baseline models for comparison (Section 4.3).

4.1 Datasets

Vulnerability Multi-classification Datasets Our datasets come from the software assurance reference dataset (SARD) [40], a growing collection of test applications with documented weaknesses. Our datasets are built from six real-world Java datasets in test suites at SARD. Each dataset contains the common 44 classes of vulnerabilities after processing, each class of vulnerabilities contains multiple samples, and each sample is a function code with vulnerability. The datasets are introduced in Table 3, and the vulnerability classes in the table are sorted by numbers of samples from left to right. We take each dataset in turn as the target domain, with the remaining five datasets as the source domains.

Table 3 The statistics of the datasets built

Domain (Dataset)	Vulnerability class				All 44
	CWE-89	CWE-606	...	CWE-839	
Apache Jena	39 (9.2%)	26 (6.1%)	...	4 (0.9%)	426
Apache Lenya	42 (10.8%)	24 (6.2%)	...	1 (0.3%)	390
Apache Lucene	38 (9.0%)	27 (6.4%)	...	4 (0.9%)	422
Apache POI	40 (9.2%)	27 (6.2%)	...	2 (0.5%)	437
Coffee MUD	41 (9.6%)	27 (6.3%)	...	2 (0.5%)	428
ElasticSearch	39 (9.1%)	28 (6.5%)	...	2 (0.5%)	430

Traffic Flow Prediction Datasets Our datasets come from three real-world public datasets.

TPD1 (Traffic Prediction Dataset 1)[①]: The TPD1 dataset is data monitored by cameras on a certain highway section in the United States, which counts the total number of detected vehicle types with a 15 minute interval and a one month statistical period. It contains approximately 3000 pieces of data in total.

TPD2 (Traffic Prediction Dataset 2)[②]: The TPD2 dataset is data collected from multiple different locations in the United States, including data from November 1, 2015 to June 30, 2017, covering 20 months with a time interval of 1 hour. The amount of data at each intersection is approximately 15000.

TPD3 (Traffic Prediction Dataset 3)[41]: The TPD3 dataset is collected from multiple different locations on the Hangzhou elevated road, covering a period of 3 months from October 4, 2015 to January 3, 2016, with a 5-minute interval. The amount of data at each intersection is approximately 26000.

In both tasks, all source datasets are used as training sets, and for the target dataset, the validation set and test set are divided in a 1:2 ratio. Before training, the raw sequential data are preprocessed through normalization or sequence cleaning operations, followed by tokenization or feature encoding according to different task types. In addition, all input sequences are unified to fixed lengths through truncation and padding operations to ensure consistent model inputs for batch training.

4.2 Evaluation Metrics and Implementation Details

For vulnerability multi-classification tasks, We evaluate and compare the model performance by four typical metrics, including F1 for single classes, and Accuracy (Acc), Kappa, and Matthews correlation coefficient (MCC) for multiple classes. For traffic flow prediction tasks, we use two metrics in the experiments: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

This paper uses the Scott Knott Effect Size Difference (ESD) statistical test method to rank the model. Scott Knott ESD is a mean comparison method that uses hierarchical clustering to divide a dataset into different groups with statistically significant differences [42]. While conducting the Scott Knott ESD test, this paper also calculates the Cohen effect d to measure whether the performance difference between our method and the baseline methods is significant. When $|d| \leqslant 0.2$, it is considered negligible (N, negligible); When $0.2 < |d| \leqslant 0.5$, it is considered small (S, small); When $0.5 < |d| \leqslant 0.8$, it is considered moderate (M, medium); When $|d| \geqslant 0.8$, it is considered large (L, large). If the effect d of one model on another model is positive and greater than 0.2, it indicates that the model has more practical value.

We use bidirectional LSTM (Bi-LSTM) as the encoders for all the methods involved in this paper. Bi-LSTM has a hidden layer dimension of 32 and an encoder layer depth of 2. In the

① TPD1 is acquired at https://www.kaggle.com/datasets/hasibullahaman/traffic-prediction-dataset/data.

② TPD2 is acquired at https://www.kaggle.com/datasets/fedesoriano/traffic-prediction-dataset/data.

training strategy, we use the AdamW optimizer and set the initial learning rate to 0.001. The training epochs are set to 30. For the hyper-parameter γ in formula (7), the optimal value of this hyper-parameter in valid dataset is 0.005. In addition, we further analyzed the influence of several key hyperparameters, including the trade-off coefficient (γ), hidden dimension size, and learning rate. The validation set was used for parameter tuning to avoid overfitting on the testing set. Our model is implemented on the PyTorch platform. Our experiments are all run on NVIDIA RTX A6000 GPU with 48G memory.

4.3 Baseline Methods

We compare MSDA4SD with the following MSDA methods: two classical methods MDAN [26] and M3SDA [35], and a SOTA method ABMSDA [25]. MDAN is a MSDA method based on adversarial training; M3SDA is a MSDA method based on moment matching; ABMSDA is a MSDA method based on attention mechanism. They are all applied in the field of computer vision.

5 Results

In this section, we conduct experiments and analyze the experimental results. Specifically, we investigate the following research questions:

RQ1: How does our method MSDA4SD perform compared with the baseline methods for vulnerability multi-classification in the dataset build?

RQ2: How does our method MSDA4SD perform compared with the baseline methods for traffic flow prediction in the dataset build?

RQ3: Does the WMMD module in MSDA4SD take effect for the multi-classification task and traffic flow prediction task?

5.1 Answering RQ1

To answer RQ1, we conduct comparison experiments using our method and all the baseline methods for all the 44 classes of vulnerabilities in the vulnerability multi-classification datasets built, and make empirical analysis.

Tables 4, 5, and 6 respectively present the evaluation metric results for each target dataset on the vulnerability multi-classification datasets using MSDA4SD and three baseline methods, with the best results highlighted in bold. For Acc, the average Acc value achieved by MSDA4SD on all target datasets is 0.982. Compared with the other three baseline methods, the relative improvement of the average Acc value is between 0.6% and 3.5%. For MCC, the average MCC value achieved by MSDA4SD on all target datasets is 0.981. Compared with the other three baseline methods, the relative improvement of the average MCC value is between 0.5% and 3.5%. However, the average Kappa achieved by MSDA4SD on all target datasets is between 0.981. Compared with the other three baseline methods, the relative improvement of the average Kappa value is between 0.6% and 3.6%. According to the experimental results, the Cohen's d of MSDA4SD relative to the baseline methods across all three evaluation indicators is positive and greater than 0.5. This indicates a statistically significant performance improvement and a relatively large difference in practical applications. Consequently, these results fully demonstrate the high practicality of the MSDA4SD framework in real-world MSDA scenarios.

Figure 5 shows the ranking results of the Scott Knott ESD test for all the methods. In the Scott Knott ESD test ranking, MSDA4SD ranks first in all three evaluation metrics, indicating that the vulnerability multi-classification model constructed using the MSDA4SD method has significant performance improvement compared to other baseline methods.

We also analyze the comparative experiments of MSDA4SD and three baseline methods in different data volume vulnerability scenarios. As shown in Figure 6, the horizontal axis represents the 44 CWE vulnerability categories included in this paper, arranged in descending order of data volume from left to right (as shown by the orange yellow line in Figure 6). Overall, compared with the baseline methods, the proposed method MSDA4SD (as shown by the black line in Figure 6) can achieve the best results in almost all data volume. In summary, MSDA4SD can also achieve the best model prediction results in vulnerability scenarios with different data volume.

The superior performance of MSDA4SD on vulnerability multi-classification tasks mainly benefits from the adaptive domain weighting mechanism introduced by WMMD. Different source domains contribute unequally to the target domain in practical MSDA scenarios. The proposed framework dynamically assigns larger weights to more relevant source domains, which effectively improves feature distribution alignment across domains. In addition, the Bi-LSTM encoder captures contextual dependencies in source code sequences, improving feature representation ability for vulnerability classification tasks.

5.2 Answering RQ2

To answer RQ2, we conduct comparative experiments and empirical analysis on the constructed traffic flow prediction dataset using our method and all the baseline methods.

Table 7 presents the evaluation metric results for each target dataset on the traffic flow prediction datasets using MSDA4SD

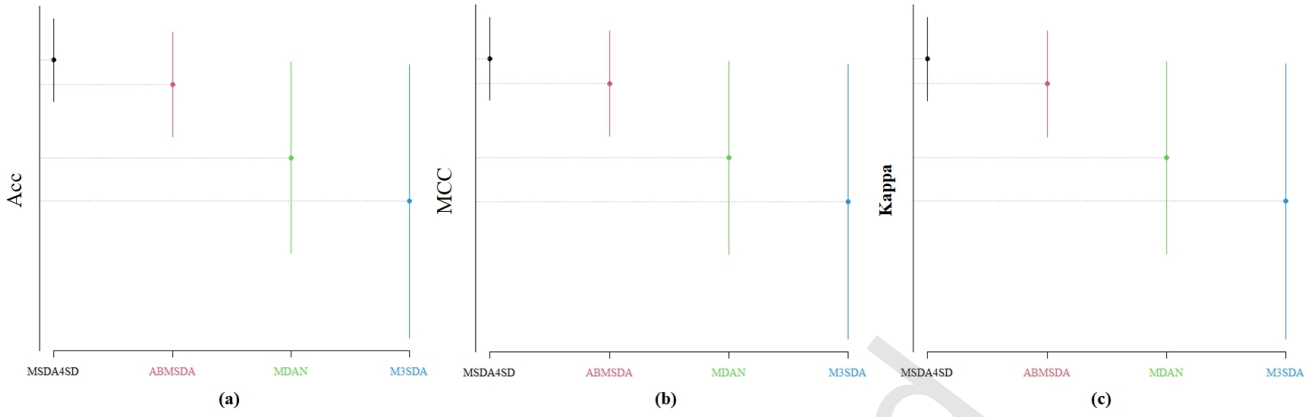

Fig. 5 Scott Knott ESD ranking results for RQ1 (a) Accuracy ; (b) Matthews correlation coefficient; (c) Kappa

Table 4 Acc results for all the methods on the 44 classes of vulnerabilities

Method	Apache Jena	Apache Lenya	Apache Lucene	Apache POI	Coffee MUD	Elastic Search	Avg.	Cohen's d
MDAN	0.954	0.974	0.980	0.928	0.940	0.980	0.959	1.323 (L)
M3SDA	0.960	0.963	0.973	0.889	0.970	0.940	0.949	1.400 (L)
ABMSDA	0.970	0.963	0.983	0.964	0.993	0.983	0.976	0.543 (M)
MSDA4SD	**0.973**	**0.974**	**0.987**	**0.974**	**0.997**	**0.987**	**0.982**	-

Table 5 MCC results for all the methods on the 44 classes of vulnerabilities

Method	Apache Jena	Apache Lenya	Apache Lucene	Apache POI	Coffee MUD	Elastic Search	Avg.	Cohen's d
MDAN	0.952	0.973	0.979	0.926	0.938	0.979	0.958	1.356 (L)
M3SDA	0.959	0.962	0.972	0.886	0.969	0.938	0.948	1.410 (L)
ABMSDA	0.969	0.962	0.983	0.963	0.993	0.983	0.976	0.530 (M)
MSDA4SD	**0.973**	**0.974**	**0.986**	**0.973**	**0.997**	**0.986**	**0.981**	-

Table 6 Kappa results for all the methods on the 44 classes of vulnerabilities

Method	Apache Jena	Apache Lenya	Apache Lucene	Apache POI	Coffee MUD	Elastic Search	Avg.	Cohen's d
MDAN	0.952	0.973	0.979	0.926	0.938	0.979	0.958	1.342 (L)
M3SDA	0.959	0.962	0.972	0.885	0.969	0.937	0.947	1.398 (L)
ABMSDA	0.969	0.962	0.983	0.963	0.993	0.983	0.975	0.512 (M)
MSDA4SD	**0.973**	**0.973**	**0.986**	**0.973**	**0.997**	**0.986**	**0.981**	-

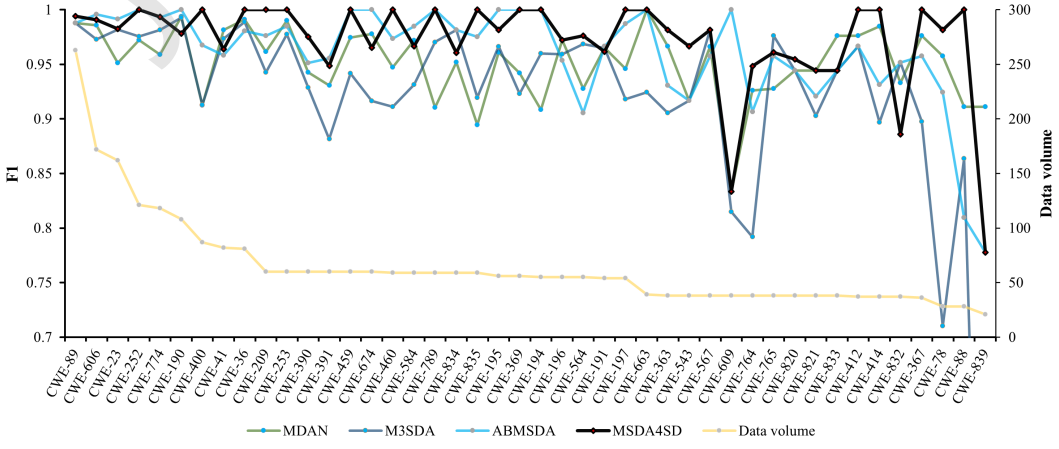

Fig. 6 Comparison of F1 between MSDA4SD and the baseline methods in scenarios with different data volume vulnerabilities

and two baseline methods, with the best results highlighted in bold. Due to the fact that ABMSDA involves label processing for classification tasks in calculations, and there is no category information in traffic flow prediction task, this paper sets the experimental results of ABMSDA to '-'. For TPD1, the MAE obtained by MSDA4SD is 41.407. Compared with MDAN and M3SDA, the MAE decreases by 19.4% and 12.3%. The RMSE obtained by MSDA4SD is 56.807. Compared with MDAN and M3SDA, the RMSE decreases by 16.9% and 11.3%. For TPD2, MSDA4SD achieve a MAE of 4.404. Compared with MDAN and M3SDA, the MAE decreases by 7.3% and 9.6%. The RMSE obtained by MSDA4SD is 7.553. Compared with MDAN and M3SDA, the RMSE decreases by 6.4% and 2.4%. For TPD3, MSDA4SD achieve a MAE of 7.390. Compared with MDAN and M3SDA, the MAE decreases by 8.8% and 21.8%. The RMSE obtained by MSDA4SD is 10.374. Compared with MDAN and M3SDA, the RMSE decreases by 7.7% and 13.7%. This fully demonstrates the practicality of the MSDA4SD framework in multi-source domain adaptation scenarios.

Table 7 MAE and RMSE results for all the methods in traffic flow prediction dataset

Models \ Datasets	TPD1		TPD2		TPD3	
	MAE	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
MDAN	51.388	68.350	4.750	8.069	8.101	11.237
M3SDA	47.225	64.018	4.874	7.741	9.447	12.027
ABMSDA	-	-	-	-	-	-
MSDA4SD	**41.407**	**56.807**	**4.404**	**7.553**	**7.390**	**10.374**

Figure 7 shows the visualization results of all the methods on the traffic flow prediction datasets. In Figure 7, the horizontal axis represents the time interval. The left vertical axis represents the predicted traffic flow values of the model in the scenario described in this section. The cyan line represents the true label, the blue line represents the MDAN prediction result, the green line represents the M3SDA prediction result, and the red line represents the MSDA4SD prediction result. For TPD1, we randomly select the predicted results of one day for visualization. Given the significant variations in traffic flow at different time points in TPD1, the model encounters relatively high prediction challenges, as depicted in Figure 7 (a). From the Figure, it can be seen that the prediction results of the three methods all deviate from the true labels. However, the prediction results of MSDA4SD are closer to the true labels compared to the baseline method; For TPD2, we randomly visualized the prediction results of a single day to showcase the performance of MSDA4SD, as depicted in Figure 7 (b). In most instances, the predictions generated by MSDA4SD exhibited a closer resemblance to the real labels; For TPD3, we randomly select the prediction results for three consecutive days for visualization, as depicted in Figure 7 (c). After 5:00, the prediction results of MDAN and M3SDA show significant deviations, while the prediction results of MSDA4SD are closer to the true labels in most moments. In summary, MSDA4SD can also achieve the best model prediction results in traffic flow prediction scenarios. The prediction results of MSDA4SD are closer to the real labels.

The effectiveness of MSDA4SD on traffic flow prediction tasks can be attributed to its ability to simultaneously model temporal dependencies and domain discrepancies. Traffic flow data exhibits strong temporal correlations and distribution shifts across different locations. The Bi-LSTM encoder effectively captures long-term temporal patterns, while the WMMD mechanism alleviates feature distribution differences between source and target domains. As a result, the prediction results generated by MSDA4SD are more consistent with the real traffic flow trends.

5.3 Answering RQ3

To answer RQ3, we conduct ablation experiments on vulnerability multi-classification datasets and traffic flow prediction datasets. Specifically, the ablation experiment uses removing the WMMD module from MSDA4SD as the baseline method for performance comparison in MSDA scenarios.

5.3.1 Ablation experiment on vulnerability multi-classification dataset

Table 8 presents the Acc results using MSDA4SD and baseline method on each target dataset, with the best results are bolded. Compared with the baseline method, MSDA4SD achieve the best model performance on all target datasets. Compared with the baseline method $\text{MSDA4SD}_{\text{w/o WMMD}}$, the proposed method MSDA4SD has a relative improvement of 1.1% in average Acc. According to the results of the Cohen effect size, the Cohen's d of MSDA4SD relative to the baseline method in Acc is positive and greater than 1. This value indicates a relatively large performance difference in practical applications. Consequently, it demonstrates that the model performance improvement of MSDA4SD over the baseline method is statistically significant.

5.3.2 Ablation experiment on traffic flow prediction dataset

Table 9 presents the evaluation metric results for each target dataset on the traffic flow prediction dataset using MSDA4SD and baseline method, with the best results highlighted in bold. For TPD1, the MAE and RMSE obtained

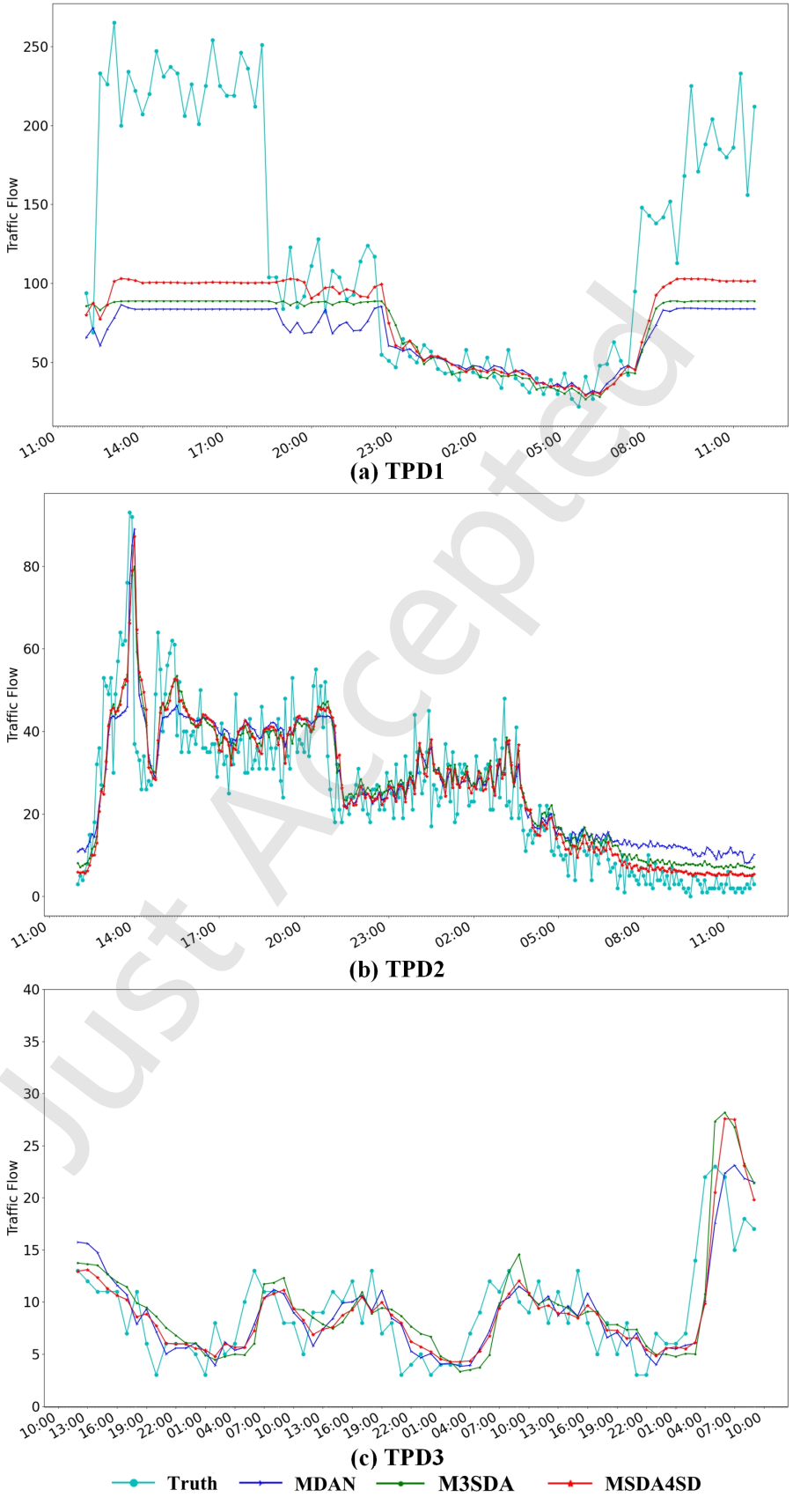

Fig. 7 Visualization results of MSDA4SD and baseline methods on traffic flow prediction dataset. (a) TPD1 (b) TPD2 (c) TPD3

Table 8 Comparison results of Acc in ablation experiments using MSDA4SD

Method	Apache Jena	Apache Lenya	Apache Lucene	Apache POI	Coffee MUD	Elastic Search	Avg.	Cohen's d
MSDA4SD	**0.973**	**0.974**	**0.987**	**0.974**	**0.997**	**0.987**	**0.982**	-
MSDA4SD$_{\text{w/o WMMD}}$	0.967	0.971	0.983	0.967	0.956	0.983	0.971	1.068 (L)

Table 9 Comparison results of MAE and RMSE in ablation experiments using MSDA4SD

Models \ Datasets	TPD1		TPD2		TPD3	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
MSDA4SD	**41.407**	**56.807**	**4.404**	**7.553**	**7.390**	**10.374**
MSDA4SD$_{\text{w/o WMMD}}$	49.319	66.692	4.463	7.676	9.954	12.086

by MSDA4SD are 41.407 and 56.807, respectively. Compared with MSDA4SD$_{\text{w/o WMMD}}$, they decreased by 16.0% and 14.8%, respectively. For TPD2, the MAE and RMSE obtained by MSDA4SD are 4.404 and 7.553, respectively. Compared with MSDA4SD$_{\text{w/o WMMD}}$, they decreased by 1.3% and 1.6%, respectively. For TPD3, the MAE and RMSE obtained by MSDA4SD are 7.390 and 10.374, respectively. Compared with MSDA4SD$_{\text{w/o WMMD}}$, they decreased by 25.8% and 14.2%, respectively. This fully demonstrates the practicality of the MSDA4SD framework in MSDA scenarios.

The ablation results further demonstrate the importance of the WMMD module in MSDA4SD. Without WMMD, the framework cannot effectively distinguish the relevance of different source domains, resulting in weaker domain alignment performance. By introducing weighted domain adaptation, MSDA4SD achieves better feature distribution alignment and improves both classification and prediction performance across heterogeneous domains.

6 Conclusion

This paper proposes Multi-Source Domain Adaptation for Sequential Data (MSDA4SD) for the processing of sequential data. Specifically, MSDA4SD uses a domain discriminator to obtain the attention scores of the source and target datasets, and combines the attention scores to weight the maximum mean discrepancy (MMD), reducing the difference in feature distribution between the source and target datasets. We compare and analyze the performance of MSDA4SD with multiple baseline methods on vulnerability multi-classification datasets and traffic flow prediction datasets. The proposed framework shows good generalization ability across different sequential data analysis scenarios, including vulnerability classification and traffic flow prediction. Nevertheless, several limitations still exist. The current framework may face additional challenges when handling very long sequential dependencies or highly non-uniform domain distributions. In such scenarios, the effectiveness of domain-correlation modeling and distribution alignment may be affected. Future work will investigate more efficient sequential modeling strategies for long-range dependencies and explore adaptive domain weighting mechanisms for more complex heterogeneous multi-source environments.

Acknowledgment

This work was supported by the Open Foundation of Yunnan Key Laboratory of Software Engineering (No.2023SE202), the CCF-NSFOCUS 'Kunpeng' Research Fund (CCF-NSFOCUS 2024012), Beijing-Tianjin-Hebei Natural Science Foundation Cooperation Project (No. 25JJJJC0030) and the National Natural Science Foundation of China (No.U24B20150).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

Reference

[1] F. Lu, W. Li, Y. Sun, C. Song, Y. Ren, and A. Y. Zomaya, Cgs-mask: Making time series predictions intuitive for all, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 13, 2024, pp. 14 149–14 157.

[2] Y. Yang and J. Jiang, Adaptive bi-weighting toward automatic initialization and model selection for hmm-based hybrid meta-clustering ensembles, *IEEE transactions on cybernetics*, vol. 49, no. 5, pp. 1657–1668, 2018.

[3] A. Ali, H. Y. Naeem, A. Sharafian, L. Qiu, Z. Wu, and X. Bai, Dynamic multi-graph spatio-temporal learning for citywide traffic flow prediction in transportation systems, *Chaos, Solitons & Fractals*, vol. 199, p. 116898, 2025.

[4] A. Ali, A. Sharafian, H. Y. Naeem, M. Zakarya, Z. Wu, and X. Bai, Advanced computational models for urban traffic flow prediction: A comprehensive review and future directions, *Computer Science Review*, vol. 60, p. 100886, 2026.

[5] A. Ali, I. Ullah, S. Ahmad, Z. Wu, J. Li, and X. Bai, An attention-driven spatio-temporal deep hybrid neural networks for traffic flow prediction in transportation systems, *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[6] J. Liu, J. Zeng, X. Wang, and Z. Liang, Learning graph-based code representations for source-level functional similarity detection, in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 2023, pp. 345–357.

[7] X. Wang, D. Guo, and P. Cheng, Support structure representation learning for sequential data clustering, *Pattern Recognition*, vol. 122, p. 108326, 2022.

[8] X. Liu, D. Chen, W. Wei, X. Zhu, and W. Yu, Interpretable system identification and long-term prediction on time-series data, *arXiv preprint arXiv:2303.01193*, 2023.

[9] Y. Li, Y. Tao, M. Zhu, Z. Chen, Z. Wen, and B. Zhu, Tb-former: Multi-scale transformer with time-behavior attention for multi-modal customer churn prediction, *IEEE Transactions on Consumer Electronics*, 2025.

[10] H. Pan, X. Chu, R. Miao, M. Wang, Y. Wang, and Z. Li, Text generation of speech imagery based on an enhanced cta-bilstm model utilizing eeg signals, *IEEE Transactions on Consumer Electronics*, 2025.

[11] C. Zhan, D. Yin, Y. Shen, and T. Hao, Gminn: A generative moving interactive neural network for enhanced short-term load forecasting in modern electricity markets, *IEEE Transactions on Consumer Electronics*, vol. 70, no. 3, pp. 5461–5470, 2024.

[12] S. Zhang, Z. Wang, Z. Zhou, Y. Wang, H. Zhang, G. Zhang, H. Ding, S. Mumtaz, and M. Guizani, Blockchain and federated deep reinforcement learning based secure cloud-edge-end collaboration in power iot, *IEEE Wireless Communications*, vol. 29, no. 2, pp. 84–91, 2022.

[13] Y. Yang, Y. Hu, X. Zhang, and S. Wang, Two-stage selective ensemble of cnn via deep tree training for medical image classification, *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9194–9207, 2021.

[14] H. J. Kim, S. E. Hong, and K. J. Cha, seq2vec: Analyzing sequential data using multi-rank embedding vectors, *Electronic Commerce Research and Applications*, vol. 43, p. 101003, 2020.

[15] Y. Xing, X. Qian, Y. Guan, S. Zhang, M. Zhao, and W. Lin, Cross-project defect prediction method using adversarial learning, *Journal of Software*, vol. 33, no. 6, pp. 2097–2112, 2022.

[16] J. M. L. Alcaraz and N. Strodthoff, Diffusion-based time series imputation and forecasting with structured state space models, *arXiv preprint arXiv:2208.09399*, 2022.

[17] S. Sun, H. Shi, and Y. Wu, A survey of multi-source domain adaptation, *Information Fusion*, vol. 24, pp. 84–92, 2015.

[18] P. Wang, Y. Yang, Y. Xia, K. Wang, X. Zhang, and S. Wang, Information maximizing adaptation network with label distribution priors for unsupervised domain adaptation, *IEEE Transactions on Multimedia*, vol. 25, pp. 6026–6039, 2022.

[19] F. Qiao, J. Wu, J. Li, A. K. Bashir, S. Mumtaz, and U. Tariq, Trustworthy edge storage orchestration in intelligent transportation systems using reinforcement learning, *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4443–4456, 2020.

[20] T. Alsarhan, S. S. Ali, I. I. Ganapathi, A. Ali, and N. Werghi, Ph-gcn: boosting human action recognition through multi-level granularity with pair-wise hyper gcn, *IEEE Access*, vol. 12, pp. 162 608–162 621, 2024.

[21] A. Ali, I. Ullah, S. K. Singh, W. Jiang, F. Alturise, and X. Bai, Attention-driven graph convolutional networks for deadline-constrained virtual machine task allocation in edge computing, *IEEE Transactions on Consumer Electronics*, 2025.

[22] L. Cao, B. Wang, G. Jiang, Y. Yu, and J. Dong, Spatiotemporal-aware trend-seasonality decomposition network for traffic flow forecasting, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 11, 2025, pp. 11 463–11 471.

[23] X. He, Asiya, D. Han, S. Zhou, X. Fu, and H. Li, An improved software source code vulnerability detection method: Combination of multi-feature screening and integrated sampling model, *Sensors*, vol. 25, no. 6, p. 1816, 2025.

[24] S. Sun, J. Wang, T. Yan, and F. Qi, A novel fine-grained source code vulnerability detection model via joint token and statement representation learning, in *International Symposium on Grids and Clouds (ISGC2025)*, vol. 16, 2025, p. 21.

[25] Y. Zuo, H. Yao, and C. Xu, Attention-based multi-source domain adaptation, *IEEE Transactions on Image Processing*, vol. 30, pp. 3793–3803, 2021.

[26] H. Zhao, S. Zhang, G. Wu, J. a. P. Costeira, J. M. F. Moura, and G. J. Gordon, Multiple source domain adaptation with adversarial learning, in *International Conference on Learning Representations(ICLR)*, 2018.

[27] Y.-H. Liu and C.-X. Ren, A two-way alignment approach for unsupervised multi-source domain adaptation, *Pattern Recognition*, vol. 124, p. 108430, 2022.

[28] V.-A. Nguyen, T. Nguyen, T. Le, Q. H. Tran, and D. Phung, Stem: An approach to multi-source domain adaptation with guarantees, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9352–9363.

[29] C.-X. Ren, Y.-H. Liu, X.-W. Zhang, and K.-K. Huang, Multi-source unsupervised domain adaptation via pseudo target domain, *IEEE Transactions on Image Processing*, vol. 31, pp. 2122–2135, 2022.

[30] S. Peng, R. Zeng, L. Cao, A. Yang, J. Niu, C. Zong, and G. Zhou, Multi-source domain adaptation method for textual emotion classification using deep and broad learning, *Knowledge-Based Systems*, vol. 260, p. 110173, 2023.

[31] Z. Taghiyarrenani, S. Nowaczyk, S. Pashami, and M.-R. Bouguelia, Multi-domain adaptation for regression under conditional distribution shift, *Expert systems with applications*, vol. 224, p. 119907, 2023.

[32] Y. Lu and W. Huang, Towards discriminability with distribution discrepancy constrains for multisource domain adaptation, *Mathematics*, vol. 12, no. 16, p. 2564, 2024.

[33] Y. Dai, J. Liu, X. Ren, and Z. Xu, Adversarial training based

multi-source unsupervised domain adaptation for sentiment analysis, in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 7618–7625.

[34] S. Zhao, G. Wang, S. Zhang, Y. Gu, Y. Li, Z. Song, P. Xu, R. Hu, H. Chai, and K. Keutzer, Multi-source distilling domain adaptation, in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 12 975–12 983.

[35] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, Moment matching for multi-source domain adaptation, in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1406–1415.

[36] S. Wang, B. Wang, Z. Zhang, A. A. Heidari, and H. Chen, Class-aware sample reweighting optimal transport for multi-source domain adaptation, *Neurocomputing*, vol. 523, pp. 213–223, 2023.

[37] Y. Xu, J. Yang, H. Cao, K. Wu, M. Wu, Z. Li, and Z. Chen, Multi-source video domain adaptation with temporal attentive moment alignment network, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 3860–3871, 2023.

[38] A. Belal, A. Meethal, F. P. Romero, M. Pedersoli, and E. Granger, Attention-based class-conditioned alignment for multi-source domain adaptation of object detectors, in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2025, pp. 8566–8575.

[39] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, Domain generalization: A survey, *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 4, pp. 4396–4415, 2022.

[40] Nist software assurance reference dataset, 2022, https://samate.nist.gov/SARD/test-suites.

[41] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, Lstm-based traffic flow prediction with missing data, *Neurocomputing*, vol. 318, pp. 297–305, 2018.

[42] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto, The impact of automated parameter optimization on defect prediction models, *IEEE Transactions on Software Engineering*, vol. 45, no. 7, pp. 683–711, 2018.

Author biography

Zujun Liu received his B.S. from Wuhan University and M.S. from the University of Iowa. He is currently the Algorithm Director at Smart Steps Digital (China Unicom). His research interests include graph computing, spatio-temporal data mining, and knowledge graphs. He has led multiple national R&D initiatives, published in *Genome Research*, and holds several patents in location-based services and traffic monitoring.

Mengfan Zhao received the B.S. degree from Taiyuan University of Technology. He is currently a master student in the School of Intelligent Engineering and Automation, Beijing University of Posts and Telecommunications, Beijing, China. His research focuses on artificial intelligence and deep learning.

Ying Xing received the Ph.D. degree in Computer Science and Technology from Beijing University of Posts and Telecommunications, Beijing, China in 2014. She is currently a professor with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications. Her research interest focuses on the application of artificial intelligence.

Zan Zhou received the Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2022. From 2021 to 2022, he was a visiting student with Nanyang Technological University (NTU), Singapore. He is currently an Assistant Professor with the School of Computer Science (National Pilot Software Engineering School), BUPT. His research interests include data privacy, active defense, and federated learning. He received the IEEE UIC 2024 Outstanding Paper Award and the IEEE ICBCTIS 2025 Best Paper Award. He is a Member of the IEEE.

Fei Gao is a Professor and Ph.D. Supervisor at Tibet University, and a member of CCF. His research focuses on informatization and research management. He has led over 30 national and provincial projects and published 60+ academic papers. Prof. Gao holds multiple patents and senior certifications in blockchain and big data. He has received numerous honors, including the National Teaching Achievement Award and the Science and Technology Progress Award of the Tibet Autonomous Region.

Yiliang Lai received the Ph.D. degree in computer science. He is a Master's Supervisor at Xiamen University of Technology and a researcher at Assumption University of Thailand. His research interests include artificial intelligence, computer vision, and software supply chain security. Dr. Lai has led several major industrialization projects and has published over 30 academic works and patents. He received the 2024 IEEE Best Paper Award and has contributed to multiple national standards.