Neural Computing and Applications (2022) 34:8919–8931
https://doi.org/10.1007/s00521-022-06914-1

ORIGINAL ARTICLE

Multi-task deep learning model based on hierarchical relations of address elements for semantic address matching

Fangfang Li[1] · Yiheng Lu[1] · Xingliang Mao[2] · Junwen Duan[1] · Xiyao Liu[1]

Received: 8 May 2021 / Accepted: 4 January 2022 / Published online: 18 February 2022

Abstract

Address matching, which aims to match unstructured addresses with standard addresses in an address database, is a key part of geocoding. The core problem of address matching corresponds to text matching in natural language processing. Existing rule-based methods require human-designed templates and thus, have limited applicability. Machine learning and deep learning-based methods ignore the hierarchical relations between address elements, which easily misclassify semantically similar but geographically different locations. We note that the hierarchy of address elements can fill the semantic gap in address matching. Inspired by how humans discriminate addresses, we propose a multi-task learning approach. The approach jointly recognises the address elements and matches the addresses to incorporate the hierarchical relations between the address elements into the neural network. Simultaneously, we introduce a priori information on the hierarchical relationship of address elements through the conditional random field model. Experimental results on the benchmark datasets Shenzhen Address Database and Jiangsu-Hunan Address Dataset demonstrate the effectiveness of our approach. We achieved state-of-the-art $F1$ scores (i.e. the harmonic mean of precision and recall) of 99.0 and 94.2 on the two datasets, respectively.

Keywords Address matching · Multi-task learning · Recognizes the address elements · Hierarchical relations between address elements

1 Introduction

Address matching [1], in which unstructured addresses are matched with structured addresses to locate them on a map, is an important application area in geographic information science. Much of the urban information is related to geographic location [2]; however, most of this information does not have spatial coordinates and thus cannot be integrated for analysis. Address matching can integrate addresses and facilitate the analysis of positioning systems, which constitute the core aspects of digital city construction. Cab operations, courier logistics, and other services rely on geographic query and address matching technologies. The core problem of address matching corresponds to text matching [3] in natural language processing.

Traditional rule-based address matching methods can be divided into two categories: character-based methods, which discriminate similarity character by character, and address element-based methods, which segment address elements using manually designed rules and then match each address element. However, designing rules is not only labour-intensive, but also restricted to specific and more standardised addresses. In recent years, deep learning has been increasingly applied to geographic information science: first, it can automatically model address features and

✉ Xingliang Mao
maoxingliang@hutb.edu.cn

Fangfang Li
lifangfang@csu.edu.cn

Yiheng Lu
luyiheng@csu.edu.cn

Junwen Duan
jwduan@csu.edu.cn

Xiyao Liu
lxyzoewx@csu.edu.cn

[1] School of Computer Science and Engineering, Central South University, Changsha, China

[2] Institute of Big Data and Internet Innovation, Hunan University of Technology and Business, Changsha, China

avoid manual design rules; second, deep learning can extract semantic information, which is suitable for a variety of address structures, especially irregular addresses. However, as shown in Table 1, there are misclassifications for addresses that are semantically similar and actually represent different locations.

In response to the above issues, *we note that there is a semantic gap between the semantic similarity of addresses and ascertain whether they match, and that the information of the hierarchical relationship of address elements can solve the semantic gap problem, which is ignored by the current deep learning methods.* When we determine whether an address pair matches, we compare only the following address elements if the front address elements are the same.

To solve the above problem, we incorporate information about the hierarchical relationships of address elements from both the data and model aspects. On the data side, we first train an address element recognition model to tag the elements of a large number of unlabelled addresses. On the model side, we use an address element recognition module to assist the address matching module from the perspective of joint multi-task learning [4]. In addition, we introduce a priori information about the hierarchical relationships of address elements to effectively improve the address matching performance. The main contributions of the study are as follows.

(1) We incorporate information about the hierarchical relationship of address elements into a deep learning model to facilitate the development of address matching.
(2) We pre-trained the model to tag address elements, which solved the problem that a large amount of untagged address data could not be used.
(3) We propose a multi-task learning model for address element recognition and address matching, thus incorporating information about the hierarchical relationships of address elements in the model, using the transition probability matrix of the conditional random field (CRF) classifier [5] to incorporate a priori information about the hierarchical relationship of address elements into the model.
(4) After the experimental comparison, our model outperforms existing methods and achieves the best results.

The article is structured as follows. In Sect. 2, we introduce the development and status of address matching. We introduce our proposed multi-task learning model for address element recognition and address matching in Sect. 3, and we conduct comparison and ablation experiments on the Shenzhen Address Database and the Jiangsu-Hunan Address Dataset in Sect. 4. Finally, we conclude the paper in Sect. 5.

2 Related works

Address matching is generally divided into rule-based matching and semantic similarity matching based on machine learning and deep learning.

Rule-based address matching methods are divided into two categories: a character-based method that discriminates similarity character by character, and an address element-based method that segments address elements using manually designed rules and then matches each address element. Tian et al. [6] and Koumarelas et al. [7] designed some great rules for address matching, but the effect of address alias processing needs to be improved. Santos et al. [8] integrated multiple character similarities to consider address similarity and achieved some results. However, designing rules is not only labour-intensive but can only handle more standardised addresses.

In recent years, an increasing number of machine learning and deep learning methods have been applied to many natural language processing applications, Zhou et al. [9] focus on modeling and analyzing the patient-physician-generated data based on an interrated CNN-RNN famework, and to geographic information science to extract text semantics [10–14]. Acheson et al. [15] used rules combined

Table 1 Samples of the Shenzhen Address database

No	Address pairs	Hierarchical relations of address elements	Match or not
1	2-1, Lane 1, Longtengge, Baishixia Community, Fuyong Street, Baoan District, Shenzhen (深圳宝安区福永街道白石厦社区龙腾阁1巷2-1)	City-District-Street- Community, Village-Road, Lane - House Number (市-区-街道-社区、村-道路、巷-门牌号)	True
	No. 2, Lane 1, Longtengge, Defeng Road, Fuyong Street, Baoan District, Shenzhen (深圳宝安区福永街德丰路龙腾阁一□2号)	City-District-Street- Road, Lane-House Number (市-区-街道-道路、巷-门牌号)	
2	101, No. 24, Xiangnan Village, Nanshan Street, Nanshan District, Shenzhen(深圳市南山区南山街道向南村24号101)	City-District-Street- Community, Village -House Number (市-区-街道-社区、村-门牌号)	False
	No. 101, Xiangnan Community, Nanshan Street, Nanshan District, Shenzhen (深圳市南山区南山街道向南社区101号)	City-District-Street- Community, Village -House Number (市-区-街道-社区、□-门牌号)	

with random forest methods in machine learning for cross-gazetteer matching. Comber et al. [16] used CRF and Word2Vec [17] for address matching without manually designing complex rules, but only shallow semantic features were extracted. Santos et al. [18] used deep neural networks for address matching, and Lin et al. [19] also used the classical enhanced sequential inference model (ESIM) [20], a deep learning model, for address record pair modelling, which extracts the deep semantic features of addresses and achieves better results; however, it ignores the information about the hierarchical relationship of address elements. There are misclassifications for some semantically similar addresses that represent different locations, hierarchical information seems to be important, Shi et al. [21] proposed a hierarchical ASM search strategy to make pathological organ segmentation framework more efficient and robust.

Our proposed address matching method based on joint multi-task learning with the hierarchical relationship of address elements not only automatically extracts the deep semantic features of address text, but also incorporates the knowledge of the hierarchical relationship of address elements, and it enables the model to learn the information of the hierarchical relationship of address elements by multi-task learning.

3 Methods

We propose a multi-task address matching deep learning model based on address element recognition to discriminate address matches. The overall structure of our model is shown in Fig. 1. To learn the deep semantics of addresses, we design a deep learning address matching model based on address element recognition and incorporate the knowledge of the hierarchical relationship of address elements by imitating the process of human discriminating whether an address matches or not, that is, dividing addresses hierarchically and comparing and analysing the address elements at each level. As shown in Fig. 1, the model mainly contains three modules: the address element tagging network based on the word segmentation features, the knowledge module of the hierarchical relationship of address elements, and the multi-task network for joint learning of address element recognition and address matching. The knowledge module of the hierarchical relationships of address elements encodes a priori hierarchical relationships of address elements into the address element identification network during model training. The address element recognition and address matching multi-task learning network are a joint learning of the address element recognition task and address matching task, which acts on the training of the model simultaneously.

3.1 Word embedding layer with segmentation features

Word embedding is a distributed representation of words, and distributed representations are more suitable as inputs to neural networks. We used the CBOW model of Word2Vec to train the address corpus. As shown in Fig. 2, CBOW is a shallow neural network consisting of a three-layer network of input, projection, and output layers, which maps address text into a low-dimensional dense feature space with specific meanings. The CBOW model trains word embeddings by maximising the average log probability.

$$\frac{1}{T}\sum_{i=1}^{T}\log(p(w_i|Context_{w_i})) \tag{1}$$

$$p(w_b|w_a) = \frac{\exp(e'(w_b)^T e(w_a))}{\sum_{k=1}^{|V|}\exp(e'(w_k)^T e(w_a))} \tag{2}$$

where T is the number of words in the text, $Context_{w_i}$ is the context of w_i, $p(w_b|w_a)$ is the probability of predicting the occurrence of the bth word by the ath word in the text, $|V|$ is the total number of classes of words in the text, $e(w_i)$ denotes the word embedding representation of word w_i, and $e'(w_i)$ denotes another word embedding representation of word w_i. Therefore, words with similar meanings are eventually more similar in semantic feature space.

In addition, the Jieba word splitting tool is used to split the original address, the splitting information is encoded according to the following formula, and then, the CBOW model is used to map the encoded text into a fixed dimensional splitting vector. Finally, the word vector is spliced with the word vector of the original text and used as the input for the model. For example, 'Bai Shi Xia Community, Fuyong Street, Baoan District, Shenzhen (□□□□□□□□□□□□□□□)' is divided into 'Shenzhen/Baoan District/Fuyong Street/Bai Shi Xia Community (□□□/□□□/□□□□/□□□□□)' and encoded as '0 1 2/0 1 2/0 1 1 2/0 1 1 1 2'.

$$f(x) = \begin{cases} 0 & x \text{ is at the beginning of } w \\ 1 & x \text{ is in the middle of } w \\ 2 & x \text{ is in the end of } w \end{cases} \tag{3}$$

where x is a character in the current word w.

3.2 Address element tagging network

To automatically tag a large number of unlabelled data, we manually tag the address elements of a small-scale dataset and design an address element tagging network based on word segmentation features. The address element tagging network is trained through a small-scale manually tagged

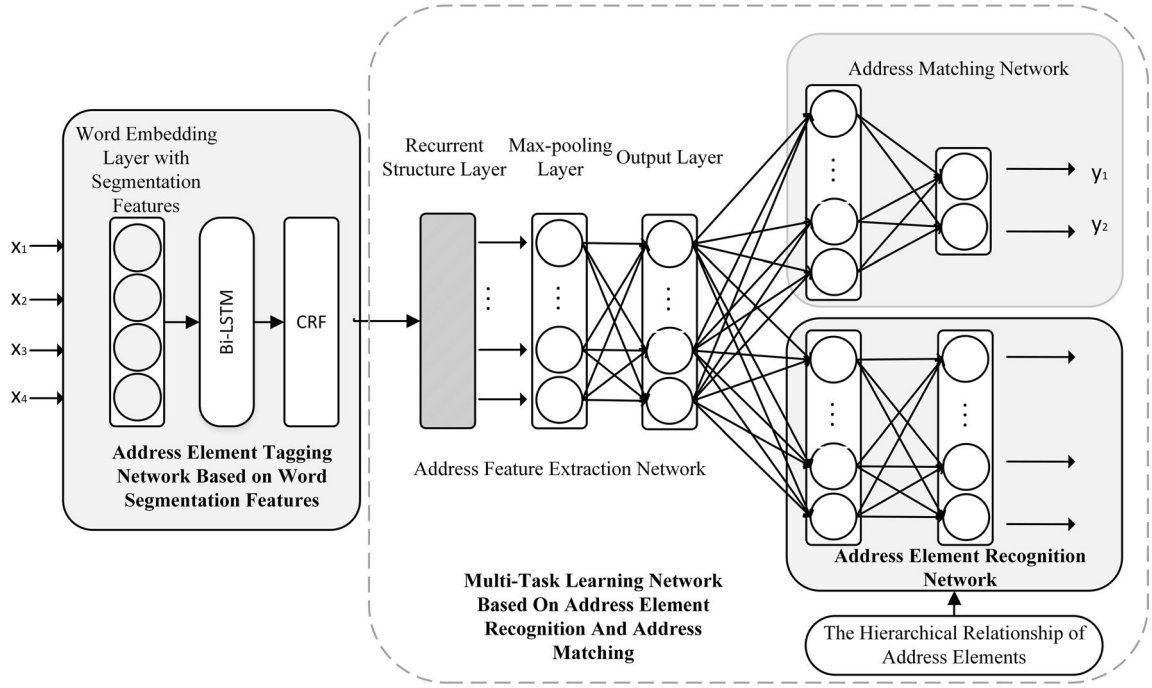

Fig. 1 Overall structure of the model

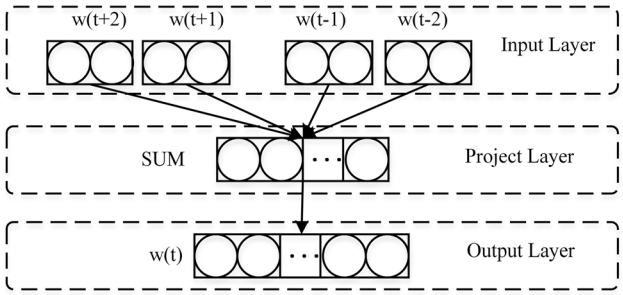

Fig. 2 The structure of the CBOW model

dataset, and the trained address element tagging network is used to tag a large-scale dataset. It is worth noting that the label obtained in this way is pseudo, but the accuracy of the address element tagging network itself is high. Therefore, obtaining a pseudo-label by this method not only saves a lot of manual tagging workload but also has less damage to the credibility of the label, a lot of works has proved the effectiveness of pseudo tags, Zhou et al. [22] designed a auto-labeling scheme based on Deep Q-Network(DQN) to improve the learning efficiency in IoT environments.

3.2.1 Address element label

To train an address element tagging network and tag large-scale datasets through the network, we first need to design a tagging system. To solve this problem, an appropriate address hierarchy element tagging system according to the suggestions of professionals is used. The address hierarchy element tagging system is shown in Table 2.

3.2.2 Structure of address element tagging network

As shown in Fig. 3, we use a bidirectional long short-term memory (Bi-LSTM) model [23], which is a type of recurrent neural network [24], combined with CRF [25] to tag the address hierarchy elements. Compared with traditional recurrent neural networks, LSTM introduces a gate mechanism that includes an input gate, output gate, and forgetting gate. The forgetting gate can filter out useless information and capture long-distance dependencies, which alleviates the problem of key information being forgotten due to gradient dispersion in traditional recurrent neural networks and still preserves important information above when processing below. Thus, LSTM can remember longer sequential information and is suitable for modelling address text. To obtain contextual information simultaneously, BiLSTM concatenates the hidden states of the forward LSTM and backward LSTM to more comprehensively represent the semantic information of sentences.

The CRF model combines the advantages of the hidden Markov model (HMM) [26] and max entropy Markov model (MEMM) [27] and also avoids the label bias problem in MEMM. CRF is a key technique for named entity recognition [28].

3.2.3 Loss function

We use the CRF to calculate the loss of the address hierarchy element tagging network. In contrast to using the cross-entropy function directly as a loss, the CRF layer can add some constraints to the final predicted labels to ensure that the predicted labels are reasonable, with the following loss function:

$$loss = -\log(p(y_{gold}|X)) = -s(X, y_{gold}) + \log\left(\sum_{y \in Y_x} e^{s(X,y)}\right) \quad (4)$$

where X denotes the input word sequence, y_{gold} denotes the actual label sequence, $s(X, y_{gold})$ denotes the score of the label sequence, and Y_x denotes all possible prediction sequences corresponding to the input sequence x.

3.3 Multi-task learning

To enhance the robustness and accuracy of the address matching model, we imitated the human judgement process of address matching, compared different hierarchical elements among different addresses, and designed a multi-task learning network of address element recognition and address matching, so that the model can learn the dependency relationship among different address elements in the process of judging address matching and assist the model in judging the address matching more reasonably and accurately. We used parameter sharing to train the multi-task model.

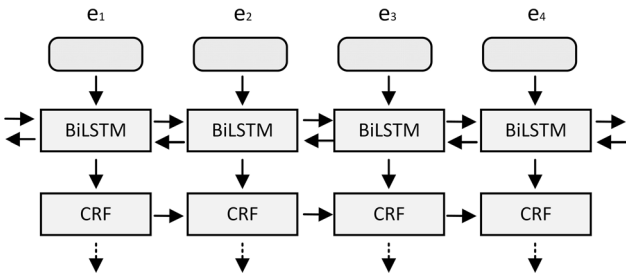

Fig. 3 The structure of the address element tagging network

3.3.1 Address feature extraction network

Considering the hierarchical structure of addresses and the interaction of neighbouring elements, recurrent convolutional neural networks (RCNN) [29] are used as the basic feature extractor. As shown in Fig. 4, the feature extractor mainly consists of a word representation layer, which extracts information about the address element hierarchy, and a text representation layer, which extracts more global information about the entire address. For example, 'Yantian District (□□□)' in 'Building A, No. 1051 Wutong Road, Tiandong Community, Haishan Street, Yantian District, Shenzhen City (□□□□□□□□□□□□□□□□□□□□ 1051□A□)' is represented by 'Shenzhen City (□□□)' above, 'Haishan Street (□□□□)' below, and itself.

3.3.2 Address element recognition network based on address element hierarchy information

After the feature extraction layer, the features associated with the address elements are further extracted using the fully connected layer and combined with the CRF layer for address element recognition.

Table 2 Address element classification table

Element type	Element tag	Explain
Administrative area	XZQHS	Municipalities/provinces/autonomous regions and special administrative regions
	XZQHCS	Provincial capital/prefecture-level city/autonomous prefecture/region/alliance
	XZQHQX	County /(county-level) city/flag/autonomous town/(municipality) district
	XZQHJD	Street office/community/township/town
	XZQHSQ	Village
Road	JD1	Road/avenue /lane/alley
Compound	JD2	District/garden/community
Building	JD3	Apartment/Building
House number	MP1	House Number (House number/section 1 of Building)
Building brand	MP2	Building Number/Dormitory (House number/section 2 of Building)
Unit no	DYS1	Unit (Unit Room/Floor 1 section)
Room no	DYS2	Room (Unit Room/Floor 2 section)
Local point position	POI	Supermarket/market/hotel/building/city/garden building/square/centre/building

To provide the model with a priori knowledge of the hierarchical relationships of address elements, enhance the robustness of the model, and accelerate the convergence of the model, we incorporate the coding of the hierarchical relationships of address elements into the training process of the address element recognition network. First, the transition probabilities $P_{i,j}$ between the various types of address elements in the training corpus were counted.

$$p_{i,j} = \frac{n_{i,j}}{\sum_{k=1}^{t} n_{i,k}} \tag{5}$$

where t denotes the total number of types of address elements, $n_{i,j}$ denotes the number of samples where the ith class of address elements is followed by the jth class of address elements, and the transition probability matrix is used as the transition matrix of the CRF loss function.

3.3.3 Address matching network

After the shared feature extraction layer, based on the address elements information learned, the full connection layer and the ReLU activation function [30] are used to further globally extract the deep features that are most relevant to the address match, to discriminate whether the address pairs match.

3.3.4 Address matching task joint address element recognition task

Joint multi-task learning implies learning a shared representation from other tasks. The use of shared representations in learning different tasks allows what is learned in one task to be better learned in other tasks.

As shown in Fig. 5, we introduce the address element recognition task while performing the main address matching task, so that the address matching task can learn the relationships between different address elements, thus making the address matching model more robust. We perform joint learning of the two tasks through parameter-sharing, a hard-share approach [31], first proposed in 2008. By balancing the noise in both tasks through joint learning of address matching and address element recognition, the model focuses on addressing hierarchical features and is able to capture address representations that incorporate address element hierarchy information, thus reducing the risk of overfitting the model on address matching tasks.

3.3.5 Loss function

The training goal of the network was to minimise the total loss of the model $L(\theta)$

$$L(\theta) = \lambda_1 loss_{cls}(\theta) + \lambda_2 loss_{ner}(\theta) \tag{6}$$

$$\lambda_1 + \lambda_2 = 1 \tag{7}$$

where θ is the model parameter, $loss_{cls}(\theta)$ is the cross-entropy loss of the address matching network, and $loss_{ner}(\theta)$ is the CRF loss of the address hierarchy element recognition network. λ_1 and λ_2 are the weight coefficients of the two aforementioned losses, respectively.

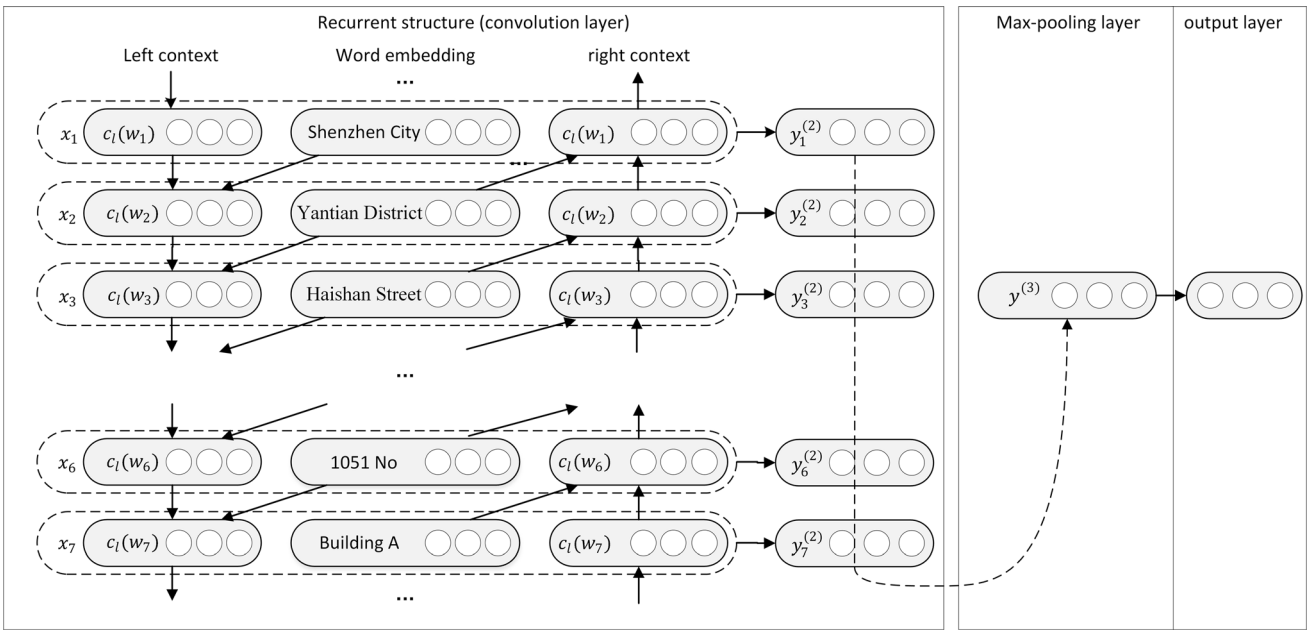

Fig. 4 The network structure of the RCNN [29]

4 Experiments

4.1 Experiment setting

4.1.1 Dataset

To evaluate the effectiveness of our proposed model, we conduct experiments using the Shenzhen Address Database proposed by Yue Lin et al. and on our Jiangsu-Hunan Address Datasets, both of which are used for address matching. At the same time, the self-labelled address element recognition dataset is used to train and evaluate the address element tagging network. Levenshtein distance [32] is a measure of similarity between two strings; the smaller the Levenshtein distance, the more similar the strings are to each other. In addition, the Jaccard similarity coefficient [33] is also a string similarity measure; the higher the Jaccard similarity coefficient, the smaller the difference between two strings.

Shenzhen address database [34]. As shown in Table 3, the Shenzhen address dataset contains 59,153 real addresses in Shenzhen, Guangdong Province, China, each containing two addresses and a label indicating whether they match or not, with 42,237 positive and negative samples each.

Jiangsu-Hunan address dataset As shown in Table 3, we generated 7600 address matching datasets for Jiangsu Province and Hunan Province based on the deviation of coordinate positions, with 3450 matching addresses and 3420 mismatched addresses.

Address element recognition dataset is composed of 36,962 address texts all over the country. These texts are labelled by us, and 30,285 samples are used as the training set; 3410 samples are used as validation set, and 3267 samples are used as the test set.

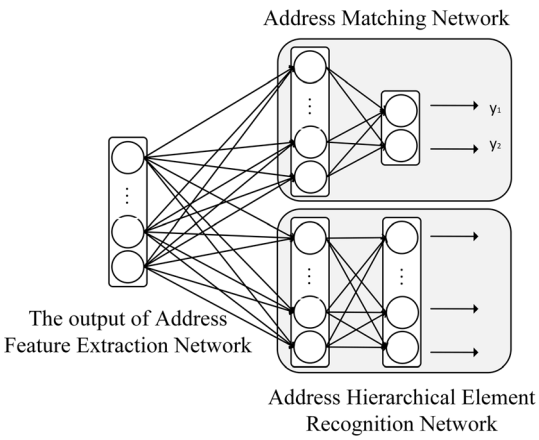

Fig. 5 The architecture of multi-task joint learning model

4.1.2 Benchmark

We compare other mainstream address matching methods to verify the validity of our model, which include Levenshtein distance, Jaccard similarity coefficient, random forest (RF) classifier [35], support vector machine (SVM) classifier [36], ESIM, and Transformer [37].

Levenshtein distance is a measure of similarity between two strings; the smaller the Levenshtein distance, the more similar the strings are to each other.

Jaccard similarity coefficient is also a string similarity measure; the higher the Jaccard similarity coefficient, the smaller the difference between two strings.

RF is a classical integrated learning algorithm for classification that contains multiple decision trees. The results of multiple decision trees jointly determine the final result of the random forest and, therefore, produce higher accuracy.

SVM is a supervised learning approach for classification. Its goal is to maximise the classification interval and thus, enhance the robustness of the model. Low-dimensional indistinguishable data can be processed by soft interval or kernel transformation, where kernel transformation is used to map the data from low-dimensional space to high-dimensional space, thus making the data distinguishable.

ESIM is a classic interaction-based deep learning model for text matching with a finely designed sequential inference structure that considers both local and global inferences. ESIM achieved the best results on the Stanford Natural Language Inference (SNLI) dataset [38]. Yue Lin used ESIM to perform local inference between address pairs, synthesised local inferences for global prediction, and achieved better results.

Transformer is a model consisting of attention mechanisms. The Transformer differs from previously existing sequence-to-sequence models in that it does not use recurrent neural networks but relies entirely on self-attentive mechanisms. It also uses positional encoding to complement the positional information of the sequences and thus, can run efficiently in parallel, achieving the best results on multiple tasks at that time.

4.1.3 Model setting

The Shenzhen Address Database uses 59,151 samples as the training set, 8487 samples as the validation set, and 16,834 samples as the test set. The Jiangsu–Hunan address dataset uses 5054 samples as the training set, 606 samples as the validation set, and 985 samples as the test set.

We used the default parameters in the CBOW algorithm in Word2Vec to train the word vectors of the address text.

Table 3 Address character similarity analysis

Attribute	Shenzhen	Jiangsu-Hunan
Total number of address pairs	59,153	7600
Number of matching address pairs	29,576	3450
Number of unmatched address pairs	29,576	3420
Average length difference for matching address pairs	3.41	7.46
Average length difference for unmatched address pairs	5.16	10.19
Average Levenshtein distance for all address pairs	10.91	11.08
Average Levenshtein distance for matching address pairs	6.43	8.89
Average Levenshtein distance for unmatched address pairs	15.39	13.17
Average Jaccard similarity coefficient for all the address pairs	0.48	0.58
Average Jaccard similarity coefficient for matching address pairs	0.70	0.51
Average Jaccard similarity coefficient for unmatched address pairs	0.25	0.66

In the address element tagging network, the maximum length is set to 50, the batch size is 32, and the learning rate is 0.001. The hidden layer dimension of bidirectional LSTM is 100. At the same time, epoch is set to 1 and 10, respectively, to compare the impact of address tagging networks with different performance on address matching task. In the multi-task learning network based on address element recognition and address matching, the maximum length is set to 50, the batch size is 64, the learning rate is 0.0001, the maximum epoch is 25, and the number of steps set by early stop is 1500. The RCNN in the address feature extraction network uses a bidirectional LSTM with a hidden layer dimension of 200 dimensions. The Adam [39] optimizer is used to optimise the training objective function.

4.1.4 Metrics

We evaluated our model using precision, recall, and $F1$ [40]. Precision is the ratio of the number of correctly classified positive samples to the number of samples judged as positive by the classifier.

$$precision = \frac{TP}{TP + FP} \tag{8}$$

where TP denotes the number of correctly classified positive samples and FP denotes the number of negative samples judged as positive samples.

Recall is the ratio of the number of correctly classified positive samples to the number of true positive samples.

$$recall = \frac{TP}{TP + FN} \tag{9}$$

where FN indicates the number of positive samples judged as negative samples.

The $F1$ score is the summed average of the precision and recall, which is defined as

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{10}$$

4.2 Comparative experiment

The results of the comparison experiment are presented in Table 4. Seven mainstream address matching methods are compared with our method on the Shenzhen Address Database and the Jiangsu Hunan Address Dataset. As shown in Table 4, our method performs better than the previous methods. Compared with traditional methods, our method avoids manual design rules and has a wider range of applications. Compared with previous machine learning and deep learning methods, we incorporate the hierarchical relationship of address elements into the deep learning model from both data and model aspects, which alleviates the gap between the semantics of addresses and address matching.

In addition, the result of the above comparison experiments shows that the deep neural network methods (ESIM, Transformer, and RCNN) outperform the machine learning methods (RF and SVM), indicating that the deep neural network-based methods can effectively learn the semantic representation of text. Deep learning can capture more valid features and contextual information than traditional methods. Moreover, deep learning avoids manual feature extraction and the design rules.

When comparing ESIM, Transformer, and RCNN, we can see that the RCNN achieves better results. This indicates that RCNN is more suitable for constructing semantic representations of addresses compared to other neural networks. We believe that the main reason is that RCNN can not only represent the current address element by surrounding address elements, but also obtain information about the most critical address elements in address matching through the pooling layer.

When comparing whether to use the address element recognition task as an auxiliary task to help address matching task learning, we find that the multi-task learning improves the model performance. We believe that an effective address representation can improve the performance of multiple related tasks, while sharing parameters weakens the network capability to a certain extent and prevents model overfitting. At the same time, the recognition ability of a single address matching model for similar strings representing the different hierarchy of information shown in Table 1 is poor, resulting in misjudgement, and adding hierarchical relationship information can effectively alleviate this semantic gap. The hierarchical relationship information of address elements is easily learned by the address element recognition task, but difficult to learn by the address matching task, probably because the address matching task is more focused on other features, which hinders the model's ability to learn the features of the hierarchical relationship of address elements. With multi-task learning, we can allow the model to eavesdrop, that is, to learn the feature using the address element recognition task.

To enable the model to learn the relationship between address elements, we also incorporate the knowledge of hierarchical relationships of address elements to enhance the model's effectiveness. We believe that introducing the knowledge of hierarchical relationships of address elements not only helps the model learn the relationships of address elements, but also narrows the search space of the model and prevents overfitting of the model.

4.3 Ablation study

The effect of model hyperparameters [41] on the experimental results on the Shenzhen Address Dataset is shown in Table 5. The best result is obtained when the number of hidden layer neurons in the RCNN is set to 200, the batch size is set to 64, the learning rate is set to 0.0001, the number of RNN layers is set to 2, and the weight (subtask weighting in Table 5) of the hierarchical element recognition network in the multi-tasking network is set to 0.1.

As shown in Fig. 6, in Experiment 2 of Table 5, $F1$ and loss values of the training set and the validation set change with the number of training rounds. When the model reaches the 20th training round, the loss of the validation set stabilises and reaches a minimum value, the $F1$ score reaches a maximum value, and the model converges.

To verify the impact of address tagging network accuracy on multi-task learning network, we use the address tagging network with poor accuracy for comparison. The accuracy of the address tagging network and the comparison results of address matching networks under corresponding conditions is shown in Table 6. It is obvious that the higher performance address tagging network can provide higher quality labels for the address element recognition network in the multi-task learning network, so better address matching results are obtained.

To verify the effectiveness of each module in our proposed model, ablation experiment is carried out. The ablation experimental results of each module are shown in Table 7, which shows the impact of each module on the overall accuracy. Among them, single address matching network refers to the construction of address matching model only based on RCNN. As shown in Table 7, the model achieves its best results when multi-task learning is used simultaneously and incorporates address element hierarchy information. These results demonstrate the effect of our proposed model.

To verify whether our proposed model can alleviate the misjudgement caused by the semantic gap shown in Table 1, 103 texts containing the semantic gap are selected. These 103 data were used for ablation experiments, and the experimental results are shown in Table 8. Among them,

Table 4 Comparison of address matching models

	Methods	Shenzhen			Jiangsu-Hunan		
		Precision	Recall	$F1$	Precision	Recall	$F1$
1	Jaccard similarity	96.0	75.0	84.0	77.9	77.8	77.8
2	Levenshtein distance	90.0	81.0	85.0	26.3	51.3	34.8
3	Word2Vec+RF [19]	92.0	89.0	91.0	83.3	77.3	80.2
4	Word2Vec+SVM [19]	87.0	81.0	84.0	83.4	83.7	83.6
5	Word2Vec+ESIM [19]	97.0	97.0	97.0	89.2	89.1	89.1
6	Word2Vec+Transformer	97.1	97.2	97.2	89.5	89.5	89.5
7	Word2Vec+RCNN	97.9	97.8	97.8	89.6	89.4	89.4
8	Word2Vec+RCNN+Multi+hierarchal relations	**99.0**	**99.0**	**99.0**	**94.3**	**94.2**	**94.2**

The best results are highlighted in bold

Table 5 The influence of hyperparameters on the results of multi-task network

	Number of hidden layer neurons	Hyperparameters				Metrics		
		Batch size	Learning rate	RNN layer	Subtask weighting	Precision	Recall	*F*1
Base	200	32	0.0001	2	0.1	98.5	98.5	98.5
1	250	32	0.0001	2	0.1	98.6	98.6	98.6
2	200	64	0.0001	2	0.1	**99.0**	**99.0**	**99.0**
3	200	32	0.001	2	0.1	98.7	98.6	98.6
4	200	32	0.0001	1	0.1	98.5	98.5	98.5
5	200	32	0.0001	2	0.2	98.7	98.7	98.7
6	200	32	0.0001	2	0.05	98.5	98.4	98.4
7	200	32	0.0001	2	0	98.3	98.3	98.3

The best results are highlighted in bold

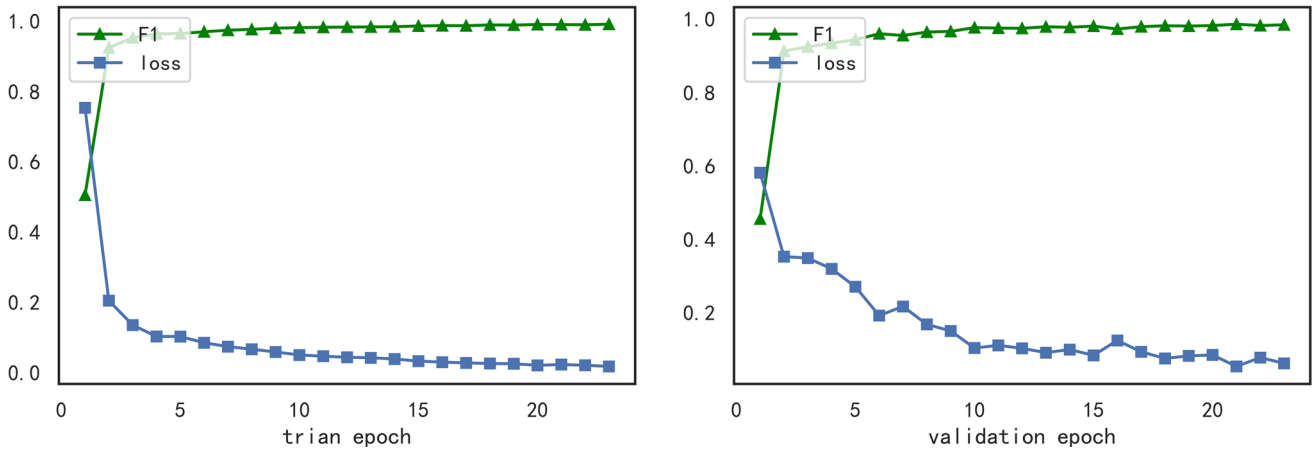

Fig. 6 Evolution of *F*1 and loss values of the training and validation sets with the number of training rounds

Table 6 Comparison of the impact of address tagging network on address matching

	Address tagging					Address matching					
						Shenzhen			Jiangsu-Hunan		
	Epoch	Accuracy	Precision	Recall	*F*1	Precision	Recall	*F*1	Precision	Recall	*F*1
1	1	96.1	92.2	94.0	93.1	98.8	98.8	98.8	90.2	90.2	90.2
2	10	**99.1**	**98.6**	**98.6**	**98.6**	**99.0**	**99.0**	**99.0**	**94.3**	**94.2**	**94.2**

The best results are highlighted in bold

Table 7 The ablation experiments of different modules

Methods			Shenzhen			Jiangsu-Hunan		
Single address matching network	Multi-task	Hierarchical relations	Precision	Recall	*F*1	Precision	Recall	*F*1
√			97.9	97.8	97.8	89.6	89.4	89.4
√	√		98.7	98.7	98.7	93.3	93.3	93.3
√	√	√	**99.0**	**99.0**	**99.0**	**94.3**	**94.2**	**94.2**

The best results are highlighted in bold

Table 8 The ablation experiments of different modules data with semantic gap

Methods			DSG		
Single address matching network	Multi-task	Hierarchical relations	Precision	Recall	*F*1
√			73.1	96.4	79.7
√	√		77.3	87.4	84.0
√	√	√	**83.3**	**98.5**	**89.2**

The best results are highlighted in bold

103 texts are referred to as Data with Semantic Gap (DSG) in the table.

As shown in Table 8, the model achieves its best results when multi-task learning is used simultaneously and incorporates address element hierarchy information. These results demonstrate the importance of hierarchical information. The inclusion of deleted hierarchical relations will greatly damage the accuracy of the model, which also shows that our proposed model can alleviate the semantic gap shown in Table 1.

5 Conclusions

Address matching is a key aspect of geocoding. Previous rule-based methods require human-designed complex templates and have limited applicability. Machine learning and deep learning-based methods ignore the hierarchical relationship between address elements, leading to misclassification. We propose a multi-task learning model for address-element recognition and address matching. First, we use a pre-trained model to identify address elements, thus solving the problem of utilising a large amount of unlabelled address data. Then, the model learns the hierarchical information of the address elements using multi-task learning. Finally, information about the hierarchical relationships between the address elements is explicitly incorporated through the CRF model. The effectiveness of our model is demonstrated by comparing previous methods on the Shenzhen Address Dataset with the Jiangsu and Hunan Address Datasets.

Our proposed model has the ability to distinguish similar characters representing the different hierarchy of information by adding the address element recognition task for joint training. It has the following prospects:

(1) It has achieved better performance compared with other existing models in detailed address recognition, which can be applied to delicate address matching.
(2) It can simultaneously predict whether two address texts match and identify the address elements. The address element recognition results can be used in:
   (a) Address matching: Adding manually designed rules to further improve the accuracy of address matching.
   (b) Model generalization: Adding customized rules to improve the model adaptability and generalizing ability.
   (c) Diversion in the express industry.
(3) It can be used not only for address matching, but also for address error correction by mapping the wrong address to the correct address through address matching.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by FL, YL, XM, JD and XL. The first draft of the manuscript was written by YL, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This research is supported by National Natural Science Foundation of China [62172449, 71790615, 62006251, 62172441], Hunan Provincial Natural Science Foundation of China [2021JJ30870, 2021JJ40783, 2020JJ4746], Changsha Municipal Natural Science Foundation [kq2014134] and National Key Research and Development Program of China [2020YFC0832700]. This work was supported in part by the High Performance Computing Center of Central South University.

Availability of data and materials The 498,294 records of the corpus derived from the Shenzhen Address Database are available in Zenodo with the identifiers https://doi.org/10.5281/zenodo.3477007. Complete corpus from the Jiangsu–Hunan Address Dataset cannot be made publicly available to protect personal information and to follow the national policy on data security.

Code availability The codes that support the findings of this study are available with the identifier(s) at the private link: https://figshare.com/s/a815fddc2429d4bd6cb2.

Declarations

Conflict of interest No potential conflict of interest was reported by the authors.

Consent to participate Not applicable.

Consent for publication Not applicable.

Ethics approval Not applicable.

References

1. Drummond WJ (1995) Address matching: GIS technology for mapping human activity patterns. J Am Plann Assoc 61(2):240–251. https://doi.org/10.1080/01944369508975636
2. Edwards SE, Strauss B, Miranda ML (2014) Geocoding large population-level administrative datasets at highly resolved spatial scales. Trans GIS 18(4):586–603. https://doi.org/10.1111/tgis.12052
3. Hu W, Dang A, Tan Y (2019) A survey of state-of-the-art short text matching algorithms. In: Tan Y, Shi Y (eds) Data mining and big data. DMBD 2019. Communications in computer and information science, vol 1071. Springer, Singapore. https://doi.org/10.1007/978-981-32-9563-6_22
4. Zhang Y, Yang Q (2021) A survey on multi-task learning. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2021.3070203
5. Lafferty J, McCallum A, Pereira FCN (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)
6. Tian Q, Ren F, Hu T, Liu J, Li R, Du Q (2016) Using an optimized Chinese address matching method to develop a geocoding service: a case study of Shenzhen, China. ISPRS Int J Geo-Inf 5(5):65. https://doi.org/10.3390/ijgi5050065
7. Koumarelas I, Kroschk A, Mosley C, Naumann F (2018) Experience: enhancing address matching with geocoding and similarity measure selection. J Data Inf Qual (JDIQ) 10(2):1–16. https://doi.org/10.1145/3232852
8. Santos R, Murrieta-Flores P, Martins B (2018) Learning to combine multiple string similarity metrics for effective toponym matching. Int J Digit Earth 11(9):913–938. https://doi.org/10.1080/17538947.2017.1371253
9. Zhou X, Li Y, Liang W (2020) CNN-RNN based intelligent recommendation for online medical pre-diagnosis support. IEEE/ACM Trans Comput Biol Bioinform 18(3):912–921. https://doi.org/10.1109/tcbb.2020.2994780
10. Yao Y, Li X, Liu X, Liu P, Liang Z, Zhang J, Mai K (2017) Sensing spatial distribution of urban land use by integrating points-of-interest and Google Word2Vec model. Int J Geogr Inf Sci 31(4):825–848. https://doi.org/10.1080/13658816.2016.1244608
11. Li H, Lu W et al (2019) Neural Chinese address parsing. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL2019)
12. Srivastava S, Vargas Munoz JE, Lobry S, Tuia D (2020) Fine-grained landuse characterization using ground-based pictures: a deep learning solution based on globally available data. Int J Geogr Inf Sci 34(6):1117–1136. https://doi.org/10.1080/13658816.2018.1542698
13. Wang Y, Wang Q, Suo D et al (2020) Intelligent traffic monitoring and traffic diagnosis analysis based on neural network algorithm. Neural Comput Appl. https://doi.org/10.1007/s00521-020-04899-3
14. Li S, Chen J, Xiang J (2020) Applications of deep convolutional neural networks in prospecting prediction based on two-dimensional geological big data. Neural Comput Appl 32:2037–2053. https://doi.org/10.1007/s00521-019-04341-3
15. Acheson E, Volpi M, Purves RS (2020) Machine learning for cross-gazetteer matching of natural features. Int J Geogr Inf Sci 34(4):708–734. https://doi.org/10.1080/13658816.2019.1599123
16. Comber S, Arribas-Bel D (2019) Machine learning innovations in address matching: a practical comparison of word2vec and CRFs. Trans GIS 23(2):334–348. https://doi.org/10.1111/tgis.12522
17. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: International conference on machine learning (PMLR)
18. Santos R, Murrieta-Flores P, Calado P, Martins B (2018) Toponym matching through deep neural networks. Int J Geogr Inf Sci 32(2):324–348. https://doi.org/10.1080/13658816.2017.1390119
19. Lin Y, Kang M, Wu Y, Du Q, Liu T (2020) A deep learning architecture for semantic address matching. Int J Geogr Inf Sci 34(3):559–576. https://doi.org/10.1080/13658816.2019.1681431
20. Chen Q, Zhu X, Ling Z, Wei S, Jiang H, Inkpen D (2017) Enhanced lstm for natural language inference. In: Association for Computational Linguistics (ACL). https://doi.org/10.18653/v1/P17-1152
21. Shi C, Cheng Y, Wang J, Wang Y, Mori K, Tamura S (2017) Low-rank and sparse decomposition based shape model and probabilistic atlas for automatic pathological organ segmentation. Med Image Anal 38:30–49. https://doi.org/10.1016/j.media.2017.02.008
22. Zhou X, Liang W, Wang K, Wang H, Yang L, Jin Q (2020) Deep-learning-enhanced human activity recognition for Internet of healthcare things. IEEE Int Things J 7(7):6429–6438. https://doi.org/10.1109/jiot.2020.2985082
23. Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw 18(5–6):602–610. https://doi.org/10.1016/j.neunet.2005.06.042
24. Schmidhuber J, Hochreiter S (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
25. Panchendrarajan R, Amaresan A (2018) Bidirectional LSTM-CRF for named entity recognition. In: PACLIC
26. Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 77(2):257–286. https://doi.org/10.1109/5.18626
27. McCallum A, Freitag D, Pereira FC (2000) Maximum entropy Markov models for information extraction and segmentation. In: Proceedings of the Seventeenth International Conference on Machine Learning (ICML)
28. Nadeau D, Sekine S (2007) A survey of named entity recognition and classification. Lingvisticae Investig 30(1):3–26. https://doi.org/10.1075/li.30.1.03nad
29. Lai S, Xu L, Liu K, Zhao J (2015) Recurrent convolutional neural networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence
30. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics, in proceedings of machine learning research, pp 315–323
31. Collobert R, Weston J (2008) A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning, pp 160–167
32. Levenshtein VI (1966) Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady, pp 707–710
33. Jaccard P (1908) Nouvelles recherches sur la distribution florale. Bull Soc Vaud Sci Nat 44:223–270
34. Yue L, Mengjun K (2019) Shenzhen address corpus (part) (Version v1.0). Zenodo. https://doi.org/10.5281/zenodo.3477633
35. Breiman L (2001) Random forests. Mach Learn 45(1):5–32. https://doi.org/10.1023/A:1010933404324

36. Hearst MA, Dumais ST, Osuna E, Platt J, Scholkopf B (1998) Support vector machines. IEEE Intell Syst Appl 13(4):18–28. https://doi.org/10.1109/5254.708428
37. Vaswani A, Shazeer N, Parmar N et al (2017) Attention is all you need. In: Proceedings of the 31st international conference on Neural Information Processing Systems (NIPS'17), pp 6000–6010
38. Bowman SR, Gauthier J, Rastogi A et al (2016) A fast unified model for parsing and sentence understanding. In: Proceedings of the 54th annual meeting of the Association for Computational Linguistics (ACL), pp 1466–1477. https://doi.org/10.18653/v1/P16-1139
39. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: International Conference on Learning Representations (ICLR 2015)
40. Powers D (2011) Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation. J Mach Learn Technol 2:37–63
41. Lujan-Moreno GA, Howard PR, Rojas OG, Montgomery DC (2018) Design of experiments and response surface methodology to tune machine learning hyperparameters, with a random forest case-study. Expert Syst Appl 109:195–205. https://doi.org/10.1016/j.eswa.2018.05.024