Digital Communications and Networks 9 (2023) 887–895

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.keaipublishing.com/dcan

# A label noise filtering and label missing supplement framework based on game theory

Yuwen Liu [a], Rongju Yao [b], Song Jia [c], Fan Wang [f], Ruili Wang [d], Rui Ma [e], Lianyong Qi [a,*]

[a] College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China
[b] Weifang Key Laboratory of Blockchain on Agricultural Vegetables, Weifang University of Science and Technology, Shouguang, China
[c] China Unicom Taian Branch, China
[d] School of Natural and Computational Sciences, Massey University, New Zealand
[e] General Education Department, Shandong First Medical University, Shandong Academy of Medical Sciences, Tai'an, 271 000, China
[f] College of Computer Science and Technology, Zhejiang University, Hangzhou, China

ARTICLE INFO

*Keywords:*
Label noise
FastText
Cosine similarity
Game theory
LSTM

ABSTRACT

Labeled data is widely used in various classification tasks. However, there is a huge challenge that labels are often added artificially. Wrong labels added by malicious users will affect the training effect of the model. The unreliability of labeled data has hindered the research. In order to solve the above problems, we propose a framework of Label Noise Filtering and Missing Label Supplement (LNFS). And we take location labels in Location-Based Social Networks (LBSN) as an example to implement our framework. For the problem of label noise filtering, we first use FastText to transform the restaurant's labels into vectors, and then based on the assumption that the label most similar to all other labels in the location is most representative. We use cosine similarity to judge and select the most representative label. For the problem of label missing, we use simple common word similarity to judge the similarity of users' comments, and then use the label of the similar restaurant to supplement the missing labels. To optimize the performance of the model, we introduce game theory into our model to simulate the game between the malicious users and the model to improve the reliability of the model. Finally, a case study is given to illustrate the effectiveness and reliability of LNFS.

## 1. Introduction

The Internet of things [1], mobile crowdsourcing [2] and other new technologies have been widely popularized and applied in various industries, giving birth to massive data. The formation of such data is often inseparable from the participation of people. The hidden value of such data is huge. For many industries, how to use the big data is the key to win the market competition. With the continuous advancement of computer equipment, people have been able to deal with the large-scale data. However, as most of the data come from the network, its authenticity will be reduced by people's malicious behaviors. For example, in machine learning [3], a common work is to use labeled data to train neural networks for classification, regression or other purposes [4]. At this time, the quality of data will directly affect the model performance. The labels are often assigned by humans, and the task of labeling the data is time-consuming, laborious, and expensive. Some malicious marking behaviors will lead to massive label noise. Therefore, the influence of noise labels must be considered in practical application. We need a model that can accurately identify label noise to actively eliminate the impact of noise data.

Currently, people are happy to mark some popular locations to prove that they have been there. At the same time, users will leave their comments and label the restaurants where they have been to show how much they like the restaurants and help others to judge whether the restaurants are in line with their interests. Then other users can find their preferred restaurant by searching for different labels. An app such as Dianping [5] can achieve this function. However, the quality of such labels is uneven as they are labeled by various people. Some malicious users will add invalid labels to the location, making some labels accurate and authentic, some wrong or fuzzy [6]. Therefore, a very huge challenge is how to select a correct label for the location to help other users with real needs to search and filter out the inaccurate noise labels. We propose a label noise

* Corresponding author.
*E-mail addresses:* yuwenliu97@gmail.com (Y. Liu), yaorongju8@wfust.edu.cn (R. Yao), 18653809711@163.com (S. Jia), fanwang1997@gmail.com (F. Wang), Ruili.WANG@MASSEY.AC.NZ (R. Wang), 49803937@qq.com (R. Ma), lianyongqi@qfnu.edu.cn (L. Qi).

https://doi.org/10.1016/j.dcan.2021.12.008
Received 14 March 2021; Received in revised form 4 December 2021; Accepted 29 December 2021
Available online 4 January 2022

filtering framework to solve the above problems.

First of all, the labels are diverse because they are marked by various people according to their own understanding. And some malicious users will deliberately add false labels to competitors to damage the reputation of other stores. Therefore, we need a model that can identify the incorrect labels timely and generate the correct label for the location. The most appropriate label for a location must cover most of the meanings of other general labels. Therefore, we first assume that the label most similar to all other labels in the location is the most representative. Secondly, since the computer cannot process the text directly, we need to take an embedding method to transform the labels into the form of word vectors. Then an appropriate label similarity measurement method is adopted to judge the similarity between each label. And we select the label which is most similar to all other labels as the unique label of the location. To enhance the recognition ability, we use the idea of game theory. We first identify the unique label, then simulate the attacker, actively increase the label noise to the location, and then use the model to identify the unique label. It is worth mentioning that while there are many labels for some places, which need to filter the label noise, there are still some places still lack labels. For the location without a label, we consider the similarity of locations to make an appropriate label supplement. Finally, we find the best label for each location so that the data can play the maximum value in practical application. The main thrust of our research is as follows.

(1) We propose a label noise filtering and missing label supplement model. We use FastText to train appropriate word vectors for all user text labels, and use cosine similarity to calculate the similarity between the labels. Then, the label most similar to all other labels is selected as the unique label of the location, and the noise of other labels is deleted. For locations without labels, we can find similar places through users' comments. The label of the similar place is used as the label of that place.
(2) In order to continuously optimize the performance of our proposed model, we apply the evolutionary game theory to the process of location label filtering and discuss the detailed simulation scenarios of game modeling.
(3) We apply our evolutionary game method to a location labels noise filtering scheme and make a case study. The results show that our method can describe a detailed evolution process, and our model can achieve better performance.

The rest of this paper is arranged as follows. Section 2 introduces the research work related to game theory and label noise filtering. Section 3 introduces the research motivation of our work. Then Section 4 introduces the related knowledge involved in this study, and Section 5 introduces the label noise filtering method we propose. Finally, Section 6 and Section 7 present a case study and conclusions of the proposed method respectively.

## 2. Related work

### 2.1. Game theory

Game theory is a new branch of modern mathematics. It is a mathematical theory and method for studying phenomena with struggle or competition. At present, the combination of game theory and various applications has made great contributions to social progress [7,8]. He et al. [9] proposed to introduce game theory into an integrated energy system. It can solve problems in the energy supply side, distribution network, demand side and common planning and dispatching problems in integrated energy systems. Game theory is also used in solving power system problems. Abapour et al. [10] made a systematic summary of the application of game theory in power systems. In addition, to improve the performance of radar, it is necessary to control the polarization state of transmit waveforms. Zhang et al. [11] proposed a framework to find the optimal transmit polarizations. The polarization problem of the radar system is transformed into a zero-sum game between the radar and the jammer, so as to improve the anti-jamming performance of the system. Traditional game theory assumes that all players are absolutely rational, which is difficult to guarantee in most cases. Therefore, the evolutionary game theory provides new ideas to solve this problem [12]. At this time, people are no longer absolutely rational players, and the evolutionary game has begun to develop rapidly. Lin et al. [13] proposed to use the evolutionary game theory to study long-term green behavior. They have fully considered the price elasticity, market competition, green investment and green relevance to formulate a price decision model. Shipping companies can maximize their profits while reducing environmental pollution. However, the evolutionary game theory has not been applied to noise filtering of location labels. This will be an appropriate strategy to optimize the noise filtering model.

### 2.2. Problems in label noise filtering

Data with label noise will affect the performance of many tasks, such as prediction [14–17], security [18,19] and privacy protection [20] [–] [22]. Datasets with label noise have many negative effects on classifiers. At present, there have been many studies focusing on mitigating the negative effects of label noise. Some studies focus on establishing a robust model to achieve better model performance in the data with label noise [23,24]. Some research aims to distinguish the label noise from the real labels in the dataset and to get a better dataset for model training. Next, this section will introduce the research progress on label noise.

Label noise has hindered some sampling techniques [25,26]. Researchers often use the Synthetic Minority Over-sampling Technique (SMOTE) method when they achieve unbalanced classification tasks. In the process of data sampling, once the collected data contains too much label noise, it is difficult for SMOTE to achieve the desired effect. Chen et al. [27] proposed an adaptive robust SMOTE, which uses relative density to measure the local density of a small number of samples, and then distinguishes features according to the relative density for imbalanced classification of label noise. Establishing a model in the presence of label noise, Huang et al. [28] predicted the fault of rolling bearing in the wind turbine gearbox. They faced the problem of insufficient data and excessive noise labels and used an improved label-noise robust Auxiliary Classifier Generative Adversarial Network (rAC-GAN) driven by the limited data. This method improved the generalization ability of the model in practical operation and had good robustness. In addition to some text label noises, there are also image label noises [29]. It is a difficult task to extract information from the image and then label the image. For example, extracting buildings from satellite images requires extensive accurate data to train the deep neural network and to realize the automatic generation of building labels. But the model training needs a lot of accurate data, and the image label noises will hinder the model training. To cope with this challenge, Zhang et al. [30] proposed a general label noise adaptive neural network framework to judge the relationship between real labels and noise labels. This framework helps to automatically generate image labels and reduce manual annotation. In addition, Cai et al. [31] proposed a method of uploading data efficiently under the condition of privacy protection, and Qi et al. [32] proposed a model of data fusion in the smart city scene. Zhang et al. [33] proposed an improved noise loss correction algorithm for learning from noise labels.

As for the distinction of label noise, there are also diverse studies. In industrial informatics, data is high-dimensional, noisy and easily mislabeled. Guan et al. [34] integrated the traditional two-step method, and then proposed a Sequential Ensemble Noise Filter (SENF) to generate noise fractions for each feature instance, so as to identify the noise label. In terms of classification technology, Quadratic Discriminant Analysis (QDA) is often used. However, label noise and measurement noise will seriously damage the prediction ability of the model. Vrankx et al. [35] proposed a new real-time robust QDA method which takes the most atypical observation as label noise. This method can not only distinguish

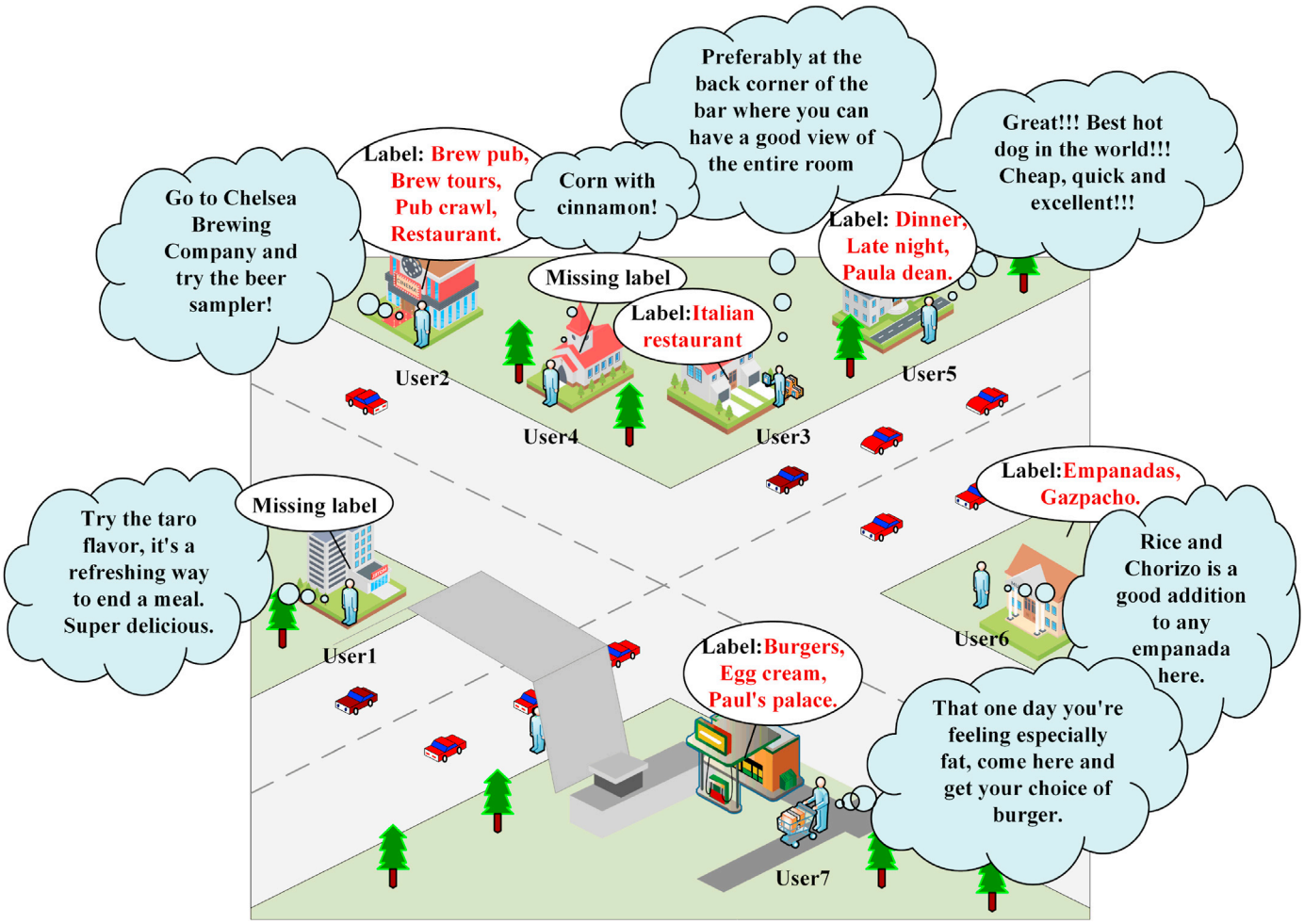


**Fig. 1.** Multiuser dynamic: an example.

label noise, but also recognize the label noise in real time. However, the above methods are less effective when faced with complex multi-classification problems. To make up for this shortcoming, Xia et al. [36] proposed a label noise filtering framework for multi-classification. In the multi-class environment, the filtering method of multiclass Complete Random Forest (mCRF) and multi-class relative density is adopted. Considering the parameter optimization, a parameter optimization method using the 2-means clustering algorithm is proposed. This framework can not only filter label noise, but also deal with the problem of data imbalance in multi-classification. In addition, Feng et al. [37] specially proposed a label noise cleaning technology. This technique used three different classifiers (bagging, AdaBoost and k-Nearest Neighbor (k-NN)) to recognize label noise.

In a word, the research on label noise, whether it is text label noise or image label noise, will have a negative impact on the actual model application. At present, the research on label noise has made some progress. However, the accuracy of identifying labels from massive labels is still not high enough in various industries. In addition, the conditions of different scenarios are different, and it is difficult to have a perfect framework to achieve label noise filtering in all scenarios [38]. In order to meet this challenge, we propose a noise filtering framework for text labels. This framework can not only filter out the redundant label noises, but also supplement the missing labels. To introduce our framework in more details, we select a scene of location label selection to describe the noise filtering framework of text labels in detail.

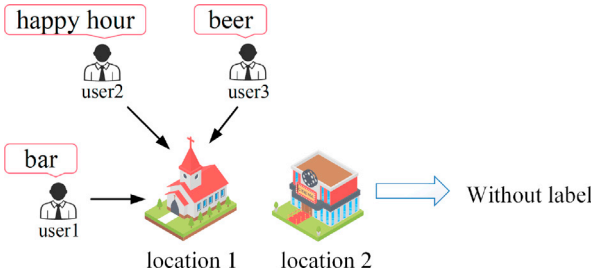


**Fig. 2.** Label noise and label missing.

## 3. Motivation

In the Location-Based Social Network (LBSN), users often check in at different locations. For example, users will mark the restaurant they have been to, and leave their comments about the restaurant. As shown in Fig. 1, user 1 gives a comment “Try the taro flavor, it's refreshing way to end a meal; Super delicious” after eating in an English restaurant, while user 2 who likes lounge bar gives a comment like “Preferably at the back corner of the bar where you can have a good view of the entire room”. In addition to leaving comments on the restaurant, LBSN also provides the function to give labels on restaurants to facilitate the user retrieval. In Fig. 1, the location accessed by user 2 has four labels, while the location accessed by user 4 has no label. Users can take the initiative to label the restaurant as they think, therefore location labels are subjectively added by a variety of users and there may be malicious users who deliberately

add wrong labels. We take a real application as an example to illustrate the influence of noise labels on the performance of the model as shown in Fig. 2. It can be seen from the figure that location 1 has three different labels. If user 2 adds an erroneous label "happy hour" to location 1, other users may not be satisfied when searching for location 1 according to the erroneous label. And location 2 has no user to add labels. This situation will lead to the situation that some labels are accurate, while some labels are redundant or inappropriate. There are also some locations that are not labeled by users. It is a serious obstacle for users to search a location accurately. At present, there are few studies on location label noise filtering in LBSN.

To solve the above problems, we propose a framework of label noise filtering and missing label supplement. (1) We assume that the label most similar to all other labels in the location is the most representative. For each location with redundant labels, cosine similarity is used to calculate the similarity between each label and all other labels, and then the sum of similarity is used as the feature of the keyword. Finally, the largest label of all features is taken as the label of the location. (2) For locations missing the label, we use the user's comments to find a similar location by judging the similarity of comments, and then use the label of the similar location to assign a label to the location. (3) To optimize the performance, we apply the evolutionary game theory to our scheme. By continuously increasing the label noise and performing corresponding noise filtering, the proposed model is optimized.

## 4. Preliminaries

### 4.1. Problem formulation

Users $U = \{u_1, u_2, \ldots, u_M\}$ will visit different locations $V = \{v_1, v_2, \ldots, v_N\}$ and post their comments $C_V = \{c_1, c_2, \ldots, c_O\} \in C$. The comments often imply the users' attitude towards the locations. For different locations $v_i \in V$, users will label them subjectively with $L_V = \{l_1, l_2, \ldots, l_Q\} \in L$. In particular, $M$, $N$, $O$ and $Q$ refer to the total number of users, locations, comments and labels respectively. However, the user's subjective labeling of locations leads to the decrease of the reliability of labels. Some location labels are accurate, which is conducive to other users' retrieval. Meanwhile, some labels are added by users arbitrarily, which will mislead users' search results. Still, some locations have not been labeled by any users, which is also not conducive to being retrieved by other users.

**Definition 1.** Label noise filtering: We need to select a unique label for each location $v_j \in V(1 \le j \le N)$, i.e., for each label $l_k \in L(1 \le k \le Q)$, if $k \ge 2$, it is proved that there is label noise. We need to identify and delete the location labels.

**Definition 2.** Label supplement: For locations without labels, i.e., $l_k \in L(k = 0)$, we need to find similar locations according to the users' comments between locations, and use the labels of similar locations to supplement the label for locations without labels.

### 4.2. FastText

Informal discussions on social platforms have accumulated a large body of knowledge in the form of natural language text [39]. For example, text descriptions can be used to mine and create compatible mashups [40]. There are many ways to represent the text, such as Term Frequency-Inverse Document Frequency (TF-IDF) and TextRank based on one-hot encoding [41], Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) based on the topic model [42], and fixed representation based on word vectors, such as world2vec [43] and FastText [44]. One-hot coding is a waste of memory and cannot capture the relationship between words. LDA is suitable for topic modeling and infers the subject topic distribution of the document from the text. Obviously, it needs a lot of information to train the document topics which are not suitable for labeled data. Word2vec does not share any parameters. Every word is learned according to the context. Word2vec can't deal with any word haven't been met in training. For example, word2vec cannot find the connection between words with the same root, such as "eat" and "eaten". But FastText can deal with such problems. Its key idea is to use the internal structure of words to obtain the vector representation of words. When each location contains too many labels or lacks a label, we need to find the most appropriate label for the location, so that users can accurately retrieve it. Since the computer cannot deal with the text information directly, the first step we consider is to train a certain word vector for each label. FastText is a tool for the word vector calculation and text classification, and its advantages are also obvious. In text classification tasks, FastText can often achieve the same accuracy as a deep network, but the training time is much faster than a deep network.

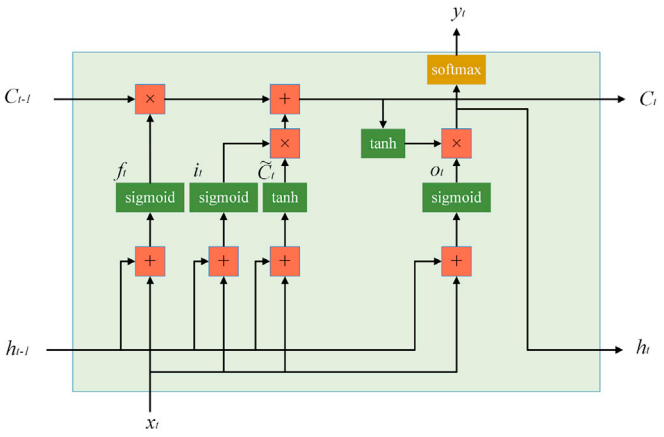


**Fig. 3.** The structure of LSTM.

### 4.3. Long short-term Memory(LSTM)

The Long Short-Term Memory (LSTM) model is a special kind of Recurrent Neural Network (RNN) [45]. This model can better realize the function of memory and forgetting, and can capture the long-term and short-term dependence between input data. The following Fig. 3 shows the structure of the LSTM gate. And the LSTM is calculated as follows:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

## 5. Framework for label noise filtering and missing label supplement:LNFS

This section will introduce the implementation process of our proposed label noise filtering and missing label supplement framework in detail.

### 5.1. Training the word vector of labels

Since our model cannot process the text-type label data directly, the first step we need to do is to transform the text-type label into vector. At present, there are many methods for representing word vectors, such as one-hot encoding. However, this method requires high-dimensional

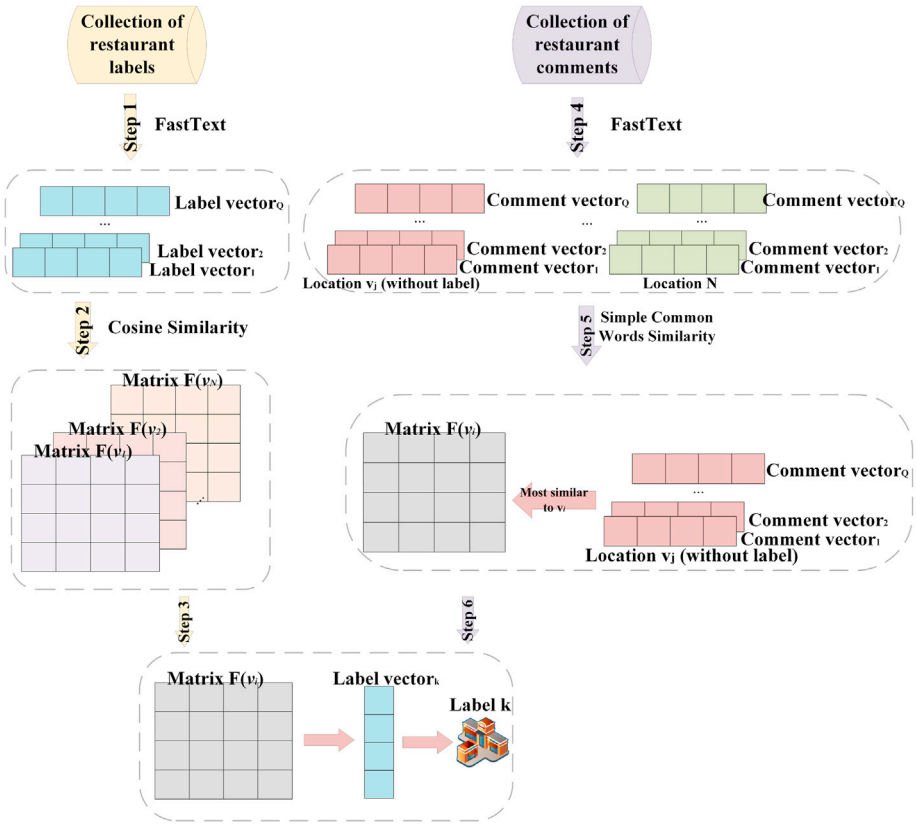


**Fig. 4.** The technical architecture of our LNFS.

space in the case of too many words, which will slow down the patrol speed of our model [46]. Moreover, the word vectors trained by one-hot encoding method are not related to each other, so we do not use one-hot encoding in this framework. FastText is used to train the word vector of the label. There are some advantages to use FastText. First, FastText training is fast, and it can achieve the same training results as the neural networks. Second, the word vectors trained by FastText are related to each other, and the word vectors of similar words are often more similar. Thirdly, FastText adds the n-gram feature, which can be used for character level training. For example, in the case of “apple” and “apples”, other word embedding methods cannot find the relationship between the two words, but FastText can do well.

### 5.2. Label noise filtering

Some locations are labeled randomly by users, or labeled intentionally by malicious users, and such unreliable labels will have a bad impact on the reputation of the place. Therefore, we propose the assumption that the labels most similar to all other labels of the location are the most representative. Many users have added labels to the site, so the most appropriate label is often a collection of public opinions. Choosing a unique label is also to represent a location as accurately as possible. In the subsequent user interest mining, the unique label can be used as the user's interest. When multiple labels are retained, there may have some duplicate labels, or there may have some labels that do not match the location very well. This will lead to information redundancy. But it doesn't mean that selecting multiple labels is unreasonable. A location with multiple labels is easier to be searched. Therefore, different number of labels can be reserved according to different scenarios. In our research scenario, the label most similar to all other labels of the site is selected as the only label. Next, take a specific location and its labels as an example to introduce our method. Suppose a place contains four labels. First, we extract the word vectors of the four labels. Then, we use cosine similarity [47] to calculate the similarity between the four labels. The calculation formula of cosine similarity is shown in (7) and (8).

| | label 1 | label 2 | label 3 | label 4 | sum |
|---|---|---|---|---|---|
| label 1 | 0 | 0.27 | 0.09 | 0.092 | 0.452 |
| label 2 | 0.27 | 0 | 0.03 | 0.04 | 0.34 |
| label 3 | 0.09 | 0.03 | 0 | 0.97 | 1.09 |
| label 4 | 0.092 | 0.04 | 0.97 | 0 | 1.102 |

**Fig. 5.** Location containing four labels for noise filtering: an example.

$$F(v_i) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}|*|\vec{y}|} \tag{7}$$

$$\cos\theta = \frac{\sum_{i=1}^{n}(x_i y_i)}{\sqrt{\sum_{i=1}^{n}(x_i)^2}\sqrt{\sum_{i=1}^{n}(y_i)^2}} \tag{8}$$

where $\vec{x}$ represents the target label vector and $x_i$ represents the vector elements. $\vec{y}$ represents other label vectors of the location and $y_i$ represents the vector elements. Furthermore, we get the similarity matrix of the location, take the sum of the similarity between each label and other labels as the feature of the label, and get the feature vector of the location. Among them, the label with the largest eigenvalue is most similar to other locations, and this label is selected as the unique label of the location. The process is shown in Fig. 4 step 1–3. First, we get the label

vector through FastText, and then we use cosine similarity to calculate the eigenvalue of all label vectors in each location. Finally, the label with the largest eigenvalue is selected. As shown in Fig. 5, there are four labels for a location. After the similarity between labels is obtained by using cosine similarity, the similarity matrix is obtained. By summing the similarity matrix of each location, we get the eigenvalues of the four labels. It can be seen that the feature value of label 4 is the largest, so we choose label 4 as the label of the location. In addition, the specific experimental process is shown in Algorithm 1.

**Algorithm 1** Label noise filtering

**Require:** Locations $V$ and their labels $L$
**Ensure:** Unique label $\{l_k\}$ for each location $v$
//Select a location with labels ≥ 3
$V_f = \emptyset$
**for** each location $v$ in $V$ **do**
  **if** $L_V = \{l_1, l_2, , l_Q\}, Q \geq 3$ **then**
    Add $v$ in $V_f$
  **end if**
**end for**
// Establishing similarity matrix
Training label vectors with FastText
**for** each location $v$ in $V_f$ **do**
  Calculate $F(v)$ by (7)
  Sum each line $vl_i$ of $F(v)$
  Find the max $vl_i$
**end for**
**return** label $\{l_k\}$

However, the label noise filtering model requires optimization. Locations in LBSN constantly receive labels added by users, including incorrect labels added by malicious users. Therefore, in order to make our model have a better noise recognition ability, we simulate a group of users to add labels. The users include ordinary users and attackers. The ordinary users add normal labels, and the attackers add wrong labels. The model is used to recognize label noise again. Through many times of gaming, the model can achieve the purpose of model optimization.

### 5.3. Missing label supplement

In the LBSN, many locations are still not labeled by users. These locations may be rarely visited by people, or they may be newly built, or users are too reluctant to add labels. In any case, the lack of labels will make it difficult for other users to search these restaurants, which is not conducive to the development of restaurants. Although these restaurants lack labels, they collect users' comments. Since the restaurants that users leave comments often have the same type of location, our proposed method is to add a label for the location by judging a labeled location similar to it. The process is shown in Fig. 4 step 4–6. This involves how to judge the similarity between user comments. Due to the large number of user comments, we choose a more complex method which is time-consuming and labor-consuming. Therefore, we choose the simple common word similarity method. Intuitively, we first count the total number of characters of the two locations under comparison, then count the total number of characters of the common words about the location, and finally divide the common words by the number of characters in the longest document to get the similarity measurement value.

Suppose that for location $A = \{c_1, c_2, \ldots, c_Q\} \in C$ without a label, we compare it with other locations with a unique label to find a similar location. Let's assume there is a location $B = \{c_1, c_2, \ldots, c_P\} \in C$, then the text similarity of location $A$ and $B$ is calculated by the following equations (9) and (10).

$$S(v_{ij}) = \frac{A \cap B}{Q} (Q \geq P) \tag{9}$$

$$S(v_{ij}) = \frac{A \cap B}{P} (Q < P) \tag{10}$$

Where $A$ represents the comments words about location $A$, $B$ represents the words about location $B$, $Q$ represents the total number of words about location $A$, and $P$ represents the total number of words about location $B$. By judging the similarity between the comments on different locations, we can find the label of locations which are similar to the location without a label and then add the label to the location. In addition, the specific experimental process is shown in Algorithm 2.

**Algorithm 2** Missing label supplement

**Require:** Users $U$, locations $V$, labels $L$ and comments $C$
**Ensure:** Unique label $\{l_{k2}\}$ for every missing location v
//Select a location without label
$V_s = \emptyset$
**for** each location $v$ in $V$ **do**
  **if** $L_V = \emptyset$ **then**
    Add $v$ in $V_s$
  **end if**
**end for**
//Finding similar locations
Training label vectors with FastText
**for** each location $v$ in $V_s$ **do**
  **for** each comments $c$ in $C$ **do**
    Calculate $S(v_{ij})$ by (9) (10)
    $l_v = l_v\_sim_v$
  **end for**
**end for**
**return** label $\{l_{k2}\}$

Finally, we summarize the proposed label noise filtering and missing label supplement framework. For a location with only one label, we default that this label is the only label of the location. For locations with three or more labels, we assume that the label most similar to other labels about the location is the most representative, and then we use cosine similarity to calculate the similarity matrix of each location, and sum up the eigenvalues of each label. The label with the largest eigenvalue is taken as the unique label of the location. For the location with missing label, we use the simple common word similarity method to find a similar location with a unique label, and make its label as the label of the location. In particular, the above solution cannot be used for locations with two labels. Therefore, we first find a similar location according to the common word similarity method. At this time, the location has three labels, and finally determine the unique label according to the label noise filtering method.

### 5.4. Application of location label

After choosing the right label for each location, we need to prove the performance of the LNFS framework. Because location labels are added by network users, there is no fixed standard for location labels. There is no way to compare the noise filtered labels with the standard location labels. We can measure the practicability of our method from the perspective of application. We first select the LBSN location related scenes, in which the users will check in. Therefore, being able to predict the locations that users may visit next and recommend them to users in a timely manner is very helpful to the development of the smart city industry [48,49]. But it is very difficult to find the user's check-in preference only according to the specific location. Therefore, we choose the location label as the user, preference to predict the user, check-in behavior.

We input the check-in information of users who have performed label

**Table 1**
User check-in data.

| Users | Location |
|---|---|
| 35 443 | 899 |
| 24 973 | 42 406 |
| 14 860 | 177 |
| … | … |
| 133 970 | 41 096 |

**Table 2**
Location-label data.

| Location | Labels |
|---|---|
| 15 | |
| 20 | bakery,bar,barbeque,bbq,bistro, boutique, bravo, bravoandy, celebrity sighting, coffee |
| 25 | colombian, cupcake, fish tacos, grilled corn, long wait, pork, rice and beans |
| … | … |
| 558 276 | donut shop |

**Table 3**
Labels: vectorization implementation.

| Label | Vector |
|---|---|
| bakery | [0, 0, 1, 0, 1, 1, 1, 0, 1, 1] |
| bbq | [1, 0, 0, 0, 0, 1, 1, 1, 0, 0] |
| barbeque | [1, 1, 0, 1, 0, 1, 1, 0, 0, 0] |
| … | … |
| 133 970 | [1, 0, 0, 0, 0, 1, 1, 1, 0, 0] |

noise filtering and missing label supplement into LSTM for training according to the time sequence so as to predict the label categories that users may visit next. Then, the check-in information of users who do not perform label noise filtering and missing label supplement is input into an LSTM model for training, and the label categories that users may visit next are predicted. Comparing the performance of these two LSTM models, we can well prove the advantages of our proposed LNFS framework.

### 5.5. A case study

In this section, we use a specific case study to demonstrate the reliability of the LNFS framework.

#### 5.5.1. Step 1: embedding

The label data of the location is processed into a word vector. The data we use includes the user-location data pairs shown in Table 1, and each data pair indicates that a user has checked in once at a certain location. And the users will leave comments in their check-in positions. The location-labels data is shown in Table 2, which represents all user-added labels received at each location. We transform all label information into a 300-dimension vector through FastText for subsequent processing, which is shown in Table 3.

#### 5.5.2. Step 2: label noise filtering

For the location $v_i$ with more than 3 labels, we need to select the most representative label. Therefore, we use (1) to find the similarity between all the label vectors of $v_i$. The similarity matrix of $v_i$ can be obtained. Then, each row of the similarity matrix is added to get the eigenvalue of each label, and the label with the largest eigenvalue is the unique label of $v_i$. Finally, Fig. 6 shows the label noise filtering process. Then, the attackers add malicious labels, and the LNFS model performs label noise filtering. The performance of the model is optimized by the game between the model and the attackers.

#### 5.5.3. Step 3: missing label supplement

Each location contains comments left by users. For location $v_j$ without a label, we find a similar location by judging the similarity of comments, and use the label of the similar location to supplement the location without a label. In addition, for the location with two labels, the similarity between the two labels is the same, so a unique label cannot be found by calculating the eigenvalue. We first use the similarity of comments to find a label of the similar location, and then use the label filtering method to select a unique label. Finally, Fig. 7 shows the missing label supplement process.

#### 5.5.4. Step 4: user check-in location prediction

Through the previous step1-3, we have been able to select a unique label for each location. We need to verify the performance of our proposed LNFS framework. As the labels are added by different people, no

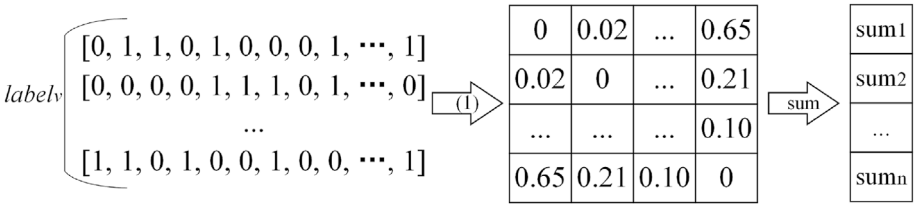


**Fig. 6.** Label noise filtering process.

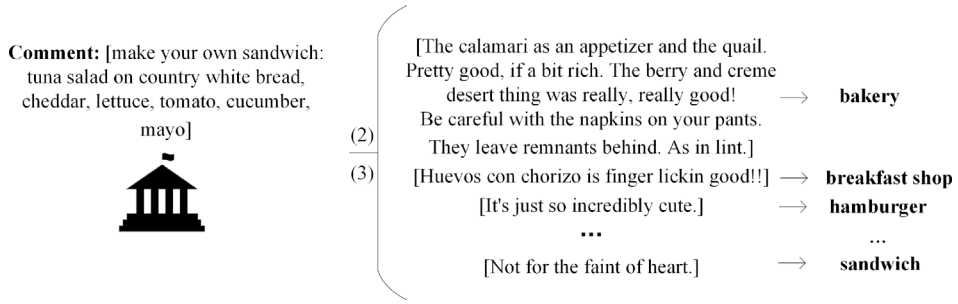


**Fig. 7.** Missing label supplement.

**Table 4**
Comparison between predicted labels and actual labels.

| User | Predicted labels | Actual labels | After_Accuracy |
|---|---|---|---|
| 1 | coffee, bbq, bar, coffee | Coffee, barbeque, bar, coffee | 75% |
| 2 | bistro, brunch, vegan friendly, beer | college, brunch, burger, beer, | 50% |
| 3 | crunk, chicken, vegetarian, burger | crunk, black tea, balls, hot dog | 25% |
| … | … | … | … |
| 133 970 | private dining, bakery, beer, soups | record shop, beer, paella, bakery | 50% |
| User | Predicted labels | Actual labels | Before_Accuracy |
| 1 | bbq, haunted, brunch, beer | Coffee, barbeque, pizza, coffee | 0% |
| 2 | coffee, diner, vegan friendly, french | college, brunch, burger, bbq | 0% |
| 3 | crepes, boy, ice cream, balls | black tea, art galleries, balls, hot dog | 25% |
| … | … | … | … |
| 133 970 | cookies, tapas, cozy, tacos | record shop, beer, paella, bakery | 0% |

standard can directly prove the performance of our framework. To meet this challenge, we use a specific application: we input the data before and after noise filtering into the LSTM model for training, then we use the trained model to predict the user, check-in behavior and predict the labels of the places they may visit next. The prediction results of the model are shown in Table 4. The case study also proves that our LNFS framework is effective and reliable.

## 6. Conclusions

Aiming at solving the problem of unreliable labels in LBSN, we propose a framework of label noise filtering and missing label supplement. To solve the problem of label noise filtering, we first use FastText to convert the restaurant's labels into the vector form, and then use cosine similarity to get the label which is most similar to other labels of the location. To solve the problem of label missing, we use the simple common word similarity to judge the similarity of users' comments, and then use the label of the similar restaurant to supplement the missing labels. In order to increase the reliability of the model and protect the reputation of the location from being affected by malicious labels, game theory is used to simulate malicious users adding wrong labels, and LNFS model is constantly used to identify label noise. Finally, we apply a case study to demonstrate the effectiveness and reliability of our method.

In future work, we will apply this research to point-of-interest recommendation. According to the unique label of the location, the users' interest preference for the locations can be mined. Then we can recommend interesting locations to users.

## Declaration of competing interest

We declare that there are no conflicts of interest regarding the submission and the manuscript has not been submitted to other journals or conferences for consideration.

## Acknowledgments

This study was supported by the National Natural Science Foundation of China (No. 61872219) and the Natural Science Foundation of Shandong Province (ZR2019MF001).

## References

[1] Z. Cai, Z. He, Trading private range counting over big iot data, in: 2019 IEEE 39th International Conference on Distributed Computing Systems, 2019, pp. 144–153.
[2] Y. Wang, Y. Gao, Y. Li, X. Tong, A worker-selection incentive mechanism for optimizing platform-centric mobile crowdsourcing systems, Comput. Network. 171 (2020) 107144.
[3] H. Huang, J. Lin, L. Wu, B. Fang, Z. Wen, F. Sun, Machine learning-based multi-modal information perception for soft robotic hands, Tsinghua Sci. Technol. 25 (2) (2020) 255–269.
[4] Z. He, M. Yang, H. Liu, L. Wang, Calibrated multi-label classification with label correlations, Neural Process. Lett. 50 (2) (2019) 1361–1380.
[5] F. Wei, M. Choi, X. Shang, A study on the corporate culture of dianping, Int. J. Adv. Cult. Technol. 7 (4) (2019) 69–75.
[6] N. Bhardwaj, P. Sharma, An advanced uncertainty measure using fuzzy soft sets: application to decision-making problems, Big Data Min. Anal. 4 (2) (2021) 94–103.
[7] C. Esposito, O. Tamburis, X. Su, C. Choi, Robust decentralised trust management for the internet of things by using game theory, Inf. Process. Manag. 57 (6) (2020) 102308.
[8] V. Hassija, V. Chamola, G. Han, J.J. Rodrigues, M. Guizani, Dagiov: a framework for vehicle to vehicle communication using directed acyclic graph and game theory, IEEE Trans. Veh. Technol. 69 (4) (2020) 4182–4191.
[9] J. He, Y. Li, H. Li, H. Tong, Z. Yuan, X. Yang, W. Huang, Application of game theory in integrated energy system systems: a review, IEEE Access 8 (2020) 93380–93397.
[10] S. Abapour, M. Nazari-Heris, B. Mohammadi-Ivatloo, M.T. Hagh, Game theory approaches for the solution of power system problems: a comprehensive review, Arch. Comput. Methods Eng. 27 (1) (2020) 81–103.
[11] X. Zhang, H. Ma, J. Wang, S. Zhou, H. Liu, Game theory design for deceptive jamming suppression in polarization mimo radar, IEEE Access 7 (2019) 114191–114202.
[12] C. Adami, J. Schossau, A. Hintze, The reasonable effectiveness of agent-based simulations in evolutionary game theory, Phys. Life Rev..doi:10.1016/j.plrev.2016.11.005.
[13] D.Y. Lin, C.J. Juan, M. Ng, Evaluation of green strategies in maritime liner shipping using evolutionary game theory, J. Clean. Prod. 279 (2021) 123268.
[14] Y. Liu, F. Wang, Y. Yang, X. Zhang, H. Wang, H. Dai, L. Qi, An attention-based category-aware gru model for next poi recommendation, Int. J. Intell. Syst..doi:10.1002/int.22412.
[15] X. Chen, Z. Yuan, Z. Cui, D. Zhang, X. Ju, Empirical studies on the impact of filter-based ranking feature selection on security vulnerability prediction, IET Softw. 15 (1) (2021) 75–89.
[16] F. Wang, H. Zhu, G. Srivastava, S. Li, M. R. Khosravi, L. Qi, Robust collaborative filtering recommendation with user-item-trust records, IEEE Trans. Comput. Soc. Syst..doi:10.1109/TCSS.2021.3064213.
[17] R. Kumari, S. Kumar, R.C. Poonia, V. Singh, L. Raja, V. Bhatnagar, P. Agarwal, Analysis and predictions of spread, recovery, and death caused by covid-19 in India, Big Data Min. Anal. 4 (2) (2021) 65–75.
[18] Z. Cai, Z. He, X. Guan, Y. Li, Collective data-sanitization for preventing sensitive information inference attacks in social networks, IEEE Trans. Dependable Secure Comput. 15 (4) (2016) 577–590.
[19] X. Chen, Y. Zhao, Z. Cui, G. Meng, Y. Liu, Z. Wang, Large-scale empirical studies on effort-aware security vulnerability prediction methods, IEEE Trans. Reliab. 69 (1) (2019) 70–87.
[20] H. Kou, H. Liu, Y. Duan, W. Gong, Y. Xu, X. Xu, L. Qi, Building trust/distrust relationships on signed social service network through privacy-aware link prediction process, Appl. Soft Comput. 100 (2020) 106942.
[21] Z. Sun, Y. Wang, Z. Cai, T. Liu, X. Tong, N. Jiang, A two-stage privacy protection mechanism based on blockchain in mobile crowdsourcing, Int. J. Intell. Syst..doi:10.1002/int.22371.
[22] Y. Khazbak, J. Fan, S. Zhu, G. Cao, Preserving personalized location privacy in ride-hailing service, Tsinghua Sci. Technol. 25 (6) (2020) 743–757.
[23] J. Bootkrajang, A generalised label noise model for classification in the presence of annotation errors, Neurocomputing 192 (2016) 61–71.
[24] A. Cappozzo, F. Greselin, T.B. Murphy, A robust approach to model-based classification based on trimming and constraints, Adv. Data Anal. Classif. (2019) 1–28.
[25] T. Liu, D. Tao, Classification with noisy labels by importance reweighting, IEEE Trans. Pattern Anal. Mach. Intell. 38 (3) (2015) 447–461.
[26] Z. Zhao, L. Chu, D. Tao, J. Pei, Classification with label noise: a Markov chain sampling framework, Data Min. Knowl. Discov. 33 (5) (2019) 1468–1504.
[27] B. Chen, S. Xia, Z. Chen, B. Wang, G. Wang, Rsmote: a self-adaptive robust smote for imbalanced problems with label noise, Inf. Sci. 553 (2021) 397–428.
[28] N. Huang, Q. Chen, G. Cai, D. Xu, L. Zhang, W. Zhao, Fault diagnosis of bearing in wind turbine gearbox under actual operating conditions driven by limited data with noise labels, IEEE Trans. Instrum. Meas. 70 (2020) 1–10.
[29] K.H. Lee, X. He, L. Zhang, L. Yang, Cleannet: transfer learning for scalable image classifier training with label noise, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5447–5456.
[30] Z. Zhang, W. Guo, M. Li, W. Yu, Gis-supervised building extraction with label noise-adaptive fully convolutional neural network, Geosci. Rem. Sens. Lett. IEEE 17 (12) (2020) 2135–2139.
[31] Z. Cai, X. Zheng, A private and efficient mechanism for data uploading in smart cyber-physical systems, IEEE Trans. Netw. Sci. Eng. 7 (2) (2018) 766–775.
[32] L. Qi, C. Hu, X. Zhang, M.R. Khosravi, S. Sharma, S. Pang, T. Wang, Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment, IEEE Trans. Ind. Inf. 17 (6) (2021) 4159–4167.
[33] Q. Zhang, F. Lee, Y.g. Wang, R. Miao, L. Chen, Q. Chen, An improved noise loss correction algorithm for learning from noisy labels, J. Vis. Commun. Image Represent. 72 (2020) 102930.

[34] D. Guan, K. Chen, G. Han, S. Huang, W. Yuan, M. Guizani, L. Shu, A novel class noise detection method for high-dimensional data in industrial informatics, IEEE Trans. Ind. Inf. 17 (3) (2020) 2181–2190.
[35] I. Vranckx, J. Raymaekers, B. DeKetelaere, P.J. Rousseeuw, M. Hubert, Real-time discriminant analysis in the presence of label and measurement noise, Chemometr. Intell. Lab. Syst. 208 (2021) 104197.
[36] S. Xia, B. Chen, G. Wang, Y. Zheng, X. Gao, E. Giem, Z. Chen, Mcrf and mrd: two classification methods based on a novel multiclass label noise filtering learning framework, IEEE Transact. Neural Networks Learn. Syst..doi:10.1109/TNNLS.2020.3047046.
[37] W. Feng, Y. Quan, G. Dauphin, Label noise cleaning with an adaptive ensemble method based on noise detection metric, Sensors 20 (23) (2020) 6718.
[38] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, L. Qu, Making deep neural networks robust to label noise: a loss correction approach, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1944–1952.
[39] X. Chen, C. Chen, D. Zhang, Z. Xing, Sethesaurus: Wordnet in software engineering, IEEE Trans. Software Eng..doi:10.1109/TSE.2019.2940439.
[40] L. Qi, H. Song, X. Zhang, G. Srivastava, X. Xu, S. Yu, Compatibility-aware web apis recommendation for mashup creation via textual description mining, ACM Trans. Multimed Comput. Commun. Appl.doi:10.1145/3417293.
[41] L. Yao, Z. Pengzhou, Z. Chi, Research on news keyword extraction technology based on tf-idf and textrank, in: 2019 IEEE/ACIS 18th International Conference on Computer and Information Science, IEEE, 2019, pp. 452–455.
[42] S. Bergamaschi, L. Po, Comparing lda and lsa topic models for content-based movie recommendation systems, in: International Conference on Web Information Systems and Technologies, Springer, 2014, pp. 247–263.
[43] M. Maimaiti, Y. Liu, H. Luan, M. Sun, Enriching the transfer learning with pre-trained lexicon embedding for low-resource neural machine translation, Tsinghua Sci. Technol..doi:10.26599/TST.2020.9010029.
[44] J. Choi, S. Lee, Improving fasttext with inverse document frequency of subwords, Pattern Recogn. Lett. 133 (2020) 165–172.
[45] M. Khan, H. Wang, A. Riaz, A. Elfatyany, S. Karim, Bidirectional lstm-rnn-based hybrid deep learning frameworks for univariate time series classification, J. Supercomput. 77 (7) (2021) 7021–7045.
[46] P. Rodríguez, M.A. Bautista, J. Gonzalez, S. Escalera, Beyond one-hot encoding: lower dimensional target embedding, Image Vis Comput. 75 (2018) 21–31.
[47] K. Park, J.S. Hong, W. Kim, A methodology combining cosine similarity with classifier for text classification, Appl. Artif. Intell. 34 (5) (2020) 396–411.
[48] J. Mabrouki, M. Azrour, D. Dhiba, Y. Farhaoui, S.E. Hajjaji, Iot-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts, Big Data Min. Anal. 4 (1) (2021) 25–32.
[49] Y. Jin, W. Guo, Y. Zhang, A time-aware dynamic service quality prediction approach for services, Tsinghua Sci. Technol. 25 (2) (2020) 227–238.