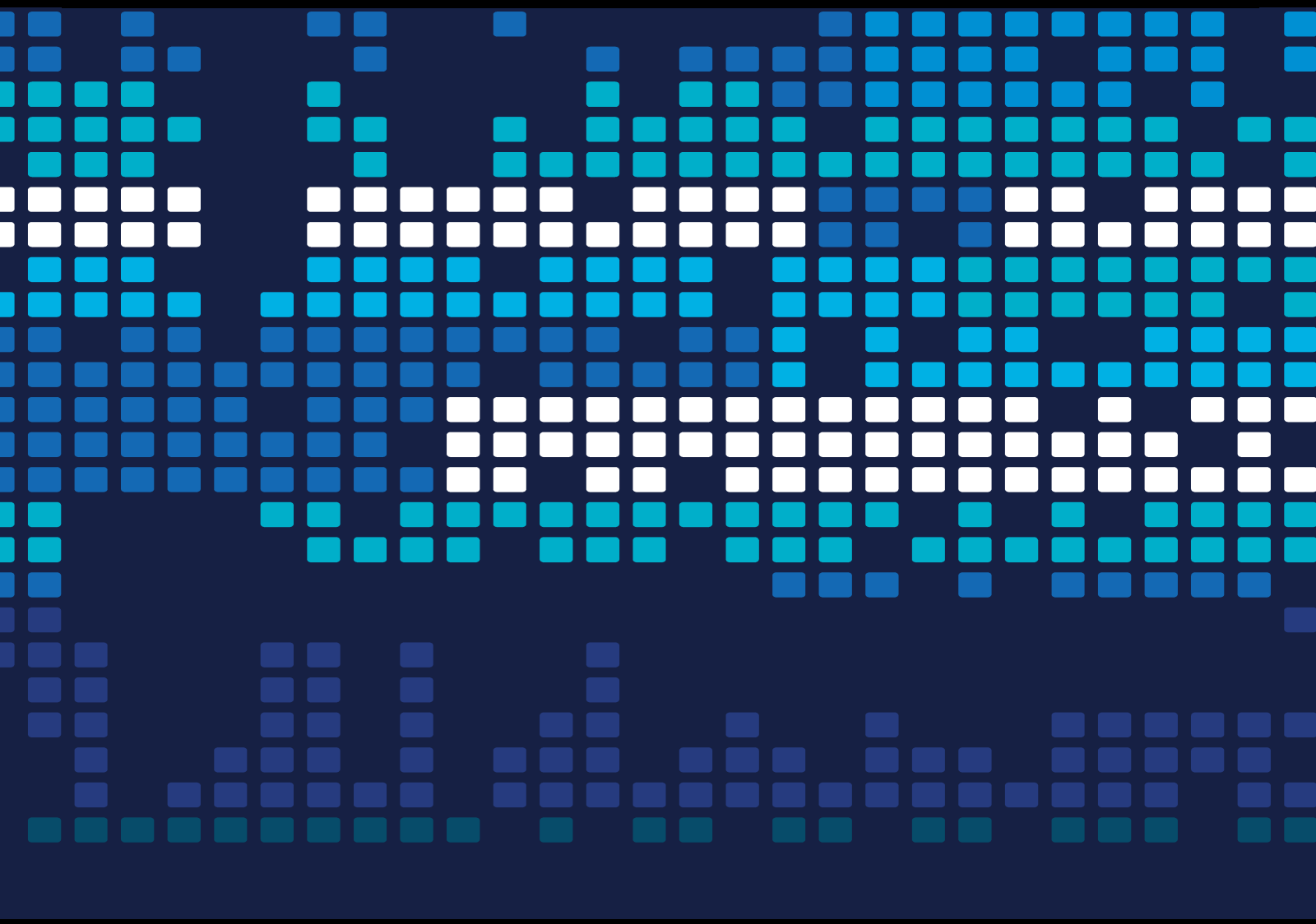
Scientific Programming

Intelligent Decision Support Systems Based on Machine Learning and Multicriteria Decision-Making

Lead Guest Editor: Rahman Ali
Guest Editors: Kifayat Ullah Khan, Asad M. Khatak, Muhammad Sajjad, and Muhammad Fahim



Intelligent Decision Support Systems Based on Machine Learning and Multicriteria Decision-Making

Scientific Programming

Intelligent Decision Support Systems Based on Machine Learning and Multicriteria Decision-Making

Lead Guest Editor: Rahman Ali
Guest Editors: Kifayat Ullah Khan, Asad M. Khatak, Muhammad Sajjad, and Muhammad Fahim

Contents

Hindawi
Scientific Programming
Volume 2020, Article ID 9353120, 17 pages
https://doi.org/10.1155/2020/9353120

Research Article

Predicting Politician's Supporters' Network on Twitter Using Social Network Analysis and Semantic Analysis

Asif Khan,[1] **Huaping Zhang**,[1] **Jianyun Shang**,[1] **Nada Boudjellal**,[1] **Arshad Ahmad**,[2] **Asmat Ali**,[1,3] **and Lin Dai**[1]

[1]*School of Computer Science and Technology, Beijing Institute of Technology, Beijing, 100081, China*
[2]*Department of Computer Science, University of Swabi, Anbar 25000, Swabi, Pakistan*
[3]*Department of Computer Science, University of Peshawar, Peshawar 25000, Pakistan*

Correspondence should be addressed to Huaping Zhang; kevinzhang@bit.edu.cn

Received 16 December 2019; Accepted 1 August 2020; Published 1 September 2020

Academic Editor: Muhammad Sajjad

Politics is one of the hottest and most commonly mentioned and viewed topics on social media networks nowadays. Microblogging platforms like Twitter and Weibo are widely used by many politicians who have a huge number of followers and supporters on those platforms. It is essential to study the supporters' network of political leaders because it can help in decision making when predicting their political futures. This study focuses on the supporters' network of three famous political leaders of Pakistan, namely, Imran Khan (IK), Maryam Nawaz Sharif (MNS), and Bilawal Bhutto Zardari (BBZ). This is done using social network analysis and semantic analysis. The proposed method (1) detects and removes fake supporter(s), (2) mines communities in the politicians' social network(s), (3) investigates the supporters' reply network for conversations between supporters about each leader, and, finally, (4) analyses the retweet network for information diffusion of each political leader. Furthermore, sentiment analysis of the supporters of politicians is done using machine learning techniques, which ultimately predicted and revealed the strongest supporter network(s) among the three political leaders. Analysis of this data reveals that as of October 2017 (1) IK was the most renowned of the three politicians and had the strongest supporter's community while using Twitter in a very controlled manner, (2) BBZ had the weakest supporters' network on Twitter, and (3) the supporters of the political leaders in Pakistan are flexible on Twitter, communicating with each other, and that any group of supporters has a low level of isolation.

1. Introduction

Social media technologies have been utilized effectively in various domains, for instance, software engineering [1–5]. Likewise, social media networks are preemptive platforms for politicians, political parties, political institutions, and foundations to connect to citizens. Hundreds of thousands of political parties and politicians are using social media nowadays. They can get the attention of people easily and rapidly when compared to other traditional ways of communication. Every politician and political party has millions of followers on these social media networks, and the politicians try interesting and advanced ways to encourage the citizens to engage in politics. Besides this, social media helps the politicians in various decision making processes by indicating recommendations, for instance, devising policies/strategies based on the past experiences, recommending and selecting suitable candidates for a particular constituency, recommending a suitable person for a particular position in the party, and launching a political campaign according to the sentiments of citizens on different issues and controversies among others [6–15].

Many researchers have analyzed Twitter in different ways: the importance of Twitter in journalism and political news coverage [6], the use and importance of Twitter in medical education [16, 17], prediction in e-commerce using sentiment analysis [18], sentiments of people towards terrorist events [19], privacy and security issues [20], use of Twitter by the government to engage citizens [21, 22], manipulation, and misinformation propagation during elections [23, 24], among numerous others. Analyzing politics on social media is a hot topic in the past few years.

Twitter is considered to be on top as it is commonly used among politicians. The style of Donald Trump's tweets and its change over time has been analyzed [25]. The work of Tromble in [26] has studied politician-citizen engagement. In political prediction, researchers mostly analyzed Twitter for political campaigns and elections predictions. Election prediction using sentiment analysis [27, 28], social network analysis [29], topic modeling and sentiment analysis [30], and aggregating online and offline data [31] were studied, but all these studies used election time data. To the best of our knowledge, no research had drawn attention to the supporters' network of political leaders in normal time (long before the election), which motivated us to investigate specifically the supporters' network of political leaders of Pakistan on Twitter in normal time that can lead to the prediction of a political leader's fate in the election.

Every politician has supporters and adversaries that can change their futures. These supporters can help a politician to be more influential and win an election. A question still lies there, if the followers of a politician are supporters or adversaries? This study focuses on the supporters' network of three famous political leaders of Pakistan, namely, Imran Khan (IK), Maryam Nawaz Sharif (MNS), and Bilawal Bhutto Zardari (BBZ). This study is mainly based on the following research questions:

RQ1: which political leader has the strongest supporter community based on the analysis of network structure and community detection techniques?

RQ2: which political leader has the strongest supporters' community based on analyzing the content of supporters on Twitter?

To address the aforementioned research questions, this study has utilized multicriteria decision making and machine learning algorithms by (1) studying the politician's supporter network using social network analysis techniques and (2) analyzing the politician's supporter networking using semantic analysis.

Initially, fake supporters were identified and removed. All the tweets, comments, and retweets of the identified fake supporters were discarded. The approach used in the first part predicts the strongest supporters' network by analyzing network structure and running community detection on the supporters of three political leaders on Twitter. Additionally, the supporters' reply network and retweet network of each political leader have been investigated in this study. The retweet behavior of supporters shows us the information diffusion among the supporters, and the replying behavior shows the communication between supporters of the leader(s).

The second approach predicts the strongest supporters' network by analyzing the sentiments of supporters of every political leader using supervised machine learning techniques such as Support Vector Machines (SVM), Naïve Bayesian (NB), and K-Nearest Neighbour (KNN). These sentiments are classified into Positive, Neutral, and Negative. Next, a word cloud is built for every political leader from the content used by supporters.

The main contributions of this study are

(1) identifying and removing the fake supporters in a supporter's network of political leaders
(2) identifying the existing supporters' communities of political leaders on Twitter
(3) determining the strongest supporter's network of the three political leaders using network analysis
(4) determining and categorizing the strongest supporter's network of the three political leaders using semantic analysis.

This study can particularly assist in understanding the supporters of the political leaders of Pakistan. In general, it helps political leaders to understand their supporters in a better way; they can make decisions based on the supporter communities' sentiments, information diffusion patterns, and communication patterns. Besides this, the political leaders can adapt their party agenda and slogans according to the sentiments and needs of the people, which will ultimately assist in gaining more support in their political campaigns. Nevertheless, this study can also help other countries in policymaking while enhancing bilateral relations.

The rest of this paper is organized as follows: Section 2 provides all the related work to this study; the research methodology is discussed in Section 3; the results and discussion are in Section 4; and, finally, Section 5 provides the conclusions and the possible future work of this study.

2. Related Work

Many researchers have analyzed different political leaders and parties on twitter. Verweij [32] analyzed the Twitter network of Dutch journalists and politicians using some basic network metrics like in-degree and out-degree and they concluded that the subgroups do not influence the contact between political leaders and journalists. In 2015, Dokoohaki et al. [29] implemented link analysis techniques to predict Swedish elections. The data of the political party was collected during election days. They presented evidence that link analysis can play an important role in prediction. The work of [33, 34] investigated the aspect of the Greek MPs Twitter network. They studied MPs (members of parliament) who belonged to the two major parties of Greek. They also studied the common followers and unique followers of MPs.

In [35], the authors studied the political alignment during parliamentary election 2015 in Venezuela using social network analysis and unsupervised machine learning methods. The data was taken a week before elections. Japan's 48th general election for the lower house has been analyzed in [36]; they investigated the ruling party and opposition party's retweet network and found that they are almost similar. The works of [37, 38] predicted the 2017 French elections using term weight and sentiment analysis. In 2016, Xie et al. [31] studied the three leaders of Taiwan during the election, and they predicted the popularity of leader based on the number of mentioning a leader, proportion of Amazon

web data of politicians, daily average Facebook likes, and the ratio of search indexes.

Furthermore, in 2018 in [39], they predicted Taiwan's election based on politician's popularity. They predicted popularity based on Facebook likes count, comment count, and event detection. They used the data during the election campaign.

This study focuses on the political leaders of Pakistan on Twitter. Some researchers have work on the politics of Pakistan on Twitter, but their works were limited to the prediction of elections using the data collected during election campaign time. The work of [27, 40] analyzed the 2013 General Election of Pakistan, and they have analyzed the four significant parties, Pakistan Tehreek-i-Insaf (PTI), Muttahida Qauomi Movement (MQM), and Pakistan People's Party using sentiment analysis. Furthermore, in the work of [41], they investigated political parties on Twitter and predicted the 2018 general election of Pakistan using sentiment analysis. They trained a deep neural network on 2013 elections Twitter data and tested on 2018 Twitter data. Aragón et al. [42] analyzed communication dynamics by investigating reply and retweet network of Spanish leaders during the 2011 election campaign, but their work did not study the supporters' reply and retweet networks.

This study is different from the work cited above. This study investigates the support of the three political leaders of Pakistan (MNS, IK, and BBZ). The supporters' reply and retweet networks of the three leaders have been studied, and the strongest support network of the political leader has been predicted using social network analysis and semantic analysis, sentiment analysis using machine learning techniques and word cloud.

3. Supporters' Network of Political Leaders

This section discusses the methodology used in this study. Due to the modular nature of the research questions of this study, the research methodology of each research question is explained separately in the following subsections.

3.1. Data Extraction and Finding Political Leaders' Demographics. A new application on Twitter [43] was created in order to obtain access tokens (consumer_key, consumer_secret, access_token, and access_secret) for the Twitter Search API. The tweets were collected using a python library "Tweepy" [44]. It is an open-source library for python, which facilitates access to the Twitter API [45]. It gives access to all objects as well as methods of the official Twitter API. Figure 1 shows the process of tweets collection. The data (Tweets) were in the form of JSON (JavaScript Object Notation), which is in a simple and lightweight data-interchange format. Humans can use (read and write) it quickly because it consists of key/value pairs, and it is easy for a machine to generate and parse it. The downloaded JSON file of tweets contained enormous amounts of data, such as user details, texts, retweets, replies, mentions, links, hashtags, and locations. The data was then parsed to the MySQL database while ignoring irrelevant information. Tweets that contained @ImraKhanPTI, #ImraKhanPTI @MaryamNSharif, #MaryamNSharif, @BBhuttoZardar, and #BBhuttoZardar were collected. The tweets were between 14th September 2017 to 13th October 2017. The total number of supporters was 159,683, and the number of Tweets was approximately 456,990.

Table 1 shows the Twitter profile details of each political leader of Pakistan, which is considered in this research.

Table 1 shows that IK has more followers on Twitter than the other three leaders. MNS uses Twitter more than the other two leaders. IK uses Twitter very precisely as he only has 6.2 K tweets and follows just 18 people.

3.2. Fake Supporters Detection. Fake supporters in the politician's supporters' network were detected and removed. Every day, hundreds and thousands of people are registering into the social media network. Apart from legitimate users, many fake people get register into those websites too. Those fake people always tend to be real, they often spam the real/legitimate users, and they post illegal and inappropriate content. A fake user/supporter in a supporter network of a political leader can misrepresent the ideology of a political party, or they can create some fake content to create hate for other communities (supporters of other politicians) by posting inappropriate contents about other politicians or groups There are some people (or bots) in a political network that have a special objective to manipulate the opinions of voters by promoting and spreading a specific agenda:

(1) Supporting and overcoming a weak or negative reputation of a political leader

(2) harming the reputation of a political leader (opponent)

In this study, a supporter is considered to be a fake supporter if he/she tweeted about a political leader but is not following him. The list of followers of political leaders was scrapped [46] and stored in MySQL; the supporters' list was matched with the followers' list of leaders, and unmatched supporters were considered fake and were removed along with their tweets. 82 users in the IK network, 207 in MNS network, and 25 in BBZ's network were identified and removed. Furthermore, fake supporters in the political network were detected and removed using (i) a traditional method, i.e., ratio of tweets, and (ii) using the ML technique. The first method uses supporters' metadata and basic activity feature: (a) friends count, (b) followers count, (c) statuses count, and (d) "verified," which is a binary field that shows whether the user account is verified by Twitter or not. In this research, a supporter is considered to be fake who has less than or equal to 10 followers and friends and more than 100 tweets. This method identified 18 fake supporters in IK's network and 11 in MNS's network.

The method using the ratio of tweets did not give the desired results. Similar to [24, 47–50], it is believed that there might be many bots or supporters in the network of the three political leaders that behave like a bot by copying, sharing, and retweeting content to distort and manipulate the real picture of a leader.

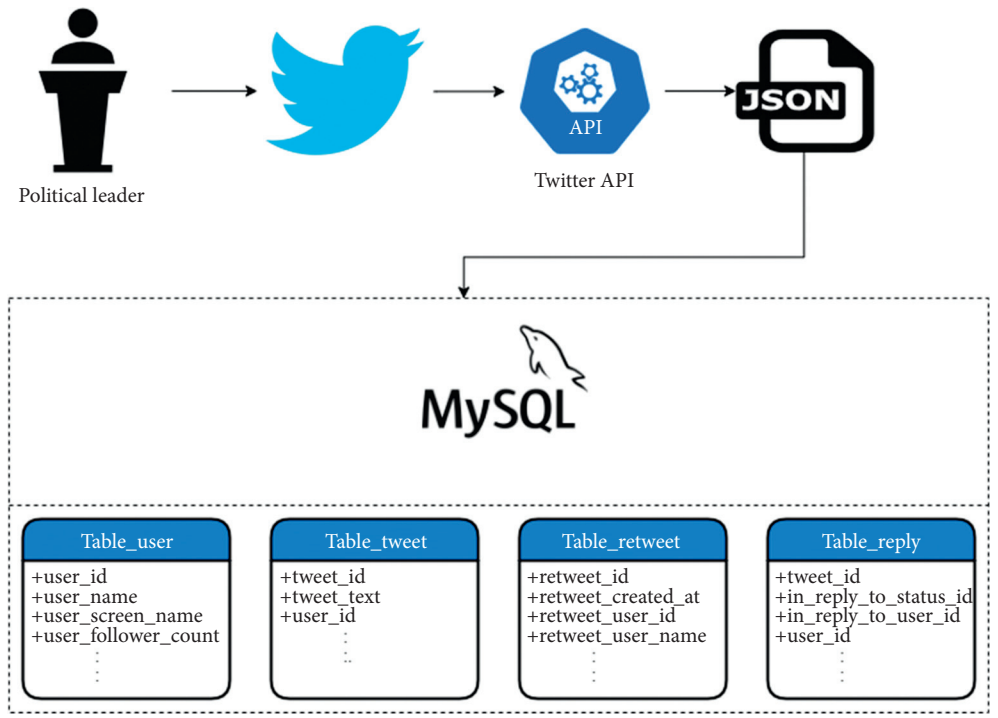


Figure 1: Data collection process.

Table 1: Twitter profile of the political leaders.

Name	Party name	Followers (millions)	Following (K)	User name	Tweets (since joining) (K)	Joined since
Imran Khan	Pakistan Tehreek-i-Insaf (PTI)	10.7	18	@ImranKhanPTI	6.2	March 2010
Maryam Nawaz Sharif	Pakistan Muslim League Noon (PMLN)	5.4	9.7	@MaryamNSharif	62	January 2012
Bilawal Bhutto Zardari	Pakistan People's Party (PPP)	3.6	1.9	@BBhuttoZardari	11.4	July 2011

The second method detects fake supporters using Botometer [51]. Botometer is publicly available and has an API. It detects fake users (bots) using machine learning techniques and has achieved a high accuracy (0.95). It classifies a user account into fake or real by using more than 1000 features of a user's metadata and information obtained by the content and network structure. Botometer grouped all those features, mainly in six classes: Network Features, Content Features, Sentiment Features, Friends Features, Temporal Features, and User Features. It gives output between 0 and 1. In this study, a supporter's account is considered to be a "fake" if the output score is 0.43 or higher. This technique identified about 36% of the total supporters of IK, 30.3% of MNS, and 20.7% in BBZ's to be fake. Those supporters and their tweets were removed. Finally, the data contains 54,058 supporters in IK's network, 36,942 in MNS's network, and 13,386 in BBZ's network.

3.3. Network Analysis of Supporters' Network. A community is a collection of individuals as a cluster to such an extent that there is high relatedness among individuals inside the community. In other words, a collection of people who tends to have common interests, goals, habits, likes, dislikes, and choices leads to making virtual communities/groups and clusters in a social network. Detecting such clusters and communities in social media is an essential task in many aspects and applications of life, such as science, sociology, psychology, computer science, and marketing. The association of individuals into groups gives critical experiences into aggregate elements that can be utilized for information dissemination, online web marketing, terrorism control, medical fields, politics, sports, culture, clubs, and many more.

The virtual groups inside online networking content are the direct impact of the homophile guideline, and the influence these groups have over their individuals is the effect of influence. Structure-based Social Network Analysis (SNA) denotes connections between individuals in the form of a graph, and it uses graph-based algorithms to mine communities or subgroups [52]. Such structures of graphs can be of incredible significance for understanding how data engenders and spreads in a network of individuals.

To address RQ1, the experiment has been set up into two parts: supporters' reply network and supporters' retweet network.

3.3.1. Reply Network Analysis of Supporters. To analyze communication patterns between supporters of a political leader and predict the strongest community, the authors studied the reply networks of each political leader. Supporters who replied to political leaders in their tweets are grouped into one network.

$$G^{\text{rep}} = \left(V^{\text{rep}}, A^{\text{rep}}\right), \tag{1}$$

(i) where V^{rep} denotes supporters, who reply to the political leaders in the network

(ii) A^{rep} is the set of directed links of reply between supporters and the leaders

Equation (1) creates a retweet network. Self-loop, i.e., $v_i^{\text{rep}} \longrightarrow v_j^{\text{rep}}$, is ignored, where v_i is a person in the network, who replies to his tweet and then analyzes the supporter's reply network of each leader and calculates the average degree, average weighted degree, network diameter, average path length, and nodes connectivity in the network.

Then, the Louvain method [53] is applied to mine communities in the reply network of each political leader, analyzing those communities, supporters existing in those communities, and analyzing whether the supporters directly took an interest in the leader by replying to him/her or through some other discussion by another group of people.

3.3.2. Retweet Network Analysis of Supporters. The supporters' retweet network of each political leader is studied. It contains those supporters who retweeted the leader or the leader retweeted someone else.

$$G^{\text{ret}} = \left(V^{\text{ret}}, A^{\text{ret}}\right), \tag{2}$$

(i) where V^{ret} denotes the supporters who contributed to the leaders' support by retweeting him/her in the network and

(ii) A^{ret} is the set of directed links of retweet links, i.e., V_i retweet V_j

Create a retweet network using (2). Self-loop here too, i.e., $v_i^{\text{ret}} \longrightarrow v_j^{\text{ret}}$, is ignored. Then, characteristics of the supporters' retweet network for a single leader were calculated, i.e., average degree, average weighted degree, network diameter, average path length, and nodes connectivity in the network.

Then, the authors used the Louvain method [53] to extract communities in a retweet network of each political leader and analyze those communities and study whether the supporters directly took an interest in the leader by retweeting him/her or by some other group of people or leader.

To validate the strongest supporter (reply and retweet) network using social network analysis techniques, this study combines the network of the three political leaders using

$$G = \left(G_{\text{MNS}}(V,E) + G_{\text{IK}}(V,E) + G_{\text{BBZ}}(V,E)\right), \tag{3}$$

(i) where $G_{\text{MNS}}(V,E)$ denotes supporters' (reply or retweet) network of MNS

(ii) $G_{\text{IK}}(V,E)$ denotes supporters' (reply or retweet) network of IK

(iii) $G_{\text{BBZ}}(V,E)$ denotes supporters' (reply or retweet) network of BBZ

(iv) V denotes supporters

(v) E denotes the links

Next, communities were extracted. The PageRank of each political leader in that network was calculated. The larger the size of the community of a leader and the higher the PageRank of a leader are, the more influential the leader is and the more support he/she has.

3.4. Semantic Analysis of Supporters' Social Network. RQ2 analyzes the tweets of supporters of every political leader by

(i) sentiment analysis and

(ii) semantic word cloud

Sentiment analysis, also known as opinion mining or emotional AI, refers to computational linguistics, Natural Language Processing, and text analysis to analytically extract, identify, and analyze affective states and subject information. People express their views and opinions on different topics over social network sites. Many government and private organizations need to know about the views, opinions, thoughts, feelings, behavior, and attitude of the public. This can help them to make new policies that can help them to get better output. Many people take an interest in politics, and they discuss politics. They try to take part in the discussion related to a political party or political leader. Citizens express their viewpoints about a leader or the campaign he/she is running. It is imperative to know what people think about a political leader.

As discussed earlier, fake supporters and their tweets were removed in this study. As a result, our data contained 213,080 tweets, 92,415 tweets from the supporters of IK, 86,721 tweets from the supporters of MNS, and 33,944 tweets from the supporters of BBZ. This study considered all those tweets written in the English language only. As a result, the final data contained 140,508 tweets in total, 73,846 tweets regarding IK, 45,197 tweets regarding MNS, and 21,465 regarding BBZ.

The extracted data from Twitter contains several syntactic features, which are not useful for this study. First of all, the tweets were preprocessed, as data filtering and cleaning techniques are required to remove irrelevant information. Many preprocessing steps are involved in this process: removing URLs, filtering, removing questions, removing special characters, removing stop-words, and removing emoticons.

Sentiment analysis can be achieved mainly using three approaches, lexicon based approach, machine learning based approach, and hybrid approach. This study used supervised machine learning techniques to predict the sentiments of supporters of political leaders. The machine learning based

Table 2: Supporters' reply network of political leaders.

Parameters	Imran Khan	Maryam Nawaz Sharif	Bilawal Bhutto Zardari
Nodes	21249	27933	5146
Edges	24753	31594	7294
Network diameter	5	17	13
Average path length	2.359	4.188	2.666
Number of weakly connected components	119	1977	217
Number of strongly connected components	2105	19138	6985

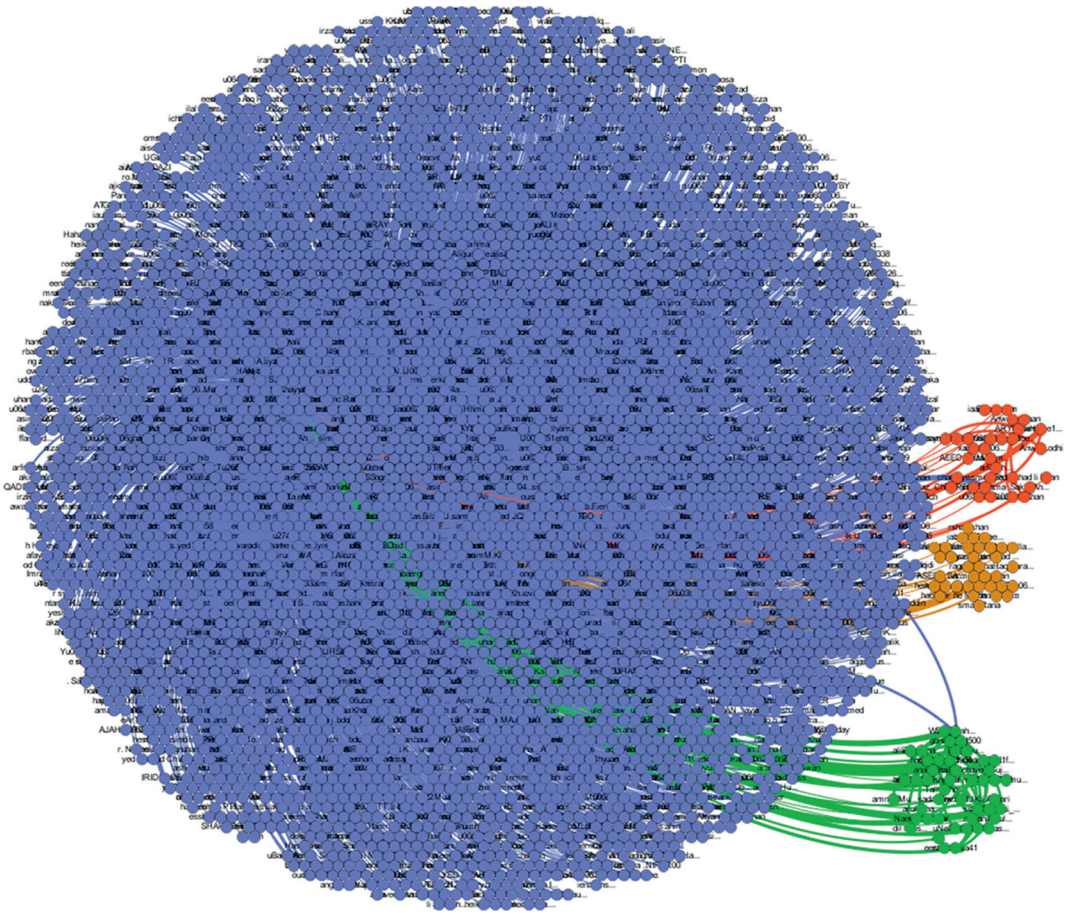


Figure 2: Communities in IK's supporter reply network. The blue colour community contains supporters of IK, green colour contains PTI's media group, orange colour contains BBZ, and red contains MNS.

approaches need labelled data to train and test the model. There are numerous labelled datasets available online; however, that did not fit the dataset used in this study. The authors manually labelled 21,000 tweets, which contained 7000 positive, 7000 negative, and 7000 neutral ones. In this study, SVM [54], NB [55], and KNN [56] have been considered. *K*-fold Cross-Validation was used to ensure that the dataset will not fall in overfitting during training and testing. $K = 10$ was taken in Cross-Validation.

At last, the best classifier was chosen for further use on unseen tweets. If the sentiments of people about a political leader are positive, the prediction about him/her will have a high probability of winning. By using sentimental analysis, a leader can know what his people think about him, what they want from him, and how they react to various things. Such information and analyses can help the politicians in decision making and refining their policies for the well-being of the public and their interest.

Furthermore, semantic word cloud of supporters' content has been generated. It maps the cooccurrence and frequency of different terms that appear in the tweets of supporters. Word clouds are used in the representation of text visually. It adds clarity during the analysis of the texts. It also divulges patterns in the data that can guide the future investigation. Word frequencies can be spotted easily. All the tweets were first preprocessed. This can be used by researchers, political leaders, and political parties to understand the sentiments, topics, and discussion used by supporters.

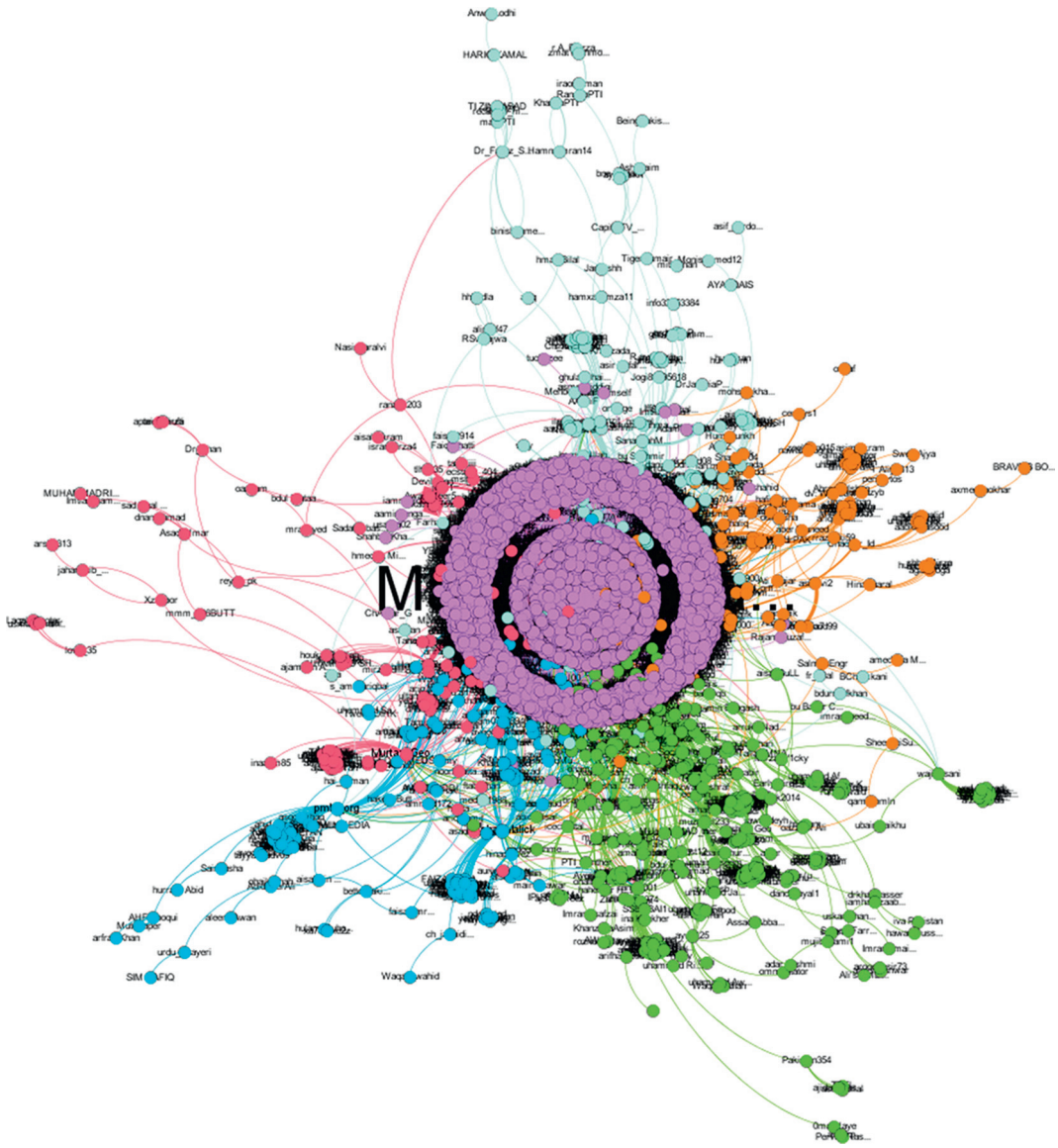


Figure 3: Communities in MNS's supporters' reply network. The lilac coloured community contains supporters of MNS, green colour contains IK, cerulean colour contains PMLN's leader, arctic colour contains BBZ, and orange colour contains PMLN's media cell.

4. Results and Discussion

To answer RQ1 and RQ2, the results of experiments are discussed mainly in two parts, social network analysis on supporters' network and semantic analysis of supporters' network.

4.1. Social Network Analysis on Supporters' Network. To answer RQ1, the following experiments are conducted to mine communities and predict the strongest support of a political leader.

4.1.1. Reply Network. Reply network of each political leader has been constructed, and Table 2 contains the properties of the supporters' reply network of each political leader.

The community detection method is applied to extract the communities for the supporter's reply network of IK. Figure 2 shows the three significant subcommunities in this network. This study shows a 60% view of final figures for clarity using Gephi [57]; the same method is followed for the rest of the experiments. The extracted communities in Figure 2 contain 86.26% of nodes and 87.37% of edges of the entire network. The other nodes in 13.74% consisting of insignificant communities containing few nodes were ignored in this analysis. The giant community that contains IK himself is 82.93%. The other three communities are having sizes 1.36%, 1.19%, and 0.79% of the total network. On manual study, it was found that the community with 1.36% relates to PTI's media group, orange to BBZ and red to MNS. Reply links between the community of IK and the other two are visible.

The communities were extracted in MNS supporter's reply network by implementing a community detection algorithm. Figure 3 shows six major communities, which consist of 78.33% of the whole network. The other 21.67% of the network consists of 438 insignificant communities. Those communities

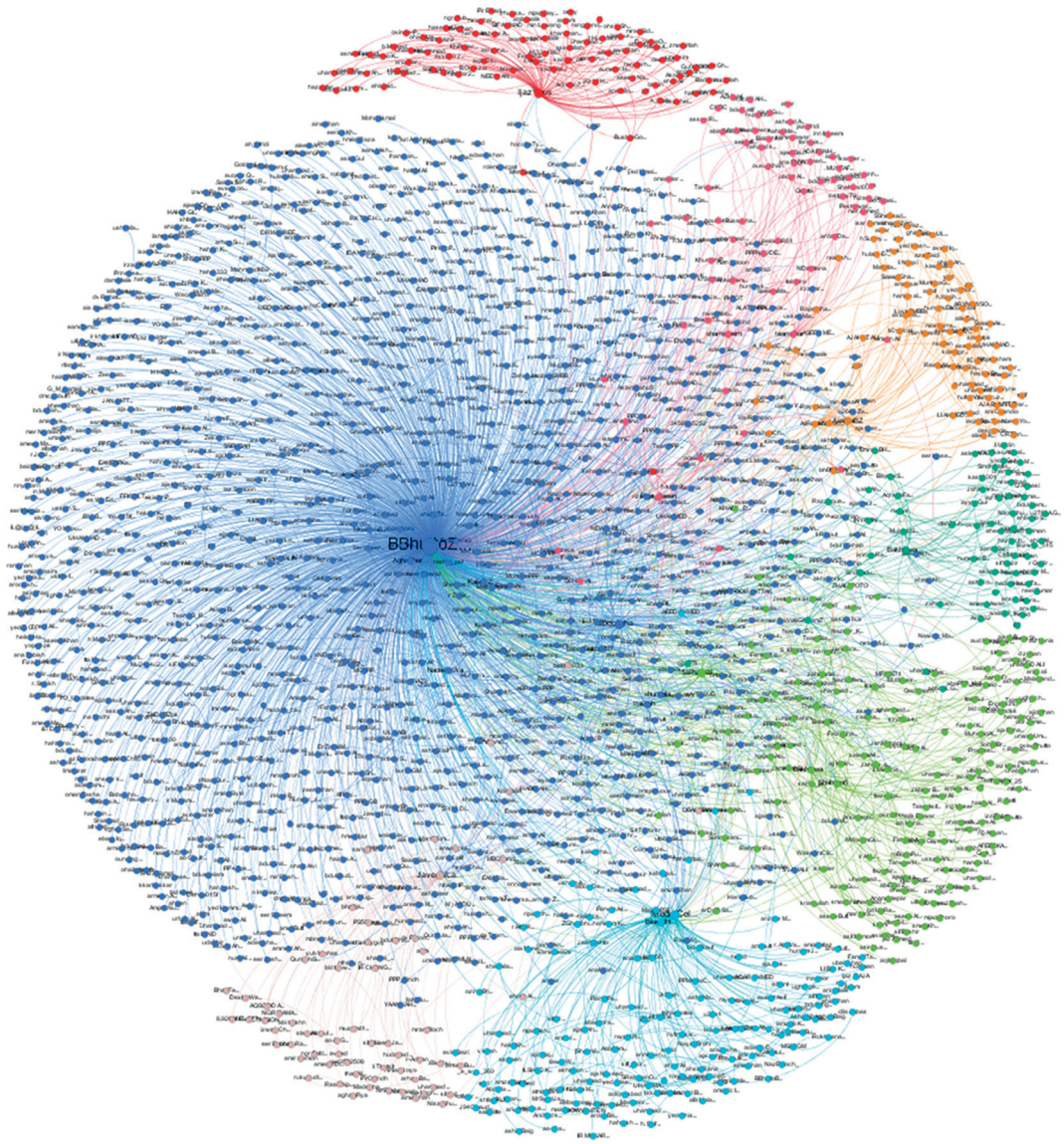


Figure 4: Communities in BBZ's supporters' reply network. The blue colour community contains supporters of BBZ, green colour contains PPP media cell, orange colour contains MNS, pink colour contains IK, and cerulean colour contains PPP's political leaders.

were ignored in this study as the sizes of those were between 0.001% and 0.94%. A giant community having lilac colour has a size of 52.14% and contains supporters lied in the MNS community and who replied directly to MNS. On manual study, it was noticed that the people in the community having a green colour weighs 4.76% of the total network related to IK. The community having cerulean colour consists of 2.87% containing another political leader who is a member of the Pakistan Muslim League Noon (party of MNS). The other community of arctic colour whose size is 2.2% of the whole network contained some supporters from BBZ who replied to MNS tweets. The remaining two communities, weighing 3.8%, each contained some people, in-touched with the media cell accounts of PMLN (Pakistan Muslim League Noon).

Figure 4 shows eight major extracted communities consisting of 80.18% nodes and 85.7% of links of the total supporters' reply network of BBZ. The giant community having the blue colour has a size of 50.08% consisting of supporters who lied in the BBZ community and replied to BBZ tweets. On a manual study, it was found that the green community which has the size of 7.47% contained the media cell of PPP (Pakistan People Party) and the cerulean colour community having sizes 6.72%, 3.82% 3.53%, 3.02%, 2.90%, and 2.64% contained other political leaders who are members of BBZ's party. Orange community having a 3.82% size contains some people from MNS's party and the pink community having a size of 3.02% contains people from IK's party.

Table 2 shows that the number of nodes is high in MNS supporters' network, but Figures 3 and 5 show that she has less support. The number of weakly connected components is less in IK, followed by BBZ and MNS. The smaller the network diameter, the stronger the network. IK has a network diameter of 5 and then BBZ and MNS, respectively.

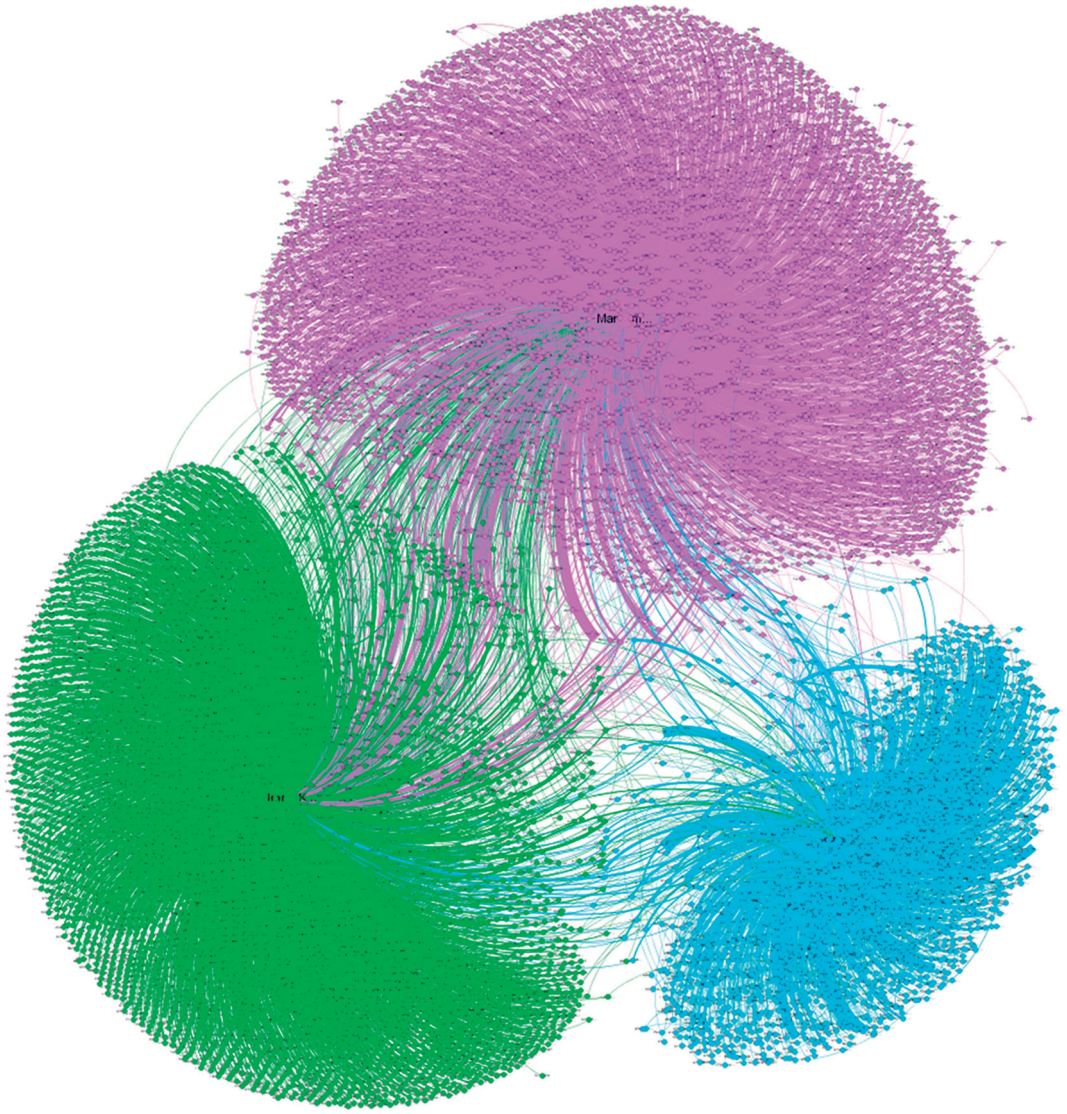


Figure 5: Supporters reply communities of IK, MNS, and BBZ. The green colour community contains supporters of IK, violet colour contains MNS, and blue colour contains BBZ.

Table 3: Supporters' retweet network of political leaders.

Parameters	Imran Khan	Maryam Nawaz Sharif	Bilawal Bhutto Zardari
Nodes	28128	10800	6180
Edges	62586	37215	14806
Network diameter	8	10	10
Average path length	4.846	5.633	3.749
Number of weakly connected components	135	164	81
Number of strongly connected components	27750	10564	6084

The average path length of the IK supporter network is less followed by BBZ and then MNS. All these characteristics lead us to the fact that IK has the strongest supporter network.

The supporters' reply network of the three leaders has been validated. The PageRank of IK is 0.177, MNS is 0.170, and BBZ is 0.051. Figure 5 shows that the three major extracted communities consist of 81.79% of the entire network. The other communities belong to some other leaders and parties. Supporters' reply community having green colour belongs to IK having size 34.70%, MNS 34.52%, and BBZ 12.57%. It concludes that IK is more influential and has strong support than MNS and BBZ, respectively. Figure 5 also shows the communication links between the supporters of the three leaders. It can be seen that the supporters of IK and BBZ are more flexible, and they have

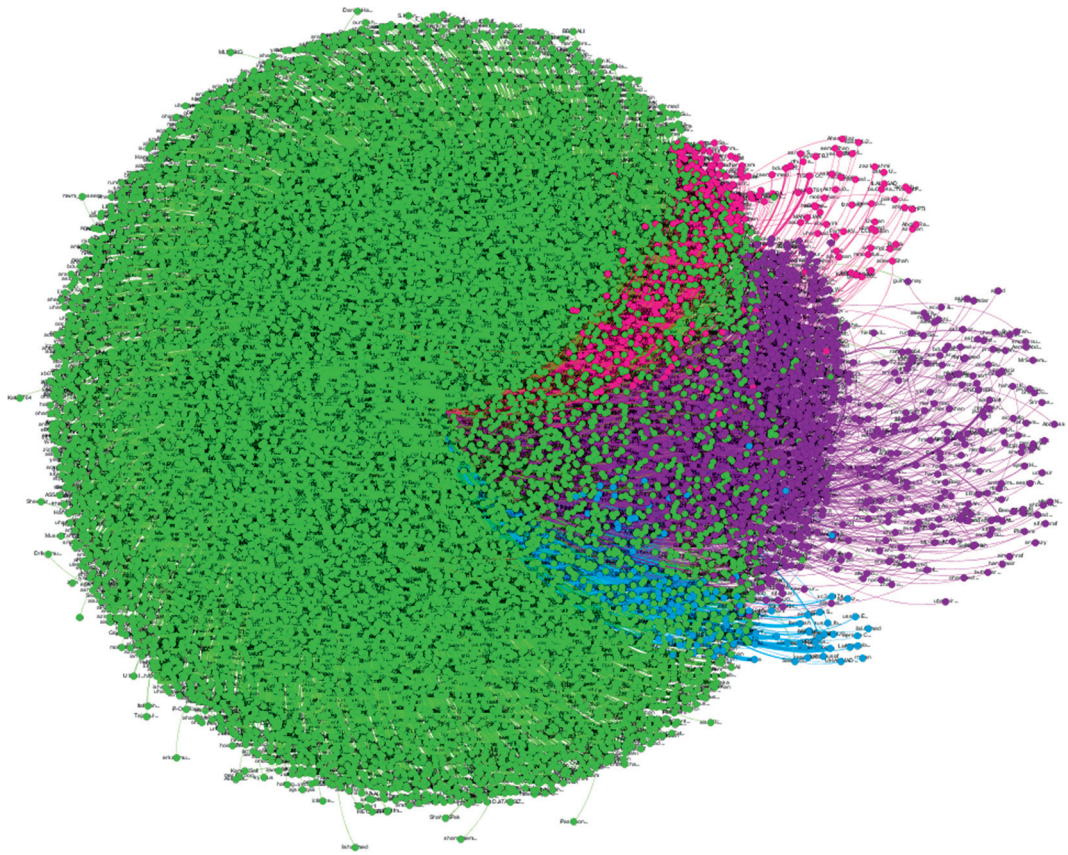


Figure 6: Communities in IK's supporters' retweet network. The green colour contains supporters of IK, violet colour contains PTI's account, and the rest are related to other leaders from PTI and other political parties.

more connections with the other communities. The supporters of MNS have more communication links with IK's supporters but few links with BBZ's supporters.

4.1.2. Retweet Network. The supporters' retweet networks of each political leader which shows information diffusion have been constructed. The properties of the supporters' retweet network of each leader are given in Table 3.

Figure 6 shows five major extracted communities in the supporters' retweet network of IK consisting of 94.90% of the entire network. The giant green community had the size of 80.16% IK and supporters who retweeted IK directly. The violet community is having the size of 10.43% consisting of the official page of PTI (Pakistan Tehreek-i-Insaf); the other communities with sizes 2. 95%, 1.23%, and 1.13% consist of other leaders of PTI and some other political parties.

Figure 7 shows five significant communities in the retweet network of MNS consisting of 89.35% of the full network. 64.46% consists of MNS and supporters who retweeted her directly. The green community with the size of 11.02% consists of people that belong to the media cells of PMLN (Pakistan Muslim League Noon). The other communities had the sizes of 6.5%, 6.38%, and 0.99%, and the rest belong to other members of PMLN and other parties.

Figure 8 shows eight major communities in the supporters' retweet network consisting of 71.31% of the entire network. The huge lilac community, which is only 24.90% of the whole network, consists of BBZ and supporters who directly retweeted him. The second-largest green community having the size 12.14% belongs to the media cell of PPP (Pakistan People's Party). The other communities belong to other parties and some prominent members of Bilawal's party who were supporting him by retweeting him and they have their followers.

Table 3 shows that the information diffusion rate of IK is higher as the number of supporters who retweeted IK is higher than in MNS and BBZ. The number of weakly connected components is less in MNS followed by IK and then BBZ, but IK has almost three times more nodes than MNS. IK has the smallest network diameter than that of the MNS and BBZ. The average path length of BBZ is 3.909, followed by IK and then MNS. On average, it can be concluded that IK has more influence and has strong support as compared to the other three political leaders.

The strongest retweet political network has been validated. PageRank of IK is 0.163, that of MNS is 0.044, and that of BBZ is 0.011. Figure 9 shows the three major extracted communities. The giant community having green colour is comprised of IK supporters, the purple community of MNS, and the blue community of BBZ. Communication links can be seen between the supporters of the three political leaders. The PageRank and the size of IK are higher than those of the other two leaders, which leads us to the strong support of

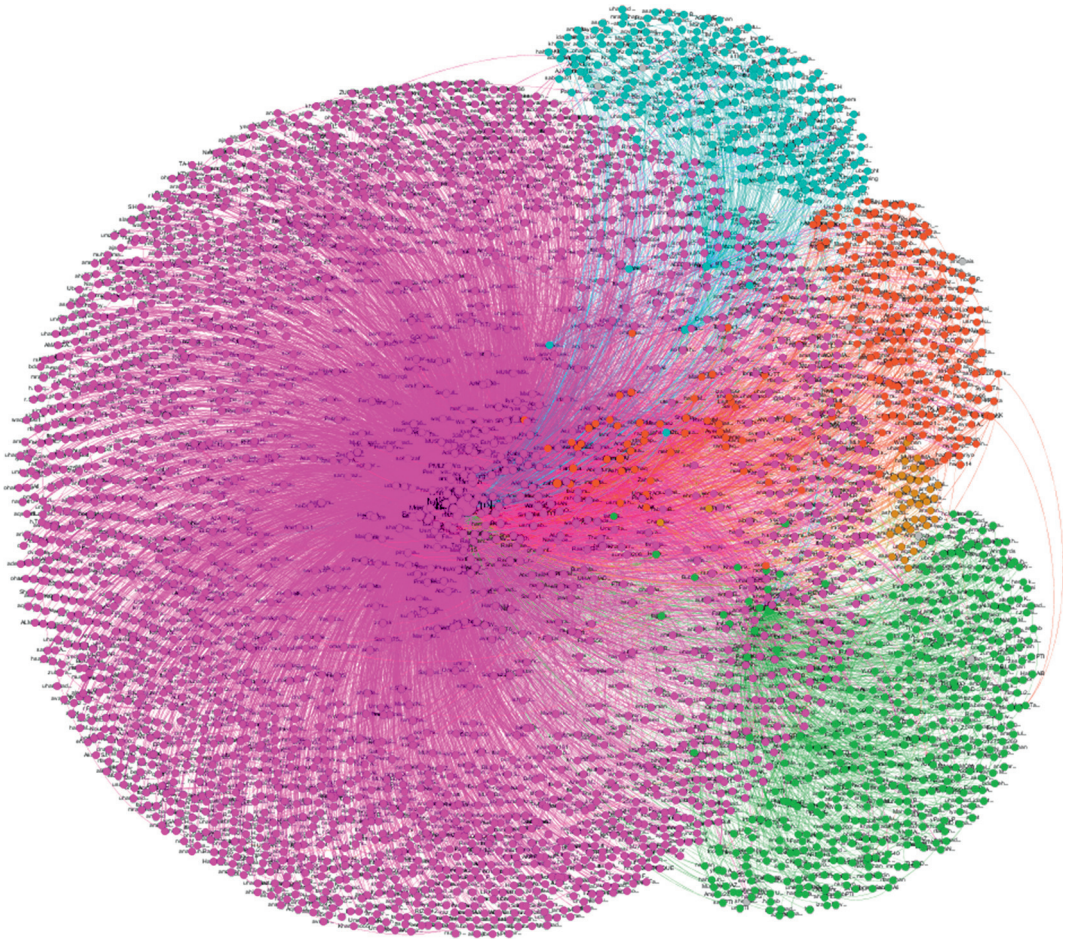


Figure 7: Communities in MNS's supporters' retweet network. The violet colour consists of MNS′ supporters; green colour consists of PMLN media cell; the rest are related to other leaders from PMLN and other parties.

people to IK. The information diffusion in the network of IK is higher.

From the above experiments and discussion, the answer of RQ1 can be concluded that IK has strong support from his supporters as compared to MNS and BBZ, respectively.

4.2. Semantic Analysis of Supporters' Network. To address RQ2, experiments were conducted to extract sentiments of supporters towards the three political leaders using machine learning techniques and generating the word cloud from the content used by supporters.

4.2.1. Sentiment Analysis of Supporters. NB, K-NN, and SVM were trained. Accuracy (in percentage) of the NB classifier, SVM classifier, and KNN classifier is plotted in Figure 10. It was found that SVM performed better than the other two classifiers with an accuracy of 78.89%; NB has 75.40%; and K-NN has 65.35%. The classifier was considered to classify unseen and unlabeled tweets in this research. Unlabeled tweets about the three political leaders were classified into three classes, i.e., Positive, Neutral, and Negative.

Figure 11 shows the percentage of polarity of supporters' tweets about IK, MNS, and BBZ. The number of supporters' tweets for every political leader is different; it is because a specific time slice has been considered in this study. IK's followers are more than those of the other two politicians. The percentage of sentiment polarity was considered to have better and balanced results. It can be seen that the people having a positive sentiment about IK have a high rate as compared to the BBZ and Maryam Nawaz. People's sentiment about BBZ has a high negative rate as compared to the other two politicians. IK and MNS are popular, and people talk more about them on Twitter as compared to BBZ. It is stated that a political leader, whose supporters have a high percentage of positive sentiments, has a strong supporter community. From these results, it can be concluded that IK has a stronger supporter community followed by MNS and BBZ, respectively.

4.2.2. Semantic Word Cloud of Supporters. In this analysis, the frequency of words used by supporters of the three political leaders is visualized using semantic word clouds. Table 4 shows some abbreviations used by the supporters in their tweets.

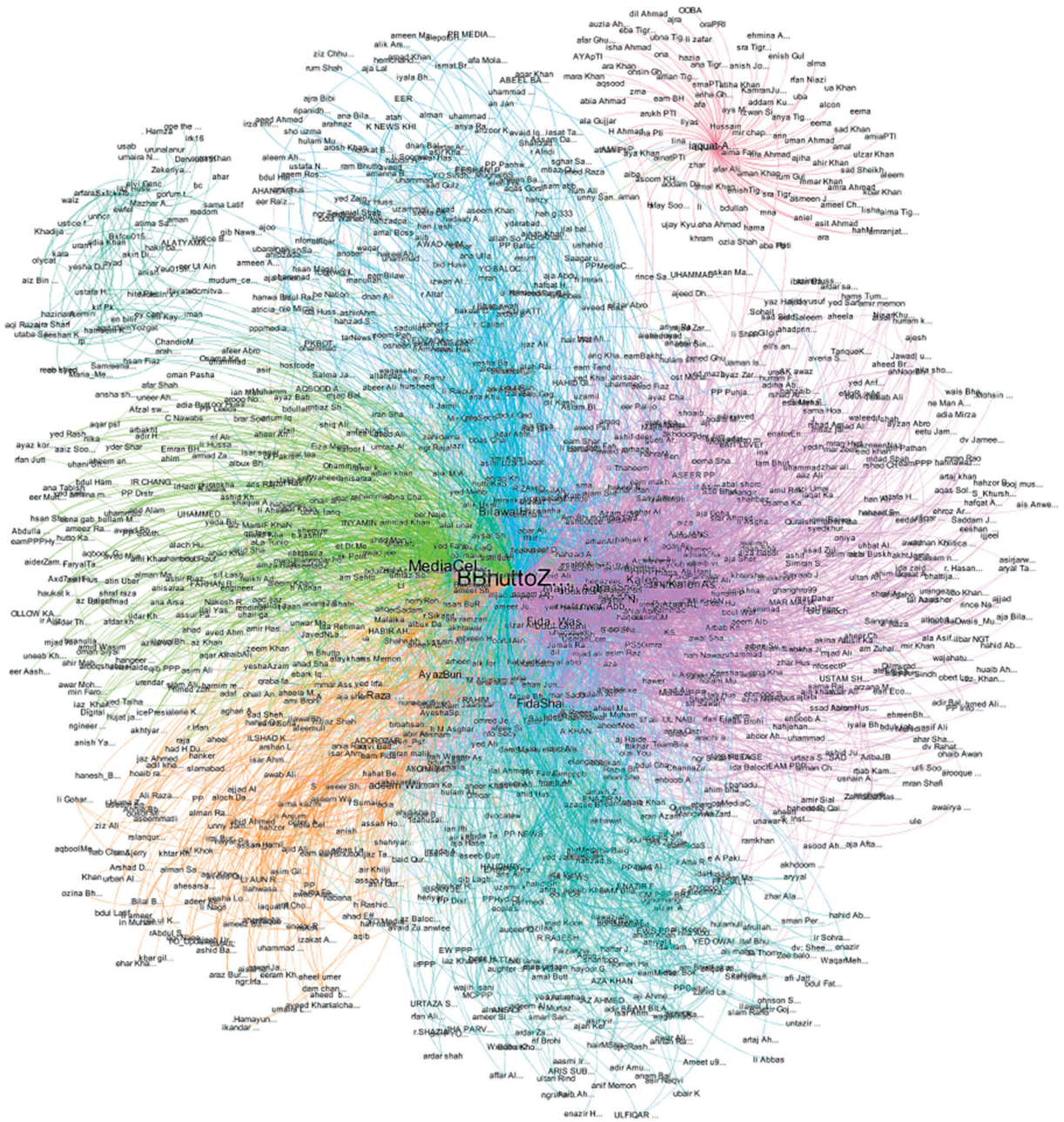


Figure 8: Communities in BBZ's supporter retweet network. The lilac colour stands for BBZ, green colour stands for PPP media group, and other communities belong to other parties and some prominent members of PPP.

Figure 12(a) shows word cloud for tweets of IK. It can be seen that the frequency of some words tends to have a positive sentiment like Great, Leader (great leader), Good, Allah, Leader, against (against corruption), More, Want, Support, Well, Right, Need, Save, and Nation. The frequency of these words is high, and it can be seen in the word cloud of IK. Figure 12(a) also shows some words which give negative sentiments, i.e., Corrupt, Shame, etc. It can be seen in the word cloud of IK that there are some words like PMLN, Go, NS, Nawaz, and Sharif. This is because IK initiated a slogan "Go Nawaz Go" which was a movement against the corruption of Nawaz Sharif (Ex-Prime Minister of Pakistan and father of MNS) and many supporters of IK raised this slogan. In the manual study, it was found that the word "corruption" in the word cloud of IK was actually against Nawaz Sharif. As can be seen that "Go, Nawaz, NS, and Corrupt" words have almost the same frequency.

Figure 12(b) shows word cloud for MNS. It can be seen that MNS's word cloud has many words having almost the same frequency as compared to IK's word cloud where some words have a very high frequency. It also shows words having positive sentiment like, Great, Allah, Love, Right, Support, People, Sher (lion), Good, God, Health, work, Ameen, Help, Nation, Pray, Happy, Better, Nice, Wish, Need, soon (better health soon), Insha'Allah, Life, Mother, and Save. During the time of the data collection for this

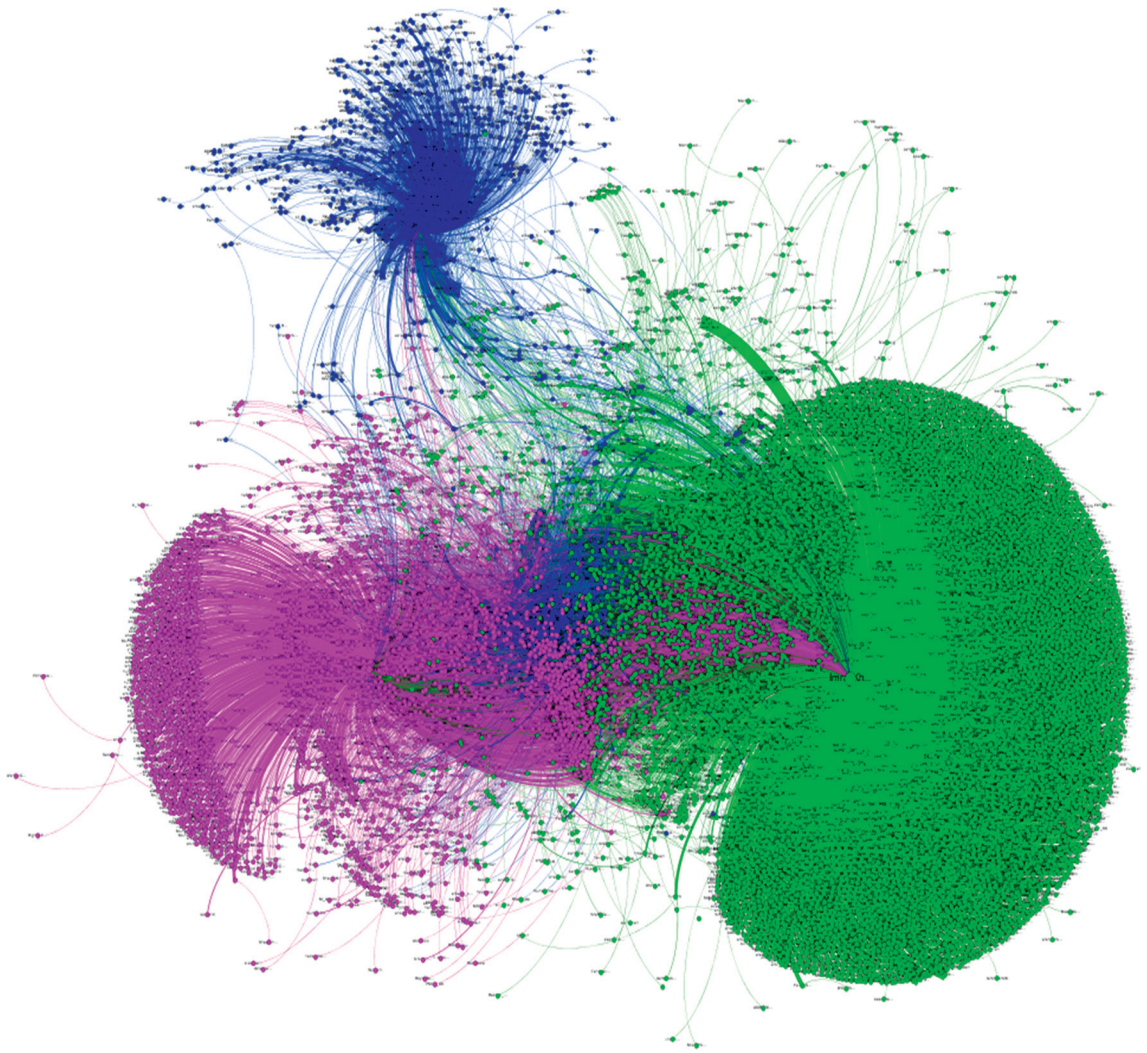


Figure 9: Communities of the supporters of IK, MNS, and BBZ. The green colour community contains supporters of IK, violet colour contains MNS, and blue colour contains BBZ.

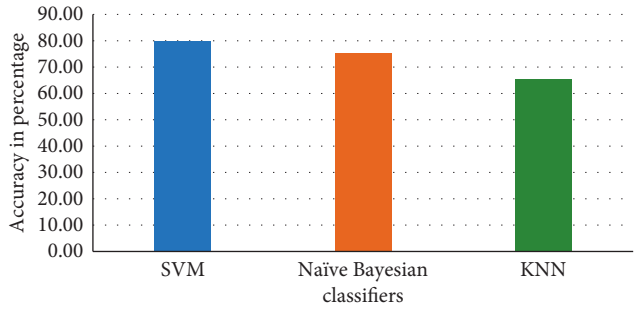


Figure 10: Classifiers accuracy.

research, MNS's mother was at the hospital and was fighting against cancer. Most of the supporters were praying for health. It can be seen that supporters of MNS have a huge love for her; they pray for her and her family. The supporters call her a great leader too. Figure 12(b) also shows us words having negative sentiments like, Chor, Loot, Corrupt, Shame, case (the case against her corruption scandal in the courts), Jail, Go, Stop, NAB, London (London flats—scandal), Liar, Never, Court, Fool, Hell, and False. Nawaz Sharif and his children (including MNS) were part of a corruption scandal like flats in London, for which they were facing some cases in NAB and courts by that time, and later they were

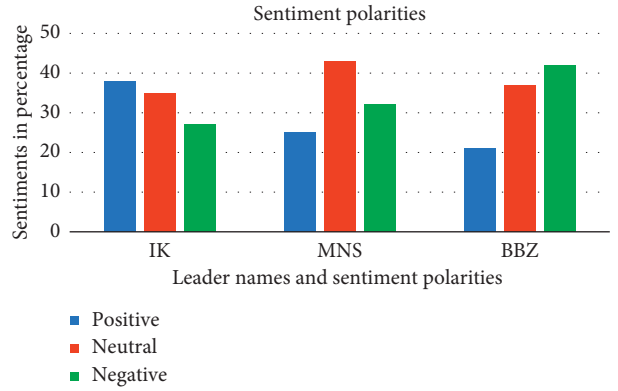


Figure 11: Sentiments of supporters of the IK, MNS, and BBZ.

Table 4: Abbreviation/acronym used by supporters of political leaders.

Abbreviation/acronym	Description
IK	Imran Khan
MN/MNS	Maryam Nawaz Sharif
NAB	National Accountability Bureau
PK	Pakistan
SC	Supreme court
BB/BBZ	Bilawal Bhutto Zardari
NA	National Assembly
KP	Khyber Pakhtunkhwa
NS	Nawaz Sharif
CM	Chief Minister
MPA	Minister Provincial Assembly
MNA	Minister National Assembly
PTI	Pakistan Tehreek-i-Insaf
PML	Pakistan Muslim League
PMLN	Pakistan Muslim League Noon
PPP/PP	Pakistan People's Party
ECP	Election Commission of Pakistan

(a) (b) (c)

Figure 12: Word cloud: (a) IK, (b) MNS, and (c) BBZ.

sentenced to jail before elections. Many people are calling them Chor (thief), liar, corrupt, etc. It can be seen in the word cloud of MNS that some people used inappropriate language, which shows the extremism of the supporters.

Figure 12(c) shows a word cloud for BBZ's. It consists of some personalities, places, etc., with high frequency, i.e., BiBi or BB (Benazir Bhutto), Benazir, Murad, CM (Chief Minister), Chairman (BBZ), Shah, Latif, Bhittai, Asif, Zardari,

Karachi, Hyderabad, Mazar, etc. This gives a viewpoint that there are some people (leaders) from the party who strongly support BBZ and want to make him superior. It can be seen in Figure 12(c) that there are other renowned personalities who are supporting BBZ. Words like, Jeay (long live), Next (next Prime Minister), Peace, Young, Work, Wish, Thank, Need, Pak, Educated Nice, Best, Peace, More, against (Against corruption), Public, and Right show positive sentiments. Words like Shame, Corrupt, Kid, Terror, Sad, and Out show a negative sentiment of people. The result discussed previously answers RQ2 that IK has the strongest supporter network on the basis of content analysis.

5. Conclusion and Future Work

This study was conducted to assist in decision making and predicting the future of a political leader based on multicriteria and machine learning algorithms. The supporters' network of three political leaders of Pakistan on Twitter was analyzed using link analysis and semantic analysis. This study analyzes the communication based on reply network and information diffusion based on the retweet network and predicted the strongest supporter network. 21,000 tweets were manually labelled and used to train machine learning techniques, i.e., SVM, NB, and KNN, to predict the strongest support network of a leader based on the sentiments. The dataset analyzed in this study was taken prior to the election in October 2017. It was concluded that SVM works better and has an accuracy of 79.89% on the dataset used in this study, which is higher than NB (75.40%) and KNN (65.35%).

This study also concludes that IK has the strongest support network and is more influential on Twitter compared to MNS and BBZ. This can assist and lead us in predicting the results of the election. Besides, the supporters of every political leader have some flexibility and they use to communicate with other leader's supporters. It was also observed that IK is more vigilant in his content by using social media. The proposed method can help political leaders, political parties, and government organizations in decision making for devising policies and their future plans. The proposed method can be used to support government organizations in many ways, i.e., identifying some groups that secretly work against the government. Besides, it can also find out the links between followers of different religions, actors, gamers, and supporters of any products.

There are some limitations in this study. The communities having small sizes were not analyzed. The number of communication links between two supporters' networks and their weights was not studied. The topics discussed and the issues raised in these tweets were not focused on. The trends and the issues that arose in the tweets concerning these politicians will be studied in the future. Some famous slogans can be investigated, which is more famous among the people of Pakistan. Detection of communities using edge content will be investigated in the future. Besides this, the Tweets posted in the Urdu language are also worth exploring in the future.

Data Availability

The data used in this study are available on Twitter and can be accessed using Twitter API. The python code for the Twitter API can be provided on request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

Asif Khan, Huaping Zhang, Jianyun Shang, Lin Dai, and Arshad Ahmad were involved in the conceptualization. Asif Khan, Huaping Zhnag, Nada Boudjellal, Jianyun Shang, Lin Dai, and Asmat Ali were responsible for the investigation and analysis. Asif Khan, Hauping Zhang, Nada Boujdlella, Jianyun Shang, Lin Dai, and Arshad Ahmad were involved in the methodology. Asif Khan, Nada Boudjellal, and Asmat Ali were responsible for the data collection. Asif Khan, Arshad Ahmad, and Asmat Ali were involved in the data labeling and validation. Asif Khan and Nada Boudjellal were responsible for experimentation. Asif Khan was responsible for original drafting. Asif Khan, Huaping Zhang, Jianyun Shang, Lin Dai, and Asmat Ali were responsible for visualization. All authors were involved in the results validation. All authors were responsible for reviewing and editing. Huaping Zhang, Lin Dai, and Jianyun Shang were involved in the administration. Huaping Zhang, Jianyun Shang, and Lin Dai were involved in the supervision. Huaping Zhang, Jianyun Shang, and Lin Dai were responsible for funding acquisition.

Acknowledgments

The research work was funded by the National Key Research and Development Project of China (Grant No. 2018YFC0832304) and by the National Science Foundation of China (Grant No. 61772075). The authors are thankful to them for their financial support.

References

[1] A. Ahmad, C. Feng, S. Ge, and A. Yousif, "A survey on mining stack overflow: question and answering (Q&A) community," *Data Technologies and Applications*, vol. 52, no. 2, pp. 190–247, 2018.

[2] A. Ahmad, C. Feng, K. Li, S. M. Asim, and T. Sun, "Toward empirically investigating non-functional requirements of iOS developers on stack overflow," *IEEE Access*, vol. 7, pp. 61145–61169, 2019.

[3] A. Ahmad, K. Li, C. Feng, and T. Sun, "An empirical study on how iOS developers report quality aspects on stack overflow," *International Journal of Machine Learning and Computing*, vol. 8, no. 5, pp. 501–506, 2018.

[4] A. Ahmad, *Research on Comprehending Software Requirements on Social Media*, Beijing Institute of Technology, Beijing, China, 2018.

[5] A. Ahmad, "An empirical evaluation of machine learning algorithms for identifying software requirements on stack

overflow: initial results," in *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences (ICSESS)*, Beijing, China, October 2019.

[6] B. J. Brands, T. Graham, and M. Broersma, "Social media sourcing practices: how Dutch newspapers use tweets in political news coverage," *Managing Democracy in the Digital Age*, Springer, Berlin, Germany, pp. 159–178, 2017.

[7] G. Enli and L. T. Rosenberg, "Trust in the age of social media: populist politicians seem more authentic," *Social Media + Society*, vol. 4, no. 1, 2018.

[8] M. A. Baum and P. B. K. Potter, "Media, public opinion, and foreign policy in the age of social media," *The Journal of Politics*, vol. 81, no. 2, pp. 747–756, 2019.

[9] S. A. Eldridge, L. García-Carretero, and M. Broersma, "Disintermediation in social networks: conceptualizing political actors' construction of publics on twitter," *Media and Communication*, vol. 7, no. 1, pp. 271–285, 2019.

[10] M. Lalancette and V. Raynauld, "The power of political image: justin trudeau, instagram, and celebrity politics," *American Behavioral Scientist*, vol. 63, no. 7, pp. 888–924, 2019.

[11] A. Hellweg, "Social media sites of politicians influence their perception by constituents," *The Elon Journal of Undergraduate Research in Communications*, vol. 2, no. 1, pp. 22–36, 2011.

[12] A. Ceron, L. Curini, S. M. Iacus, and G. Porro, "Every tweet counts? How sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to Italy and France," *New Media & Society*, vol. 16, no. 2, pp. 340–358, 2014.

[13] M. Broersma and T. Graham, "Social media as beat," *Journalism Practice*, vol. 6, no. 3, pp. 403–419, 2012.

[14] C. Vaccari and A. Valeriani, "Follow the leader! Direct and indirect flows of political communication during the 2013 Italian general election campaign," *New Media & Society*, vol. 17, no. 7, pp. 1025–1042, 2015.

[15] W. J. Grant, B. Moon, and J. Busby Grant, "Digital dialogue? Australian politicians' use of the social network tool twitter," *Australian Journal of Political Science*, vol. 45, no. 4, pp. 579–604, 2010.

[16] A. L. Walsh, M. E. Peters, R. L. Saralkar, and M. S. Chisolm, "Psychiatry residents integrating social media (PRISM): using twitter in graduate medical education," *Academic Psychiatry*, vol. 43, no. 3, pp. 319–323, 2019.

[17] B. N. Reames, K. H. Sheetz, M. J. Englesbe, and S. A. Waits, "Evaluating the use of twitter to enhance the educational experience of a medical school surgery clerkship," *Journal of Surgical Education*, vol. 73, no. 1, pp. 73–78, 2016.

[18] A. Rasool, R. Tao, K. Marjan, and T. Naveed, "Twitter sentiment analysis: a case study for apparel brands," *Journal of Physics: Conference Series*, vol. 1176, no. 2, Article ID 022015, 2019.

[19] J. G. D. Harb and K. Becker, "Comparing emotional reactions to terrorism events on twitter," *Communications in Computer and Information Science*, vol. 926, pp. 107–122, 2019.

[20] S. Ali, N. Islam, A. Rauf, I. Din, M. Guizani, and J. Rodrigues, "Privacy and security issues in online social networks," *Future Internet*, vol. 10, no. 12, p. 114, 2018.

[21] E. Bonsón, D. Perea, and M. Bednárová, "Twitter as a tool for citizen engagement: an empirical study of the Andalusian municipalities," *Government Information Quarterly*, vol. 36, no. 3, pp. 480–489, 2019.

[22] A. Haro-de-Rosario, A. Sáez-Martín, and M. del Carmen Caba-Pérez, "Using social media to enhance citizen engagement with local government: twitter or facebook?" *New Media & Society*, vol. 20, no. 1, pp. 29–49, 2018.

[23] A. Oehmichen, K. Hua, J. Amador Diaz Lopez, M. Molina-Solana, J. Gomez-Romero, and Y.-k. Guo, "Not all lies are equal. A study into the engineering of political misinformation in the 2016 US presidential election," *IEEE Access*, vol. 7, pp. 126305–126314, 2019.

[24] C. Featherstone, "South African bot behaviour post the July 2018 twitter account cull," in *Proceedings of the ICABCD 2019—2nd International Conference on Advances in Big Data, Computing and Data Communication Systems*, Winterton, South Africa, 2019.

[25] I. Clarke and J. Grieve, "Stylistic variation on the Donald Trump twitter account: a linguistic analysis of tweets posted between 2009 and 2018," *PLoS One*, vol. 14, no. 9, Article ID e0222062, 2019.

[26] R. Tromble, "The great leveler? comparing citizen-politician twitter engagement across three Western democracies," *European Political Science*, vol. 17, no. 2, pp. 223–239, 2018.

[27] T. Mahmood, T. Iqbal, F. Amin, W. Lohanna, and A. Mustafa, "Mining twitter big data to predict 2013 Pakistan election winner," in *Proceedings of the 2013 16th International Multi Topic Conference, INMIC 2013*, pp. 49–54, Lahore, Pakistan, 2013.

[28] P. Sharma and T. S. Moh, "Prediction of Indian election using sentiment analysis on Hindi twitter," in *Proceedings of the 2016 IEEE International Conference on Big Data*, pp. 1966–1971, Washington, DC, USA, 2016.

[29] N. Dokoohaki, F. Zikou, D. Gillblad, and M. Matskin, "Predicting Swedish elections with Twitter: a case for stochastic link structure analysis," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2015*, pp. 1269–1276, Paris, France, August 2015.

[30] B. Li, D. Guo, M. Chang, M. Li, and A. Bian, "The prediction on the election of representatives," in *Proceedings of the 2017 International Conference on Security, Pattern Analysis, and Cybernetics, SPAC 2017*, Shenzhen, China, December 2017.

[31] Z. Xie, G. Liu, J. Wu, L. Wang, and C. Liu, "Wisdom of fusion: prediction of 2016 Taiwan election with heterogeneous big data," in *Proceedings of the 2016 13th International Conference on Service Systems and Service Management, ICSSSM 2016*, Kunming, China, 2016.

[32] P. Verweij, "Twitter links between politicians and journalists," *Journalism Practice*, vol. 6, no. 5-6, pp. 680–691, 2012.

[33] A. Kleftodimos, G. Lappas, A. Triantafyllidou, and A. Yannacopoulou, "Investigating the Greek political twittersphere: Greek MPS and their twitter network," in *Proceedings of the South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference*, Kunming, China, 2018.

[34] G. Stamatelatos, S. Gyftopoulos, G. Drosatos, and P. S. Efraimidis, "Deriving the political affinity of twitter users from their followers," in *Proceedings of the 16th IEEE International Symposium on Parallel and Distributed Processing with Applications*, pp. 1175–1182, Melbourne, Austalia, 2019.

[35] R. Castro, L. Kuffó, and C. Vaca, "Back to #6D: predicting Venezuelan states political election results through twitter," in *Proceedings of the 2017 4th International Conference on eDemocracy and eGovernment, ICEDEG 2017*, pp. 148–153, Quito, Ecuador, 2017.

[36] M. Yoshida and F. Toriumi, "Analysis of political party twitter accounts' retweeters during Japan's 2017 election," in *Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intelligence*, Santiago, Chile, 2018.

[37] L. Wang and J. Q. Gan, "Prediction of the 2017 French election based on twitter data analysis," in *Proceedings of the*

2017 9th Computer Science and Electronic Engineering Conference, pp. 89–93, Coclchester, 2017.

[38] L. Wang and J. Q. Gan, "Prediction of the 2017 French election based on twitter data analysis using term weighting," in *Proceedings of the 2017 10th Computer Science and Electronic Engineering Conference*, Colchester, UK, 2019.

[39] Z. Xie, G. Liu, J. Wu, and Y. Tan, "Big data would not lie: prediction of the 2016 Taiwan election via online heterogeneous information," *EPJ Data Science*, vol. 7, no. 1, p. 1, 2018.

[40] S. Ahmed and M. M. Skoric, "My name is Khan: the use of twitter in the campaign for 2013 Pakistan general election," in *Proceedings of the 2014 Annual Hawaii International Conference on System Sciences*, pp. 2242–2251, Waikoloa, HI, USA, 2014.

[41] M. Bilal, S. Asif, S. Yousuf, and U. Afzal, "2018 Pakistan general election: understanding the predictive power of social media," in *Proceedings of the 12th International Conference on Mathematics, Actuarial Science, Computer Science*, Karachi, Pakistan, 2019.

[42] P. Aragón, K. E. Kappler, A. Kaltenbrunner, D. Laniado, and Y. Volkovich, "Communication dynamics in twitter during political campaigns: the case of the 2011 Spanish national election," *Policy & Internet*, vol. 5, no. 2, pp. 183–206, 2013.

[43] Twitter Apps, Twitter developers https://apps.twitter.com/.

[44] Tweepy. [Online]. Available: https://github.com/tweepy/tweepy.

[45] Twitter Developer. https://developer.twitter.com/en/docs.

[46] No Title. [Online]. Available https://github.com/irfreitas/tweepy_mysql_follower_scrape.

[47] J. Fernquist, L. Kaati, and R. Schroeder, "Political bots and the Swedish general election," in *Proceedings of the 2018 IEEE International Conference on Intelligence and Security Informatics, ISI 2018*, pp. 124–129, Miami, FL, USA, November 2018.

[48] J. A. Caetano, J. Almeida, and H. T. Marques-Neto, "Characterizing politically engaged users' behavior during the 2016 us presidential campaign," in *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018*, pp. 523–530, Barcelona, Spain, 2018.

[49] C. Shao, P.-M. Hui, P. Cui, X. Jiang, and Y. Peng, "Tracking and characterizing the competition of fact checking and misinformation: case studies," *IEEE Access*, vol. 6, pp. 75327–75341, 2018.

[50] F. Masood, G. Ammad, A. Almogren et al., "Spammer detection and fake user identification on social networks," *IEEE Access*, vol. 7, pp. 68140–68152, 2019.

[51] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer, "BotOrNot," in *Proceedings of the 25th International Conference Companion on World Wide Web—WWW'16 Companion*, pp. 273-274, Montréal, Canada, 2016.

[52] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[53] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 1, Article ID P10008, 2008.

[54] W. S. Noble, "What is a support vector machine?" *Nature Biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[55] K. M. Leung, "Naive Bayesian classifier," *Lecture Notes*, Springer, Berlin, Germany, 2007.

[56] S. Singh, J. Haddon, and M. Markou, "Nearest-neighbour classifiers in natural scene analysis," *Pattern Recognition*, vol. 34, no. 8, pp. 1601–1612, 2001.

[57] M. Bastian, S. Heymann, and M. Jacomy, "GEPHI: an open source software for exploring and manipulating networks," in *Proceedings of the 3rd International AAAI Conference Weblogs Social Media*, Ann Arbor, MI, USA, 2009.

Hindawi
Scientific Programming
Volume 2020, Article ID 8616039, 10 pages
https://doi.org/10.1155/2020/8616039

Research Article

Failure Prediction of Aircraft Equipment Using Machine Learning with a Hybrid Data Preparation Method

Kadir Celikmih,[1] Onur Inan,[2] and Harun Uguz[3]

[1]*Department of Information and Communication Technologies, Havelsan, Ankara 06510, Turkey*
[2]*Department of Computer Engineering, Necmettin Erbakan University, Konya 42090, Turkey*
[3]*Department of Computer Engineering, Konya Technical University, Konya 42250, Turkey*

Correspondence should be addressed to Harun Uguz; huguz@ktun.edu.tr

Received 12 January 2020; Revised 17 February 2020; Accepted 4 August 2020; Published 28 August 2020

Academic Editor: Rahman Ali

There is a large amount of information and maintenance data in the aviation industry that could be used to obtain meaningful results in forecasting future actions. This study aims to introduce machine learning models based on feature selection and data elimination to predict failures of aircraft systems. Maintenance and failure data for aircraft equipment across a period of two years were collected, and nine input and one output variables were meticulously identified. A hybrid data preparation model is proposed to improve the success of failure count prediction in two stages. In the first stage, ReliefF, a feature selection method for attribute evaluation, is used to find the most effective and ineffective parameters. In the second stage, a *K*-means algorithm is modified to eliminate noisy or inconsistent data. Performance of the hybrid data preparation model on the maintenance dataset of the equipment is evaluated by Multilayer Perceptron (MLP) as Artificial Neural network (ANN), Support Vector Regression (SVR), and Linear Regression (LR) as machine learning algorithms. Moreover, performance criteria such as the Correlation Coefficient (CC), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) are used to evaluate the models. The results indicate that the hybrid data preparation model is successful in predicting the failure count of the equipment.

1. Introduction

Reliability and availability of aircraft components have always been an important consideration in aviation. Accurate prediction of possible failures will increase the reliability of aircraft components and systems. The scheduling of maintenance operations help determine the overall maintenance and overhaul costs of aircraft components. Maintenance costs constitute a significant portion of the total operating expenditure of aircraft systems.

There are three main types of maintenance for equipment: corrective maintenance, preventive maintenance, and predictive maintenance [1]. Corrective maintenance helps manage repair actions and unscheduled fault events, such as equipment and machine failures. When aircraft equipment fails while it is in use, it is repaired or replaced. Preventive maintenance can reduce the need for unplanned repair operations. It is implemented by periodic maintenance to avoid equipment failures or machinery breakdowns. Tasks for this type of maintenance are planned to prevent unexpected downtime and breakdown events that would lead to repair operations. Predictive maintenance, as the name suggests, uses some parameters which are measured while the equipment is in operation to guess when failures might happen. It intends to interfere with the system before faults occur [1, 2] and help reduce the number of unexpected failures by providing the maintenance personnel with more reliable scheduling options for preventive maintenance. Assessing system reliability is important to choose the right maintenance strategy.

Machine learning is a rising technology that is supposed to develop in the future. Machine learning methods are applied in prediction/preventive systems, communications, security, energy management, and so on [3]. The data preparing level is the core module of machine learning and

the decision making system. It manages the data to make it useful for decision. The decision making depends on future forecasting, failure event, and availability of equipment [4]. Data mining is a way of classifying and clamping data into comprehensible information. It comprehends the applicable models from a mass of information and adopts different approaches to uncover secret data. Data mining can be defined as knowledge derivation from raw data [5].

Feature selection is a fundamental issue in data mining and machine learning algorithms that focus on the features which are the most relevant to the intended prediction [6]. Features collected from the observation of a circumstance are not all equivalently significant. Normally, operational data tend to be incomplete, insufficient, or partially meaningful or not meaningful at all. Some of them may be noisy, redundant, or irrelevant. Feature selection aims to choose a feature set that is relevant to a specific duty. This problem is a complex and multidimensional issue [7]. Hsu [8] proposed a novel feature selection algorithm based on the correlation coefficient clustering method. It focused on reducing noisy, repeated, or redundant features. The performance in the computational speed and the classification accuracy can be improved through the removal of the irrevelant features. Methods of data processing helps improve the quality of the data and increase the accuracy of data mining, thereby making it more efficient [8]. Data quality is important for the process of information discovery, checking data anomalies, and predicting and analyzing for decision making [9]. Predicting equipment failures are essential to reduce repair and equipment costs and to assess equipment availability [1].

Mass data can be useful for businesses and can guide systems to follow right paths. To boost performance in machine learning algorithms, it is critical that meaningful information be gathered from the dataset. To eliminate noisy and irrevelant data during data preparation, we used *K*-means clustering algorithm, which is one of the popular unsupervised machine learning algorithms. It defines k number of centroids and allocates every case to the nearest cluster while keeping the centroids small [10]. The "means" in the *K*-means refers to the averaging of the dataset to find the centroid. This algorithm assigns each case to only a single set. The purpose is to accomplish a high level of similarity within the clusters and low similarly across them [11]. It is used for more effective and better clustering with decreased complexity.

There are many studies on maintenance data and forecasting failure rates. Data preparation is a critical step in the feature selection process, and it has a major effect on the success of a machine learning algorithm. Gurbuz et al. [9] applied various techniques of preprocessing along with feature selection on 15 datasets of a Turkish airline company to understand and clean the dataset and to find the relationships between input and output features. They came up with 15 rules for creating failure alerts, and these were found useful by the experts of the aviation company. Classification algorithm was used to extract patterns within equipment data. Kutlylowska [12] proposed the application of artificial neural networks to failure rate modelling. Data from a water utility in Poland were used to estimate the output values of failure frequency. The results showed that artificial networks could be an option to assess the frequency of problems in the systems of water supply. Ramos et al. [13] carried out a study to predict malfunctions and to do predictive maintenance practices in a piece of manufacturing equipment. In this study, ARIMA forecasting methods were successfully compared with neural network models. The results indicated that both models were good at forecasting defibrator disc replacement, but the ARIMA was much better in the forecasting the distance between the discs. Trani et al. [14] introduced a basic method to estimate aircraft fuel consumption through the use of an artificial neural network. A fuel consumption model supported by a neural network was created by using the data given in the performance manual of the aircraft. The results from the neural network model were compared with analytical models. The results revealed that the proposed three-layer ANN with nonlinear transfer functions could correctly estimate fuel consumption in different stages of a flight. Ming et al. [15] investigated the use of the ANN method in vibration analysis by using the integrated data from the devices of vibration to predict equipment failures. The ANN model was applied to diagnose the faults in a mill. The results lent support to the efficiency of this methodology. Kozik and Sep [16] applied ANN forecasting to identify the demand for spare parts to be replaced during aircraft engine overhaul. The results indicated that the forecasting method that is composed of the engine's hardtime calculation should be a power in the implementation of lean manufacturing in MRO (maintenance, repair, and overhaul) facilities. Altay et al. [17] used 532 failures of 60 aircraft to model an artificial neural network to forecast failures. The proposed model produced high correlation rates of prediction between the actual and target failure times of aircraft. Benkedjouh et al. [18] proposed a method to guess the useful life (RUL) in machinery with bearings. For this purpose, the researchers used the isometric feature mapping reduction technique (ISOMAP) and support vector regression (SVR). Moura et al. [19] presented an analysis to comparatively assess the SVM effectiveness in predicting failure time. The performance of SVM regression is compared with other learning methods.

This paper discusses the feature selection of variables in the maintenance data obtained from an aviation company in Turkey. The proposed system will help companies to collect, extract, and create data to improve the maintenance actions through more accurate predictions. This study proposes a hybrid data preparation method for maintenance data and predicting failure counts of equipment by comparing the results of three different algorithms. The feature selection method (ReliefF algorihm in the present study) is used for selecting attributes, and the modified *K*-means algorithm is used to eliminate the redundant data. Three methods for predicting equipment failure counts are introduced and compared using MLP as an ANN algorithm, SVR algorithm, and LR. The next section presents an overview of the materials and methodology, followed by experimental results and conclusions.

2. Materials and Methodology

The context in which the present case study was carried out was an avitation company in Ankara, Turkey. The maintenance data were collected from the records of the

maintenance department. They included removal of equipment, repair activities, experience of the operators, flight hours of the equipment, and other information relevant to the case study.

2.1. Dataset Acquisition. In the ERP platform, a program is developed to collect data and to format the dataset for analysis through machine algorithms. The variables were grouped as the input variables, while the failure count was considered to be the output variable. Selected parameters are evaluated by the feature selection ReliefF method to find the most influential parameters that have a share in the failures.

Flow logic of the developed program is presented in Figure 1. Firstly, the selected materials' serial-numbered equipment used in the landing gear system was selected. Their maintenance and operational data were identified. Attributes of the maintenance and failure data were identified in cooperation with maintenance personnel and technicians. Each instance of a no fault found (NFF) status was examined to find confirmed failure data. The total flight hours for each piece of equipment across different aircraft were calculated for a given time period. Nine input variables that affect the failure of the equipment were determined. The failure count was calculated as an output variable. These nine input variables and an output variables were used for modelling the machine learning algorithms used in the present study (MLP, SVR, and LR).

2.2. Feature Selection and ReliefF Algorithm. Feature selection is a technique to obtain the relevant features by removing irrelevant and noisy data from the original dataset. It is the process of selecting a subset of features that could adequately depict all the datasets. The main objective of feature selection is to mine the data to obtain the minimum number of features to achieve maximum accuracy. Feature selection methods are used in data mining and machine learning, as well as in artificial intelligence. They reduce model complexity and let algorithms operate faster. Relief is a feature selection algorithm used for random selection of instances for feature weight calculation. The Relief algorithm is proposed by Kira and Rendell in 1992 [20, 21]. It estimates feature weights iteratively, according to their ability to make a distinction between neighboring models. Relief was extended to deal with noisy, irrevelant, and missing data to address multiclass issues. Kononenko [22] proposed an extension to Relief called ReliefF to address the multiclass problems. ReliefF is an extension of the Relief algorithm, which fails to remove irrelevant or incomplete features in two-class classification problems. The ReliefF algorithm finds one near miss for each different class and averages their value to revise feature weights.

2.3. Data Elimination and Modified K-Means Algorithm. *K*-means, a widely used algorithm in a wide range of applications, was first developed in 1967 by MacQueen [23]. It allows each data point to be a member of a single set. It has limitation fields: fixed K value and an initial centroid. It uses the distance criterion as the Euclidean distance measurement. Fahad and Alam [10] proposed a method by using modified *K*-means algorithm, which proved less time consuming yet more efficient in clustering. The quality of the resulting clusters depends on the selection of the initial centroid. *K*-means algorithm makes it possible to create a new data cluster by eliminating the smallest class value represented in the cluster. Yilmaz et al. [11] proposed a system using modified *K*-means algorithm to eliminate noisy and irrevelant data. In this study, we used modified *K*-means algorithm as in [11] and developed the pseudocode given in Algorithm 1.

2.4. Prediction Methods. In recent years, research on machine learning algorithms and data mining has been carried out to study failure prediction applications. In this study, the MLP, SVR, and LR algorithms were examined to model maintenance data and predict the failure count.

2.4.1. Multilayer Perceptron as an Artificial Neural Network. An Artificial Neural Network (ANN) is a mathematical model based on a biological interconnected group of artificial neurons. ANN imitates the brain's ability to process the information approach to computation [24, 25]. Neural networks are a nonlinear statistical data modelling and machine learning method. They can be used to model complex nonlinear relationships between inputs and outputs in the data. They also describe patterns or relationships in the data, and they help forecast output values with the help of training, learning, and testing processes. A cell in a neural network is called a neuron, and a fixed number of neurons build a layer. Neurons connect to other neurons in other layers by a weight factor. ANN algorithms compute weights for input values, hidden layer, and output layer neurons by a feed forward approach [26, 27]. Weights in an ANN are calculated by using a training algorithm as the most popular backpropagation algorithm. Backpropagation is a learning algorithm that seeks to minimize the difference between the real and target outputs. The weights are updated, so that the total error is distributed to the various neurons in the neural network. The error remains at a low level through feeding forward and backpropagating [17]. The predictive capability of neural networks comes from their multilayered structure. Neural networks have an input layer, one or more hidden layers, and an output layer. MLP algorithms are comprised of the activation function of the neurons [28]. In this study, multilayer perceptron (MLP) feed forward neural networks were used with a backpropagation learning algorithm.

2.4.2. Support Vector Regression. Support Vector Machines (SVM) algorithm was introduced by Cortes and Vapnik in 1995 [29]. It is a linear model used to address classification and regression problems. The SVM algorithm produces a hyperplane that classifies the data. There are two distinct classes separated by a linear plane. The training in the algorithm involves the process of identifying the parameters [11]. Support Vector Regression (SVR) is a regression

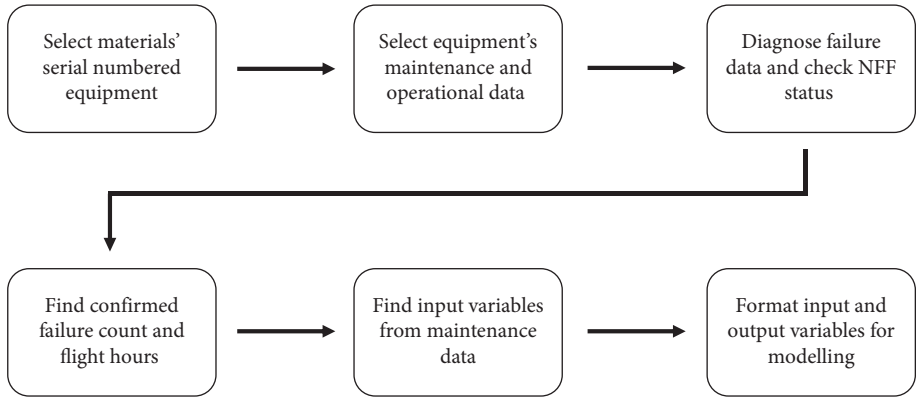


Figure 1: Flow logic of developed program.

```
(1) Procedure prepare_data_set.
(2) Get Clustured_dataset, distance center of cluster, elimination_number
(3) For i = 1 to cluster_count
(4)     Sort distance of dataset_cluster(i) to cluster_center(i) descending
(5)     For j = 1 to elimination_number
(6)         Delete j. data in clustered_dataset(i)
(7)     End for
(8) End for
(9) End procedure
```

Algorithm 1: Pseudocode of modified K-means algorithm.

algorithm that uses a similar method of SVM to carry out regression analysis. SVR is a supervised machine learning algorithm and an effective method which can be used for prediction and data mining and is successfully adopted for regression problems.

2.4.3. Linear Regression. Linear regression is defined as a machine learning algorithm that is based on supervised learning, involving a regression task. It is used to model the linear relationship among dependent variables or independent variables. It helps determine the relationship between variables and prediction. Schuld et al. [30] proposed a prediction algorithm on a quantum computer, based on a linear regression model with least-squares optimization. Its scheme focused on the machine learning task of assuming the output corresponding to a new input. The prediction result can be used for further quantum information processing routines.

2.5. Evaluation Performance Measures. In this study, the mean absolute error (MAE), root mean square error (RMSE), and correlation coefficient (CC) criteria were used to evaluate the success of the all the models. There are many error measurement techniques, and they are most commonly used to quantify error measures. The error parameters, adopted from [31], are presented in the following equations, respectively.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|X_i - Y_i\right|, \tag{1}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(X_i - Y_i\right)^2}, \tag{2}$$

$$\mathrm{CC} = \frac{\sum_{i=1}^{N}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum_{i=1}^{N}\left(X_i - \overline{X}\right)^2}\sqrt{\sum_{i=1}^{N}\left(Y_i - \overline{Y}\right)^2}}, \tag{3}$$

where N is the number of data; X_i is the observed value; Y_i is the predicted value; $\overline{X}$ is the mean of the observed data, and $\overline{Y}$ is the mean of the observed data and predicted data values. CC measures the variability of the observed data defined by the model as a correlation coefficient.

3. Proposed Methods

In this study, as noted in Section 2.1, the 585-line maintenance data in two years from a Turkish aviation company were used. The dataset consists of nine input variables and an output variable (failure count). The input variables/factors are operational and environmental parameters which could influence failure occurrence and the length of operation before failures occur. Input variables include such parameters as flight hours, the number of removals of equipment,

Table 1: The nine input variables and an output variable obtained from the maintenance data.

Parameter	Description
Flight hours (FH)	The total duration of flight for an equipment on different aircraft in a selected time period
RM	The number of removals of the equipment in the last 24 months
PR	The number of planned removals of the equipment in the last 24 months
UR	The number of unplanned removals of the equipment in the last 24 months
OR	The number of other removals of the equipment in the last 24 months
FR	The number of faults with removals of the equipment in the last 24 months
FPR	The number of faults with planned removals of the equipment in the last 24 months
FUR	The number of faults with unplanned removals of the equipment in the last 24 months
SR	The number of safe removals of the equipment in the last 24 months
NF (output)	The number of equipment failures in the last 24 months

Table 2: Sample maintenance data.

FH	RM	PR	UR	OR	FR	FPR	FUR	SR	NF
272.8	8	0	8	0	7	0	7	1	7
332.5	6	1	6	1	3	0	3	3	3
329.1	8	0	8	0	6	0	5	2	6
285.2	8	0	7	0	7	1	6	1	7
433.7	12	0	11	0	9	0	10	2	9

Table 3: Description of the selected attributes used in modelling.

Description	The full name of the equipment
Flight hours (FH)	The duration flight of the equipment on the aircraft or different aircraft in the selected period
RM	The number of removals of the equipment in the last 24 months
UR	The number of unplanned removals of the equipment in the last 24 months
SR	The number of safe removals of the equipment in the last 24 months
NF (output)	The number of equipment failures in the last 24 months

and the number of faults with planned/unplanned removals. These data were analysed and represented in a format suitable for modelling, and variables were characterised with the corresponding domain classification, shown in Table 1. The output variable is the number of equipment failures. A sample of the dataset is provided in Table 2.

Feature selection is carried out using these ten attributes. The number of equipment failures is the target of the analysis. For this purpose, feature selection ReliefF algorithm was used to find relations and weighting coefficient dependencies. According to the ranked values, four most effective attributes were selected (Table 3).

Noisy and inconsistent data in the prepared datasets often affect prediction negatively and reduce the performance of the system. Therefore, the modified K-means algorithm was used to eliminate the noisy and inconsistent data to increase the performance of the prediction. It was developed using the pseudocode given in Algorithm 1. In this model, set centers are initially allocated, and instances are properly distributed to the sets. A predetermined number ($N = 5$) of records furthest to the center in each set were eliminated. The distance criterion was the Euclidean distance measurement. The eliminated instances are shown in Figure 2.

Seventy-five records (approximately 13%) of the dataset were eliminated by the pseudocode of the proposed data preparation model. Five hundred and ten records were obtained from 585 records of the dataset. Our proposed hybrid data preparation model is comprised of two stages, as shown in Figure 3. In the first stage, nine input attributes were reduced to four attributes by feature selection ReliefF algorithm. In the second stage, the dataset was reduced to 510 records by the modified K-means algorithm. The obtained dataset with 510 records were provided as inputs to the MLP, SVM, and LR prediction algorithms.

4. Experimental Results

A program is developed to gather data for analysis through machine algorithms. Selected equipment's maintenance and operational data were identified. Nine input variables and an output variable were determined. According to using pure 585 rows, nine inputs, and an output (585×10) data, MLP, LR, and SVR models were trained and tested. The parameters of the predictors used in the study are provided in Tables 4–6, respectively.

To illustrate the performance of the suggested two-phase hybrid system, the prediction results for the raw dataset that

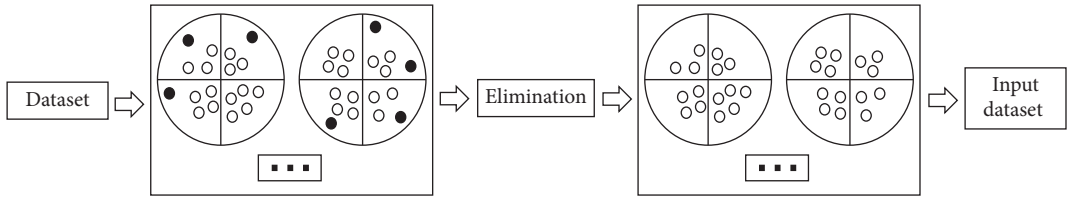


Figure 2: Data preparation model in modified *K*-means algorithm.

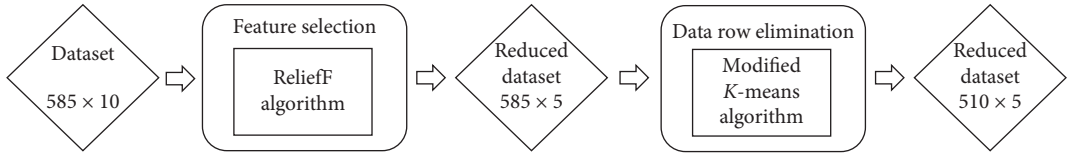


Figure 3: Block diagram of the proposed hybrid data preparation model.

Table 4: Tuned SVR parameters.

Parameters	Description
C	1.0
ε	0.001
Validation method	10-fold cross validation
Kernel function	Linear

Table 5: Tuned MLP Parameters.

Parameters	Description
Number of neurons for the hidden layer	2
Hidden layers	2
Learning rate	0.2
Momentum	0.2
Epoch	500
Goal	0.005

Table 6: Tuned LR parameters.

Parameters	Description
Batch size	100
Attribute selection method	No attribute selection
Ridge	$1.0E-8$

Table 7: Performance rating of models for (585 × 10) dataset (9 inputs 1 output).

Method	CC	MAE	RMSE
LR	0.8967	0.7341	1.0646
MLP	0.8925	0.8125	1.0992
SVR	0.9008	0.741	1.0889

is composed of 585 records and 9 attributes are presented in Table 7. Table 7 shows that based on the CC performance criterion, the best results were provided by the SVR algorithm, while the LR algorithm provided the best results based on the MAE and RMSE performance criteria.

Then, the ReliefF algorithm was applied to the raw data to identify the features that prove the most effective in prediction. Feature selection ReliefF algorithm was applied to nine input parameters, and according to ranked values, the last five of them were eliminated. MLP, LR, and SVR models are built with selected 585 rows, four inputs, and an output. The dataset (585 × 5) was trained and tested. As seen in Table 8, all the results are better than those obtained without feature selection. The results provided by the MLP (CC = 0.9127, MAE = 0.7301, RMSE = 0.9853) is better than those of the other algorithms. The performance results for the error parameters in each prediction algorithm are provided in Table 8.

In the final phase, the modified *K*-means algorithm was applied to the dataset to eliminate noisy and inconsistent data (585 × 5). The best k value was found to be ($k = 15$). Five parameters that were farthest from the center of each cluster were eliminated. As a result, 75 rows were eliminated. So, a hybrid model approach was applied to the maintenance data, and the quality of the data was improved. The LR, MLP, and SVR models were built with the selected 510 rows, four inputs, and an output. The selected data (510 × 5) were trained and tested. The results indicated that the performance of the model was highly successful, compared to the other results obtained without feature selection and data reduction. The performance of the LR, MLP, and SVR algorithms are presented in Table 9.

As shown in Table 9, based on the CC, MAE, and RMSE performance criteria, the best results were provided by the LR algorithm in the suggested two-phase hybrid system. For the test data, CC = 0.9316, MAE = 7108, and RMSE = 0.835 were obtained.

Figures 4–6 provide the linear correlation between the predicted and target results for the test data of LR, SVR, and MLP, respectively. The results indicated that the regression line in the test and the predicted data of LR provide $y_1 = 0.9976x + 0.0155$; those of SVR provide $y_2 = 0.9184x + 0.48$, and those of MLP provide $y_3 = 0.9999x - 0.0744$.

Table 8: Performance rating of the models for (585 × 5) dataset (4 inputs and 1 output).

Method	CC	MAE	RMSE
LR	0.8967	0.7341	1.0646
MLP	0.9127	0.7301	0.9853
SVR	0.9013	0.7415	1.0909

Table 9: Performance rating of models for (510 × 5) dataset (4 inputs 1 output).

Method	CC	MAE	RMSE
LR	0.9316	0.6807	0.835
MLP	0,9284	0.6816	0.8555
SVR	0.9316	0.7108	0.8558

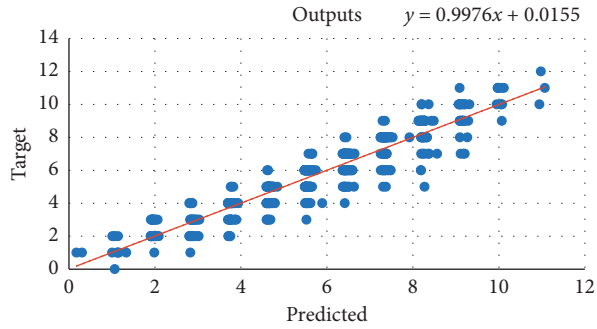


Figure 4: Correlation between predicted and target values of the dataset for LR.

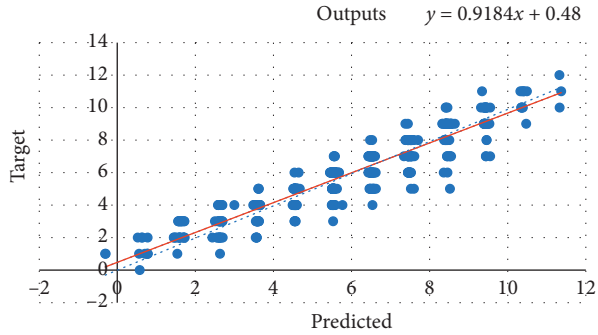


Figure 5: Correlation between predicted and target values of the dataset for SVR.

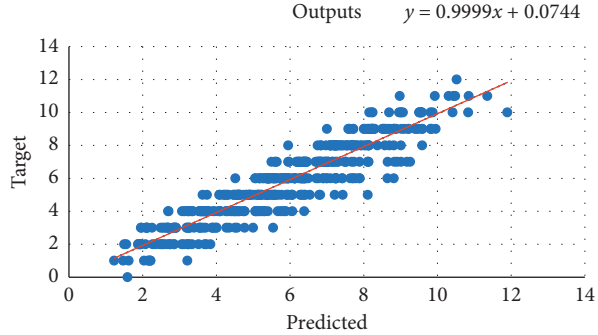


Figure 6: Correlation between predicted and target values of the dataset for MLP.

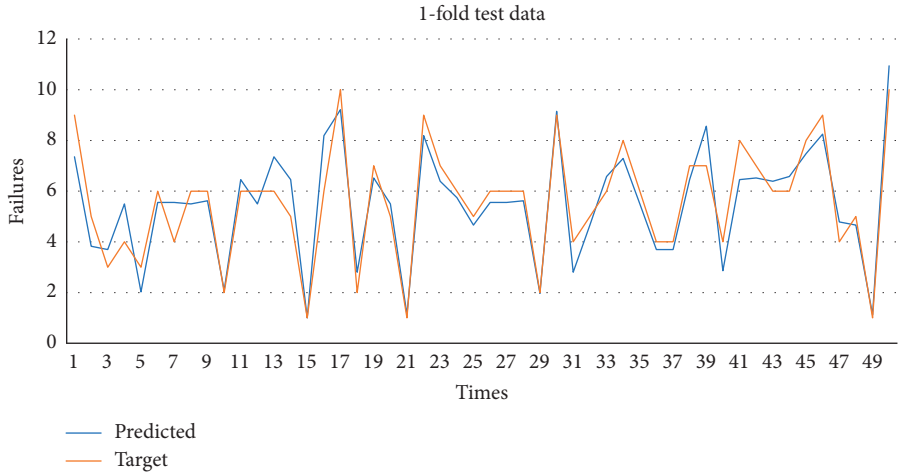


Figure 7: Test data for the prediction and target values of 1-fold CV for the LR.

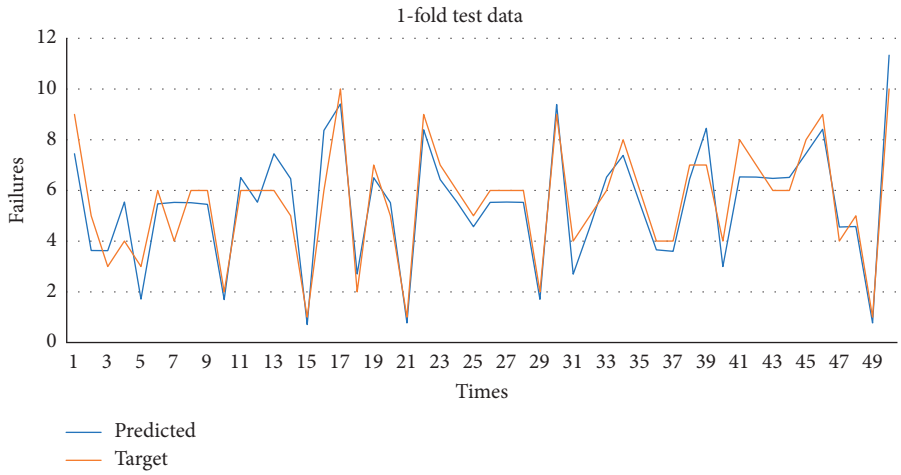


Figure 8: Test data for the prediction and target values of 1-fold CV for the SVR.

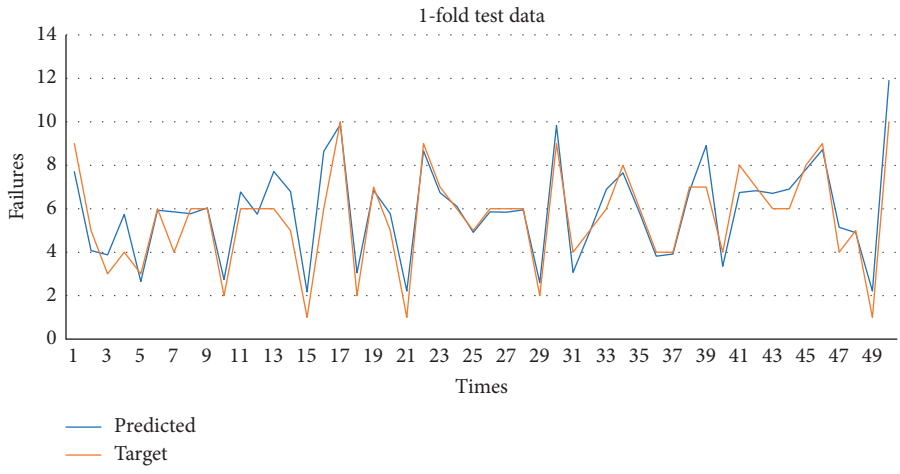


Figure 9: Test data for the prediction and target values of 1-fold CV for the MLP.

The target and predicted values in the test dataset of the suggested two-phase hybrid system were provided for the LR, SVR, and MLP in Figures 7–9, respectively. The test results provided in Figures 7–9 are the graphical representation of the test results of the 1-fold cross validation.

Table 9 presents the predicted and target values obtained through the two-phase hybrid system in Figures 7–9. As seen in Figures 7–9, the proposed hybrid data preparation model increased the performance of prediction models LR, SVR, and MLP. The suggested hybrid system helped attain higher accuracy in prediction as it enabled us to select the most effective features and eliminate noisy or redundant data that could lower the accuracy of predictions.

5. Conclusions

In aviation, the use of maintenance data is highly critical in the analysis of reliability and maintenance costs. This is because predictive maintenance scheduling can be planned in line with estimates. The main target of predictive maintenance is to predict equipment failures and planning strategies for spare parts of the system components to analyze the reliability and maintainability of a complex repairable system. In this study, a hybrid data preparation model was applied to the landing gear system maintenance dataset using feature selection ReliefF algorithm to select attributes and a modified *K*-means algorithm to eliminate noisy and inconsistent data. The proposed hybrid data preparation method was put into practice through LR, SVR, and MLP models. The results indicated that the LR model had better performance than MLP and SVR models in predicting the failure counts. The results indicate that the proposed hybrid data preparation model significantly improves the accurate prediction of failure counts. This study could function as a guide for using hybrid data preparation methods in machine learning algorithms and data mining.

Data Availability

The maintenance data used to support the findings of this study have not been made available because sharing the data might compromise data privacy. Moreover the authors are not allowed to share these data due to security concerns.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the Scientific Research Project of Havelsan and Presidency of Defence Industries project, grant no. HVL-SÖZ-18/033.

References

[1] Q. Fan and H. Fan, "Reliability analysis and failure prediction of construction equipment with time series models," *Journal of Advanced Management Science*, vol. 3, no. 3, pp. 203–210, 2015.

[2] P. Bastos, I. Lopes, and L. Pires, "A maintenance prediction system using data mining," in *Proceedings of the Proceedings of the World Congress on Engineering*, vol. 3, pp. 2–7, London, UK, July 2012.

[3] I. U. Din, M. Guizani, J. J. P. C. Rodrigues, S. Hassan, and V. V. Korotaev, "Machine learning in the Internet of Things: designed techniques for smart cities," *Future Generation Computer Systems*, vol. 100, pp. 826–843, 2019.

[4] B. Jan, H. Farman, M. Khan, M. Talha, and I. U. Din, "Designing a smart transportation system: an internet of things and big data approach," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 73–79, 2019.

[5] I. U. Din, M. Guizani, S. Hassan et al., "The internet of things: a review of enabled technologies and future challenges," *IEEE Access*, vol. 7, pp. 7606–7640, 2019.

[6] S. Lecturer and A. Pradesh, "Feature selection using ReliefF algorithm," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, no. 10, pp. 8215–8218, 2014.

[7] S. F. Rosario, "RELIEF: feature selection approach," *International Journal of Innovative Research and Development*, vol. 4, no. 11, pp. 218–224, 2015.

[8] H. Hsu, "Feature selection via correlation coefficient clustering," *Journal of Software*, vol. 5, no. 12, pp. 1371–1377, 2010.

[9] F. Gürbüz, L. Özbakir, and H. Yapici, "Data mining and preprocessing application on component reports of an airline company in Turkey," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6618–6626, 2011.

[10] S. K. A. Fahad and M. Alam, "A modified *K*-means algorithm for big data clustering," *International Journal of Computer Science Engineering and Technology*, vol. 6, no. 4, pp. 129–132, 2016.

[11] N. Yilmaz, O. Inan, and M. S. Uzer, "A new data preparation method based on clustering algorithms for diagnosis systems of heart and diabetes diseases," *Journal of Medical Systems*, vol. 38, no. 5, 2014.

[12] M. Kutyłowska, "Neural network approach for failure rate prediction," *Engineering Failure Analysis*, vol. 47, pp. 41–48, 2015.

[13] P. Ramos, J. M. Oliveira, and P. Silva, "Predictive maintenance of production equipment based on neural network autoregression and ARIMA," in *Proceedings of the 21st International EurOMA Conference-Operations Management in An Innovation Economy*, pp. 1–10, Helsinki, Finland, June 2014.

[14] A. A. Trani, F. C. Wing-Ho, G. Schilling, H. Baik, and A. Seshadri, "A neural network model to estimate aircraft fuel consumption," in *Proceedings of the AIAA 4th Aviation Technology, Integration, and Operations Forum, ATIO*, vol. 2, pp. 669–692, Chicago, IL, USA, September 2004.

[15] M. Chen, R. Zhou, R. Zhang, and X. Zhu, "Application of artificial neural network to failure diagnosis on process industry equipments," in *Proceedings 2010 6th International Conference on Natural Computation, ICNC 2010*, vol. 3, pp. 1190–1193, Yantai, China, August 2010.

[16] P. Kozik, "Aircraft engine overhaul demand forecasting using ANN," *Management and Production Engineering Review*, vol. 3, no. 2, pp. 21–26, 2012.

[17] A. Altay, O. Ozkan, and G. Kayakutlu, "Prediction of aircraft failure times using artificial neural networks and genetic algorithms," *Journal of Aircraft*, vol. 51, no. 1, pp. 47–53, 2014.

[18] T. Benkedjouh, K. Medjaher, N. Zerhouni, and S. Rechak, "Remaining useful life estimation based on nonlinear feature reduction and support vector regression," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 7, pp. 1751–1760, 2013.

[19] M. D. C. Moura, E. Zio, I. D. Lins, and E. Droguett, "Failure and reliability prediction by support vector machines regression of time series data," *Reliability Engineering & System Safety*, vol. 96, no. 11, pp. 1527–1534, 2011.

[20] K. Kira and L. A. Rendell, "Feature selection problem: traditional methods and a new algorithm," *in Proceedings Tenth National Conference on Artificial Intelligence*, pp. 129–134, 1992.

[21] K. Kira and L. A. Rendell, "A practical approach to feature selection," *Machine Learning Proceedings 1992*, vol. 1992, pp. 249–256, 1992.

[22] I. Kononenko, "Estimating attributes: analysis and extensions of RELIEF," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 784, pp. 171–182, Springer, Berlin, Germany, 1994.

[23] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, CA, USA, 1967.

[24] N. Nadai, A. H. A. Melani, G. F. M. Souza, and S. I. Nabeta, "Equipment failure prediction based on neural network analysis incorporating maintainers inspection findings," in *Proceedings of the Annual Reliability and Maintainability Symposium*, Orlando, FL, USA, January 2017.

[25] T. S. Lan, P. C. Chen, M. Y. Wang, K. S. Hsu, and T. Y. Chen, "A study of using back-propagation network to predict aircraft component life span," in 2016 International Conference on Applied System Innovation," in *Proceedings of the IEEE ICASI 2016*, Okinawa, Japan, May 2016.

[26] S. Oladokun, "Predicting mean time between failures of a maintained equipment using artificial neural network," *American Journal of Scientific and Industrial Research*, vol. 1, no. 3, pp. 500–503, 2010.

[27] P. S. Rajpal, K. S. Shishodia, and G. S. Sekhon, "An artificial neural network for modeling reliability, availability and maintainability of a repairable system," *Reliability Engineering and System Safety*, vol. 91, no. 7, pp. 809–819, 2006.

[28] M. Mahmudul, A. Mia, S. K. Biswas, M. C. Urmi, and A. Siddique, "An algorithm for training multilayer perceptron MLP for image reconstruction using neural network without overfitting," *International Journal of Scientific & Technology Research*, vol. 4, no. 2, pp. 271–275, 2015.

[29] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[30] M. Schuld, I. Sinayskiy, and F. Petruccione, "Prediction by linear regression on a quantum computer," *Physical Review A*, vol. 94, no. 2, 2016.

[31] M. Buyukyildiz and S. Y. Kumcu, "An estimation of the suspended sediment load using adaptive network based fuzzy inference system, support vector machine and artificial neural network models," *Water Resources Management*, vol. 31, no. 4, pp. 1343–1359, 2017.

Hindawi
Scientific Programming
Volume 2020, Article ID 4741963, 10 pages
https://doi.org/10.1155/2020/4741963

Research Article

Representation Learning of Knowledge Graphs with Embedding Subspaces

Chunhua Li,[1] Xuefeng Xian,[2] Xusheng Ai,[1] and Zhiming Cui[3]

[1]*Software and Service Outsourcing College, Suzhou Vocational Institute of Industrial Technology, Suzhou 215104, China*
[2]*School of Computer Engineering, Suzhou Vocational University, Suzhou 215104, China*
[3]*School of Electronics and Information Engineering, Suzhou University of Science and Technology, Suzhou 215009, China*

Correspondence should be addressed to Zhiming Cui; zmcui@usts.edu.cn

Received 12 February 2020; Revised 25 April 2020; Accepted 10 June 2020; Published 25 August 2020

Academic Editor: Kifayat Ullah Khan

Most of the existing knowledge graph embedding models are supervised methods and largely relying on the quality and quantity of obtainable labelled training data. The cost of obtaining high quality triples is high and the data sources are facing a serious problem of data sparsity, which may result in insufficient training of long-tail entities. However, unstructured text encoding entities and relational knowledge can be obtained anywhere in large quantities. Word vectors of entity names estimated from the unlabelled raw text using natural language model encode syntax and semantic properties of entities. Yet since these feature vectors are estimated through minimizing prediction error on unsupervised entity names, they may not be the best for knowledge graphs. We propose a two-phase approach to adapt unsupervised entity name embeddings to a knowledge graph subspace and jointly learn the adaptive matrix and knowledge representation. Experiments on Freebase show that our method can rely less on the labelled data and outperforms the baselines when the labelled data is relatively less. Especially, it is applicable to zero-shot scenario.

1. Introduction

There are various knowledge graphs constructed with great effort, such as Freebase [1] and WordNet [2], which have become the fundamental techniques for many intelligent applications such as web search and question answering [3]. However, the validity and integrity of the knowledge graphs cannot always be guaranteed. For instance, Freebase [1] currently contains entities over 80 million and thousands of relations as well as obtaining billions of facts about these entities. However, obviously, these are still only a small part of all human knowledge. In fact, the ability of Q&A system engine based on knowledge graphs is limited if it could not infer and fill in the missing facts from the obtained knowledge. Therefore, knowledge reasoning methods predicting new facts only based on the knowledge existing in knowledge graphs are desired. It is a key compensation for extracting relations from flat text.

Knowledge representation is the basis of knowledge reasoning. For example, using graph-based knowledge representation, to compute or infer a semantic relationship between entities needs to design specific graph-based algorithms. Knowledge graphs represent entities as nodes and relations as different types of edges in the form of a triple (head entity, relation, tail entity) [4]. Graph-based knowledge representation is facing many challenges, e.g., computing efficiency and data sparseness. In recent years, great progress has been made in the knowledge representation learning method based on embedding technology [5]. Embedding learning is to represent the entities and relations in knowledge graphs as low-dimensional dense real value vectors and embed the knowledge graph into a continuous vector space while keeping the structure characteristic of knowledge graphs [6]. Usually, in this low-dimensional dense real value vector space, the nearer the distance of two vectors, the higher the similarity of their semantics. Since the semantic relation of entities and relations can be calculated in the low-dimensional space in a highly efficient way and the problem of data sparseness can be resolved dramatically, the performance of knowledge extracting, fusion, and reasoning is greatly improved.

Most of the existing embedding learning methods are supervised methods and generally use the obtained structured knowledge to train models [7, 8]. The success of the supervised system largely depends on the quantity and quality of obtainable training data; it is sometimes even more important than the choice of specific learning algorithms. The cost of obtaining high quality structured knowledge is high and the knowledge graph obtained is facing a serious problem of data sparseness. Long-tail entities cannot be efficiently trained. On the other hand, unstructured text data involving relevant entities and relations information can be obtained easily in large quantities. Word vector representations of entity names can be obtained through cooccurrence mode of words in large quantities of an unlabelled text corpus. The word vectors estimated from the raw text through the natural language model are the low-dimensional dense vectors containing syntax and semantic attributes of words [9]. Therefore, since these vectors are obtained by minimizing the prediction errors in the common unsupervised tasks, they might not be the best for knowledge graphs.

This paper proposes a knowledge representation learning approach in which the unsupervised word vectors are adapted to knowledge graph subspaces with a small number of labelled data. The intuition behind our approach is the following. For a specific task, only partial latent aspects captured by the word vectors will be useful. Hence, instead of updating the word vectors directly with available labelled data, we estimate a projection of these vectors into a low-dimensional subspace. This simple method has two key advantages. One is that we get low-dimensional vectors, which are suitable for complex knowledge representation learning tasks. Another is that we can learn new vectors of all entities even if they are missing in labelled data.

2. Related Work

2.1. Structure-Based Knowledge Embedding. Structure-based embedding learning models learn the entity and relation vector representations through structure information located in triples of the knowledge graph. Most existing embedding methods belong to this category.

Most methods of this kind have been designed within the framework of relational learning from latent features by operating on latent representations of entities and relations, such as models based on collective matrix factorization [10, 11] or tensor factorization [7, 12, 13]. Many approaches have focused on increasing the expressivity and the generality of the model in energy-based frameworks for learning embeddings of entities in low-dimensional spaces [14–16]. The greater expressivity of these models comes at the expense of substantial increases in model complexity and in higher computational costs.

Compared to the early embedding models, TransE [8] is simpler and more effective. TransE regards the relations in the knowledge graph as certain translation vectors. For a triple (h, r, t), where h represents a head entity, and r is a relationship that connects h to a tail entity t. TransE use relation vector $\mathbf{r}$ as the translation between the head entity vector $\mathbf{h}$ and the tail entity vector $\mathbf{t}$. Therefore, TransE is also referred to as translation model. The basic idea of this model is to treat relation r as the translation between entities associated with r. If relation (h, r, t) is established, $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ in the vector space, that is $\mathbf{t}$ should be the closest vector of $\mathbf{h} + \mathbf{r}$. Otherwise, $\mathbf{h} + \mathbf{r}$ should be far away from $\mathbf{t}$.

Later knowledge embedding learning models are mostly the extension based on TransE, such as TransH [17], TransR [18], TransD [19], and TransA [20]. PTransE [21] proposes a multiple-step relation path based representation learning model. TranSparse [22] deal with the heterogeneity and the imbalance of knowledge graphs with adaptive sparse matrices. The recently proposed ProjE [23] achieves the state-of-the-art performance of this branch with relatively less model parameters.

Luo et al. [24] propose a two-stage embedding scheme to improve the performance of structure-based embedding models, such as TransE, SME, and SE. It first uses a word embedding model to learn initial embeddings of entities and relations from relation paths, viewing entities and relations in the path as pseudowords. RDF2Vec [25], metapath2vec [26] and Hussein et al. [27] transform the graph data into sequences of entities and use unsupervised language model to learn entity representations considering sequences of entities as sentences. However, these method still only utilize the structure information. Recently, Dettmers et al. [28] introduce a multilayer convolutional network model, ConvE, for link prediction, which uses 2D convolution over embeddings and multiple layers of nonlinear features to model knowledge graphs.

2.2. Knowledge Embedding With Multisources Information. Structure-based knowledge embedding learning models only utilize the triple structure information of the knowledge graph and a large amount of other related information are not efficiently used yet, such as the descriptions and categories of entities and relations.

There are some studies on using the above information to learn knowledge representation. NTN [7] represents an entity as the average of its word embeddings in entity name. Wang et al. [29] align the embeddings of entities and words in the same space by utilizing entity names and Wikipedea anchors. Recently, DKRL [30] extends TransE considering text information of entity descriptions provided by knowledge bases (i.e., knowledge graph), and building entity vector representations with CNN model based on entity descriptions, which can model the word sequence information in text. When a new entity that is not in the knowledge base occurs, DKRL can generate entity vector based on its simple description. SSP model recently proposed by Xiao et al. [31] learned entity semantic vectors from entity description text by using topic model, then projected the structural loss to the semantic space codetermined by the head entity and tail entity to learn vector representations of entities and relations. The learning process of SSP model is more closely related to text information.

Most of the above models mainly use the text information of entity names and entity descriptions. This limits

the use of abundant unstructured text information in the Web.

3. Knowledge Representation Learning Based on Subspace Projection

Some work [7, 29–31] shows that learning knowledge representation through fusing multisource information can efficiently improve the performance of knowledge representations. Abundant Web text contains large quantity of unstructured knowledge related to entities. Word embedding is a kind of useful unsupervised technology and it can offer a simplified real value vector representation for each word from unlabelled free text. We use word embedding technology to obtain vector representations of entity names from abundant Web text and then adapt these vectors to a subspace which is suitable for representations of entities in the knowledge graph through projection. Therefore, knowledge representation learning can be divided into two stages: unsupervised estimation of entity name vectors and jointly supervised learning of the subspace adaptive matrix and the knowledge representation.

3.1. Estimation of Unsupervised Entity Vectors. We obtain vector representations of entity names in knowledge graphs from unlabelled Web text through unsupervised word vector learning technology and regard it as the initial vector representation estimations of entities. Word vectors are usually trained through optimizing an objective function with unlabelled data [9, 32–34]. CBOW [34] and skip-gram [9] learn word vectors capturing many syntactic and semantic relations between words. Thus, in this study we use skip-gram [9] to learn the word vector representations of entity names.

Given a sequence of training words $w_1, w_2, \ldots, w_T$, the optimizing objective of skip-gram is to maximize the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c,\, j\neq 0}\log\, p\left(w_{t+j}\,\middle|\, w_t\right), \tag{1}$$

where c is the size of training context and $p(w_{t+j}\,|\,w_t)$ is defined using softmax function:

$$p\left(w_O\,\middle|\, w_I\right) = \frac{\exp\left(v'^{\,T}_{w_O} v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left(v'^{\,T}_{w} v_{w_I}\right)}, \tag{2}$$

where v_w is the vector representation of word w and W is the number of words in the vocabulary.

The same as other majority of neural network model, skip-gram adopts a training method based on gradient descent. The trained model, embedding vectors $v_w \in R^{e\times 1}$, encloses information of each word w and the context around it. Therefore, these vectors can be used as input of other learning algorithms to improve performances further.

3.2. Embedding Knowledge Graphs into Subspaces. As previously mentioned, word embedding is a kind of useful unsupervised technology to obtain the initial feature vector of entity names before supervised training. These initial vector representations can be retrained with obtainable labelled data. However, the knowledge graph has serious problem of data sparseness. The quantity of entities in the database is large; however the quantity of high quality triple data related to each entity that can be obtained is relatively small with high cost. Only small quantity of supervised data causes serious overfitting. Furthermore, it is likely that only a subset of entities appears in the training triples and the vector representations of the entities missing in the training triples will never be updated. We propose a simple solution to avoid this problem.

We use $\mathbf{W^E} \in R^{e\times v}$ to denote the initial entity vector matrix obtained by skip-gram as stated in the previous section. We define the adapted embedding matrix $\mathbf{W^A}$ in subspace as the following factorization:

$$\mathbf{W^A} = \mathbf{W^S}\cdot\mathbf{W^E}, \tag{3}$$

where $\mathbf{W^S} \in R^{s\times e}$, $s \ll e$. Next, we estimate parameters of the matrix $\mathbf{W^S}$ by using the triples (labelled dataset) in the knowledge graph and keep $\mathbf{W^E}$ fixed. That is to say, we find the best mapping matrix $\mathbf{W^S}$ that projects the initial vector matrix $\mathbf{W^E}$ into the subspace with dimension s.

The idea of embedding knowledge graphs into a subspace is based on the following two key principles: (1) With reduction of embedding dimension, the model can better fit the complexity of the knowledge graph task and the amount of obtainable labelled data. (2) Through projection, all the entity vectors are indirectly updated, not only those of entities that appear in training triples.

3.3. Jointly Supervised Learning of Subspace Adaptive Matrix and Knowledge Representation

3.3.1. Jointly Learning Model. After obtaining the initial vectors of entities, we use a supervised model to jointly learn the projection matrix and knowledge representation in subspace based on the idea of subspace projection. The jointly learning model utilizes the structure information of triples existing in the knowledge graph.

The concept of embedding subspace can be applied to any structure-based knowledge representation learning models. Since currently ProjE gains the best performance in relation to reasoning task and the parameters are relatively less, we use this model together with the thought of embedding subspace as a supervised training model. This method is hereinafter referred to as sub-ProjE. Let n_e, n_r, e and s be, respectively, the number of entities, relations, unsupervised entity vector dimension, and subspace dimension. The number of parameters of ProjE is $n_e \times s + n_r \times s + 5s$. The number of parameters of sub-ProjE is $s \times e + n_r \times s + 5s$. Since $e \ll n_e$, the parameter size of sub-ProjE is much smaller than that of ProjE.

We consider the relation reasoning as an entity ranking problem, which takes a partial triple as input and produces a ranked list of candidate entities as output.

Definition 1. (entity ranking problem). Given a knowledge graph $\mathcal{G} = \{\mathbf{E}, \mathbf{R}\}$ and an input tripe $(h, r, ?)$, the entity ranking problem attempts to find the optimal ordered list such that $\forall e_j \forall e_i ((e_j \in E\_ \wedge e_i \in E_+) \longrightarrow e_i \prec e_j)$, where $\mathbf{E}_+ = \{e \in \{e_1, e_2, \ldots, e_l\} \mid (h, r, e) \in \mathcal{G}\}$, $\mathbf{E}_- = \{e \in \{e_{l+1}, e_{l+2}, \ldots, e_{|E|}\} \mid (h, r, e) \notin \mathcal{G}\}$.

Similarly, we can easily substitute $(h, r, ?)$ for $(?, r, t)$. The key thought of ProjE is as follows: given two input vectors, regard prediction task as a ranking problem, keep the target of optimizing as the overall order of candidate entities, in which the entities in the front are correct entities. To generate this ordered list, we project each candidate vector to the objective vector defined by a combination operator and two input vectors. The combination operator is defined as follows:

$$\mathbf{e} \oplus \mathbf{r} = \mathbf{D}_e e + \mathbf{D}_r r + \mathbf{b}_c, \tag{4}$$

where $\mathbf{e}$ and $\mathbf{r}$ are the representations of entities and relations in embedding space, s is the dimension of the embedding space, $\mathbf{D}_e$ and $\mathbf{D}_r$ is the diagonal matrix of $s \times s$, which serve as global entity and relation weights, respectively, $\mathbf{b}_c \in R^s$ is the combination bias.

With this combination operator, we can define vector project function as follows:

$$h(\mathbf{e}, \mathbf{r}) = g\left(\mathbf{W}^c f(\mathbf{e} \oplus \mathbf{r}) + b_p\right), \tag{5}$$

where f and g are activation functions, $\mathbf{W}^c \in \mathrm{R}^{c \times s}$ is the candidate entity matrix, b_p is the projection bias, and c is the quantity of candidate entities. $h(\mathbf{e}, \mathbf{r})$ represents the ranking score vector, in which each element represents the similarity between candidate entity in $\mathbf{W}^c$ and the combined input vector.

$\mathbf{W}^c$ is the candidate entity matrix which contains c rows that exist in the entity vector matrix $\mathbf{W}^{\mathbf{E}}$ (i.e., $\mathbf{W}^{\mathbf{S}} \cdot \mathbf{W}^{\mathbf{E}}$ in knowledge graph subspace). So, $\mathbf{W}^c$ does not introduce any new variables into the model. The model can be regarded as a neural network that contains an entity vector projection layer, a combination layer, and an output projection layer. Figure 1 explains the architecture of this model through an example with input (?, City Of, Illinois). Given a tail entity Illinois and a relation City Of, our task is to calculate the scores of each head entity. In order to make it clear, we only demonstrate two candidate entities in Figure 1. However, in fact $\mathbf{W}^c$ may contain candidate entities of any quantity.

Compared to a conventional knowledge embedding model, this model has two main differences. First, the input layer is factorized into two components, the initial vector representations attained in unsupervised stage, $\mathbf{W}^{\mathrm{E}}$, and the projection matrix $\mathbf{W}^{\mathbf{S}}$. Second, the size of the subspace, in which the initial vectors are projected, is much smaller than that of the initial embedding space with typical reductions above one order of magnitude. Same as the usual neural network model, all the parameters can be trained with gradient descent methods through backpropagation.

3.3.2. Ranking Method and Loss Function. Following ProjE, we construct a binary label vector, in which all entities in $\mathbf{E}\_$ have a value of 0 and all entities in $\mathbf{E}_+$ have a value of 1, then maximize the likelihood between ranking score vector $h(\mathbf{e}, \mathbf{r})$ and the binary label vector. The loss function is defined as follows:

$$L(\mathbf{e}, \mathbf{r}, \mathbf{y}) = -\sum_i^{|y|} \frac{1(y_i = 1)}{\sum_i 1(y_i = 1)} \log\left(h(\mathbf{e}, \mathbf{r})_i\right), \tag{6}$$

where $\mathbf{e}$ and $\mathbf{r}$ are the vector representation of input training sample, $y \in R^c$ is the binary label vector, where $y_i = 1$ means candidate entity i is positive, the objective probability of a positive candidate (objective value) is 1 divided by the total number of positive candidates. We regard softmax and tanh function as $g(\cdot)$ and $f(\cdot)$, respectively, then the ranking score of the ith candidate entity is as follows:

$$h(\mathbf{e}, \mathbf{r})_i = \frac{\exp\left(\mathbf{W}^c_{[i,:]} \tanh(e \oplus r) + b_p\right)}{\sum_j \exp\left(\mathbf{W}^c_{[i,:]} \tanh(e \oplus r) + b_p\right)}, \tag{7}$$

where $\mathbf{W}^c_{[i,:]}$ represents the ith candidate in the candidate entity matrix.

3.3.3. Algorithms. Since the quantity of candidate entities (i.e., rows of $\mathbf{W}^c$) is large, we use candidate sampling to reduce the number of candidate entities in the training phase. Given an entity $\mathbf{e}$, a relation $\mathbf{r}$ and a binary label vector $\mathbf{y}$, we calculate projections of all positive candidates. For negative candidates, we only calculate projections of a sampled subset. We take negative candidate samples based on the binomial probability distribution $B(1 - p_y)$, in which p_y is the probability of a negative sample which might be sampled, $1 - p_y$ is the probability of a negative sample that is sampled. For each negative candidate in $\mathbf{y}$, we sample a value from $B(1 - p_y)$ to determine whether this candidate is included in the candidate entity matrix $\mathbf{W}^c$ or not.

The complete training process is demonstrated in Algorithm 1. Given training triples $\mathbf{T}$, we first choose at random to replace head entity or tail entity to construct real training dataset and then generate positive and negative samples from $\mathbf{T}$ according to sampling strategy. Next, calculate loss and update parameters for each minibatch in the newly generated training dataset. Among this, $\circ$ is Hadamard product and $\times$ is matrix product.

4. Experiments

We evaluate our model with entity prediction tasks and compare the performance against the native ProjE using experimental procedures, datasets, and metrics established in the related work. We also give the results of TransE [8] and TransH [17] implemented by [18] on our datasets.

4.1. Experimental Setting

4.1.1. Dataset. For evaluating our proposed model, we use FB15K and FB15K-237 datasets to conduct experiments.

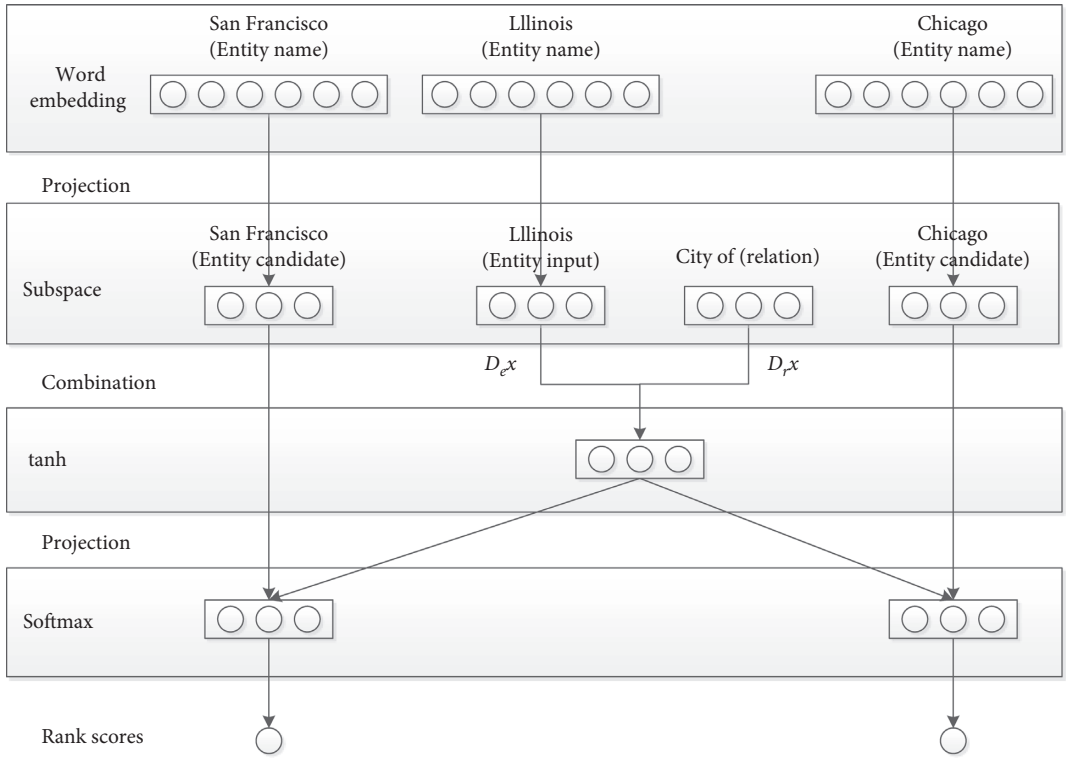


Figure 1: Illustration of the sub-ProjE model on entity prediction task.

The FB15K dataset released by [8] is a subset of Freebase [1], which contains 592,213 triples and involves 14,951 entities and 1345 relations. The initial entity vector is indicated with the entity name vector unsupervised pretrained by word2vec [9] with a large-scale web corpus. The dimension of the initial entity vector is 1000. In order to ensure each entity has a pretrained initial vector representation, we deleted 1423 entities without entity name vectors from FB15K and removed triples about these entities accordingly. The finally retained training set includes 364,424 triples, validation set includes 37,905 triples and the test set includes 44,565 triples, which totally involves 13,528 entities and 1345 relations.

FB15K-237 introduced by Toutanova and Chen [35] is a subset of FB15K where inverse relations are removed. We remove entities without entity name vectors too, the same as FB15K. The finally resulted FB15K-237 dataset contains 211,380 training triples, 21,981 validation triples, and 25,666 test triples, which totally involves 13,528 entities and 237 relations.

For zero-shot scenario, we divide entities into two groups: training entities (10,000) and test entities (3,528), while ensuring the training set and validation set only contains triples whose head entity and tail entity are both in training entity group, and the triple in test set has at least one entity in a test entity group. Finally, the FB15K dataset for zero-shot scenario has 201,272 training triples, 20,968 validation triples, and 3,012 test triples, respectively. The FB15K-237 dataset for zero-shot scenario has 117,785 training triples, 12,196 validation triples, and 1,762 test triples, respectively. Table 1 shows the statistical properties of datasets.

4.1.2. Parameter Setting. In the supervised training phase, we apply the default setting the same as ProjE: using Adam [36] as the stochastic optimizer with hyperparameter settings of $\beta_1 = 0.9$, $\beta_1 = 0.999$, $\varepsilon = 1e^{-8}$; L_1 regularized to all parameters during the training and dropout layer on top of the combination operator to avoid overfitting. Other parameters are set as follows: learning rate $l_r = 0.01$, batch size $b = 200$, regularized parameter $\alpha = 1e^{-5}$, dropout probability $p_d = 0.5$, negative sample sampling probability $p_y = 0.5$.

4.2. Experimental Results

4.2.1. Evaluation Protocol. In the entity prediction tasks, we predict the missing head entity or tail entity in the triples by ranking all of the entities in the knowledge graph. Given a test triple (h, r, t), we remove the head or tail entity and then replace it with each entity in the knowledge graph and calculate the ranking score, then rank these replacing entities in descending order and record the ranking of the right entities. Following [8], we use mean rank, HITS@k, filtered mean rank, and filtered HITS@k as our evaluation metrics.

4.2.2. Results. Table 2 shows the results of different models in entity prediction tasks trained on FB15K with different training set sizes. We can see from Table 2 that the Mean Rank of sub-ProjE is dramatically superior to that of ProjE, TansH, and TransE when the training data becomes less. Both sub_ProjE and ProjE outperform TransE and TransH.

Input: Training triples $T = \{(h, r, t)\}$, entities $\mathbf{E}$, initial entity matrix $\mathbf{W}^{\mathbf{E}}$, relations R, embedding subspace dimension s, dropout probability p_d, candidate sampling rate p_y, regularize parameter α
Output: subspace adaptive matrix $\mathbf{W}^{\mathbf{s}}$, relation embedding matrix WR, combination operators $\mathbf{D}_{eh}$, $\mathbf{D}_{rh}$, $\mathbf{D}_{et}$, $\mathbf{D}_{rt}$
Algorithm sub-ProjE ($\mathbf{T}$, $\mathbf{E}$, $\mathbf{W}^{\mathbf{E}}$, $\mathbf{R}$, s, p_d, p_y, α)
(1) Initializing adaptive matrix $\mathbf{W}^{\mathbf{s}}$, relation matrix $\mathbf{W}^{\mathbf{R}}$, combination operators (diagonal matrices) $\mathbf{D}_{eh}$, $\mathbf{D}_{rh}$, $\mathbf{D}_{et}$, $\mathbf{D}_{rt}$ with uniform distribution $(-6/\sqrt{s}, 6/\sqrt{s})$.
(2) **Loop**/*A training iteration/epoch*/
(3) $\mathbf{T}^h \leftarrow$, $\mathbf{C}^h \leftarrow$, $\mathbf{T}^t \leftarrow$, $\mathbf{C}^t \leftarrow$;//training data
(4) **for** $(h, r, t) \in \mathbf{T}$ do/*construct training data using all train triples*/
(5) $e \leftarrow \text{random}(h, t)$
(6) **if** $e == h$ **then**/*tail is missing*/
(7) $\mathbf{T}^h.\text{add}([e, r])$
(8) $\mathbf{C}^h.\text{add}(\{t' \mid (h, r, t') \in \mathbf{T}\} \cup \text{sample}(\mathbf{E}, \mathbf{p}_\mathbf{y}))$/*all positive tails from T and some sampled negative candidates*/
(9) **else**/*head is missing*/
(10) $\mathbf{T}^t.\text{add}([e, r])$
(11) $\mathbf{C}^t.\text{add}(\{h' \mid (h', r, t) \in \mathbf{T}\} \cup \text{sample}(\mathbf{E}, \mathbf{p}_\mathbf{y}))$/*all positive heads from T and some sampled negative candidates*/
(12) **end if**
(13) **end for**
(14) **for each** $(\mathbf{T}_b^h, \mathbf{C}_b^h, \mathbf{T}_b^t, \mathbf{C}_b^t) \subset (\mathbf{T}^h, \mathbf{C}^h, \mathbf{T}^t, \mathbf{C}^t)$ **do**/*minibatches*/
(15) $l \leftarrow 0$
(16) **for each** $(\mathbf{t}^h, \mathbf{c}^h, \mathbf{t}^t, \mathbf{c}^t) \in (\mathbf{T}_b^h, \mathbf{C}_b^h, \mathbf{T}_b^t, \mathbf{C}_b^t)$ do/*training instance*/
(17) $O_h \leftarrow \text{softmax}((\mathbf{W}^s \times \mathbf{W}^{\mathbf{E}})_{[\mathbf{c}^t,:]} \times \tanh(\text{dropout}(p_d, \mathbf{D}_{et} \times ((\mathbf{W}^s \times \mathbf{W}^{\mathbf{E}})_{[\mathbf{t}^t[0],:]})^T + \mathbf{D}_{rt} \times (\mathbf{W}^{\mathbf{R}}_{[\mathbf{t}^t[1],:]})^T + b_c)) + b_p)$
(18) $O_t \leftarrow \text{softmax}((\mathbf{W}^s \times \mathbf{W}^{\mathbf{E}})_{[\mathbf{c}^h,:]} \times \tanh(\text{dropout}(p_d, \mathbf{D}_{eh} \times ((\mathbf{W}^s \times \mathbf{W}^{\mathbf{E}})_{[\mathbf{t}^h[0],:]})^T + \mathbf{D}_{rh} \times (\mathbf{W}^{\mathbf{R}}_{[\mathbf{t}^h[1],:]})^T + b_c)) + b_p)$
(19) $l = l - \sum(\{1((h, \mathbf{t}^t[1], \mathbf{t}^t[0]) \in \mathbf{T}) \mid h \in \mathbf{c}^t\} \circ \log(\mathrm{O_h})) - \sum(\{1((\mathbf{t}^h[0], \mathbf{t}^h[1], t) \in \mathbf{T}) \mid t \in \mathbf{c}^h\} \circ \log(\mathrm{O_h}))$
(20) **end for**
(21) $l_r \leftarrow \text{Regu}_1(\mathbf{W}^s) + \text{Regu}_1(\mathbf{W}^{\mathbf{R}}) + \text{Regu}_1(\mathbf{D}_{eh}) + \text{Regu}_1(\mathbf{D}_{rh}) + \text{Regu}_1(\mathbf{D}_{et}) + \text{Regu}_1(\mathbf{D}_{rt})$
(22) Update all parameters w.r.t $l + \alpha l_r$
(23) **end for**
(24) **EndLoop**

Algorithm 1: Embedding knowledge graphs into subspace.

Table 1: Statistical properties of the dataset.

Dataset	Relation	Entity	Training set	Validation set	Test set
FB15K	1345	13528	364424	37905	44565
FB15K-zero-shot	1345	10000 (train)/3528 (test)	201272	20968	3012
FB15K-237	237	13528	211380	21981	25666
FB15K-237-zero-shot	237	10000 (train)/3528 (test)	117785	12196	1762

Since the training data is less, the filtered Mean Rank is similar to the original Mean Rank. When the training set is very large, the performances of ProjE, TransE, and TransH are superior to that of sub-ProjE, which verifies the fact that sub-ProjE is applicable to less training data scenario. HITs@10 of ProjE is superior to sub-ProjE. This is because ProjE only partially updates the entity vectors during training. Therefore, partially the accuracy of ProjE is higher than that of sub-ProjE, but on the whole, the performance of sub-ProjE is superior to that of ProjE. Table 3 shows the results of different models in entity prediction tasks trained on FB15K-237 with different training set sizes. The results are similar to the results of FB15K.

In the zero-shot scenario, at least one entity of a tested triple is not in the knowledge graph (i.e., the training set). The original ProjE method cannot deal with this situation since it cannot generate entity representation that is not in the knowledge graph. TransE and TransH cannot apply to zero-shot scenario too for the same reason. Our sub-ProjE method is capable of dealing with this situation because it indirectly updates the missing entity representation during training. Table 4 shows the results of zero-shot scenario experiments on FB15K and FB15K237. We can see from Table 4 that sub-ProjE can deal with the entity prediction in zero-shot scenario even when the training data is very limited.

Table 2: Prediction results with different training set sizes on FB15K.

		Mean rank		HITs@10 (%)	
Training set size	Method	Raw	Filtered	Raw	Filtered
Train = 5000	TransE	5042	5030	10.4	10.7
	TransH	5125	5113	10.2	10.5
	ProjE	1340	1328	24.3	32.8
	Sub-ProjE	534	522	19.2	19.9
Train = 10000	TransE	3372	3358	16.0	16.6
	TransH	3388	3375	16.2	16.8
	ProjE	717	705	32.9	42.8
	Sub-ProjE	408	395	22.8	23.7
Train = all	TransE	165	98	53.7	70.9
	TransH	166	100	58.0	70.3
	ProjE	157	98	42.5	66.1
	Sub-ProjE	326	266	24.8	34.3

Table 3: Prediction results with different training set sizes on FB15K-237.

		Mean rank		HITs@10 (%)	
Training set size	Method	Raw	Filtered	Raw	Filtered
Train = 5000	TransE	4559	4544	13.6	14.0
	TransH	4568	4554	13.4	13.9
	ProjE	1155	1141	29.6	41.6
	Sub-ProjE	447	434	19.5	20.3
Train = 10000	TransE	2902	2886	18.9	19.9
	TransH	2972	2955	18.5	19.5
	ProjE	614	599	38.2	52.2
	Sub-ProjE	377	362	23.7	24.9
Train = all	TransE	297	219	42.9	58.3
	TransH	293	217	42.5	58.0
	ProjE	164	98	40.8	64.9
	Sub-ProjE	313	246	25.7	35.3

Table 4: Zero-shot scenario results of sub-ProjE with different training set sizes.

		Mean rank		HITs@10 (%)	
Training set size	Dataset	Raw	Filtered	Raw	Filtered
Train = 5000	FB15K-237	767	769	12.2	12.2
	FB15K	748	746	12.4	12.5
Train = 10000	FB15K-237	699	698	13.0	13.1
	FB15K	669	667	14.4	14.4
Train = all	FB15K-237	524	523	16.3	16.4
	FB15K	531	529	17.4	17.5

With the increase of training data, the performance of the method is further improved.

In order to further analyze the stability of sub-ProjE model, we give the mean rank and HITs@10 results of the first 30 iterations on FB15K in a common scenario and zero-shot scenario, as Figures 2 and 3 show. We can see from Figure 2 that the larger the training dataset, the faster the sub-ProjE model converges. The performance of sub-ProjE model becomes stable after the first several iterations when all the training dataset is put into training. When the data is less, sub-ProjE converges slower and becomes stable after over ten times of iteration. The convergent speed of the zero-shot scenario is the same as the general scenario. We get similar results on FB15K-237. This shows sub-ProjE model is applicable to the zero-shot scenario.

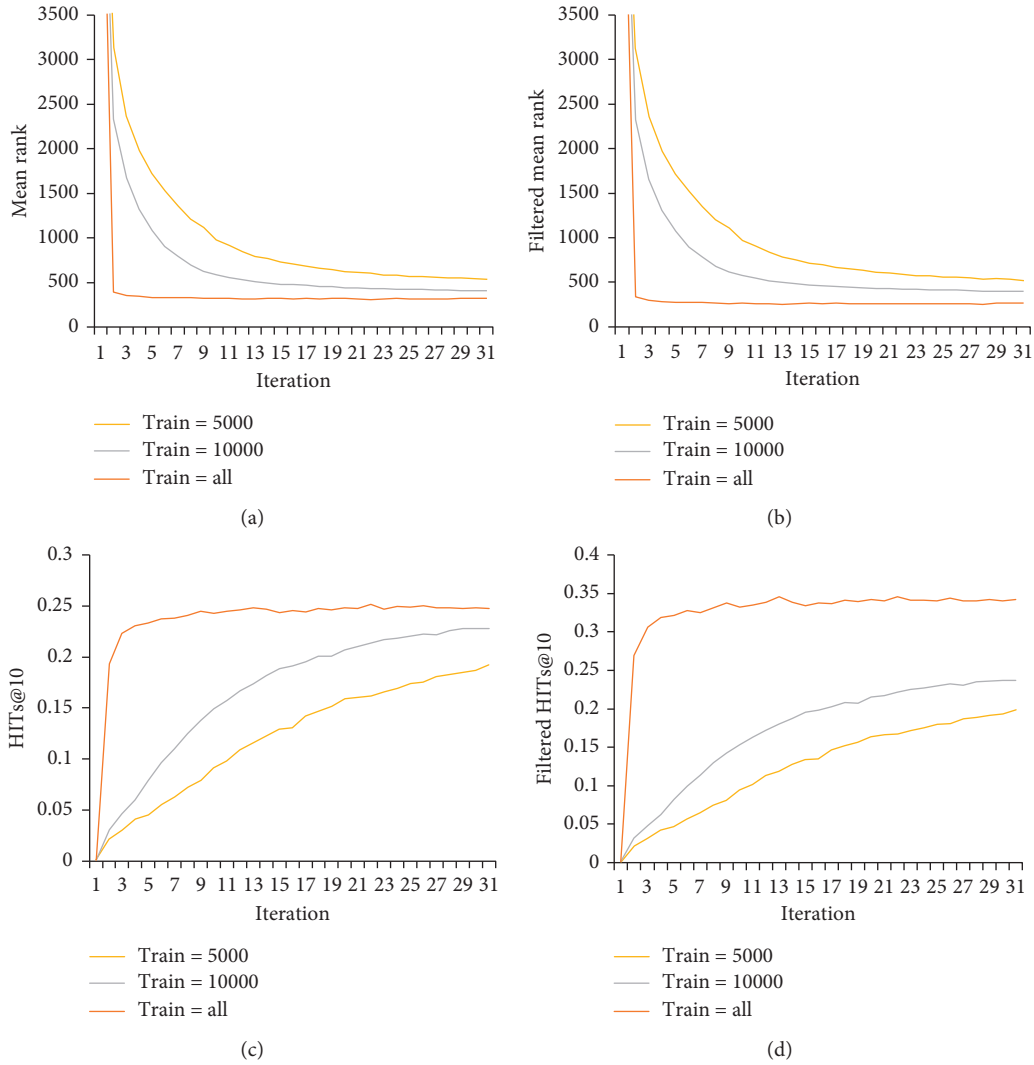
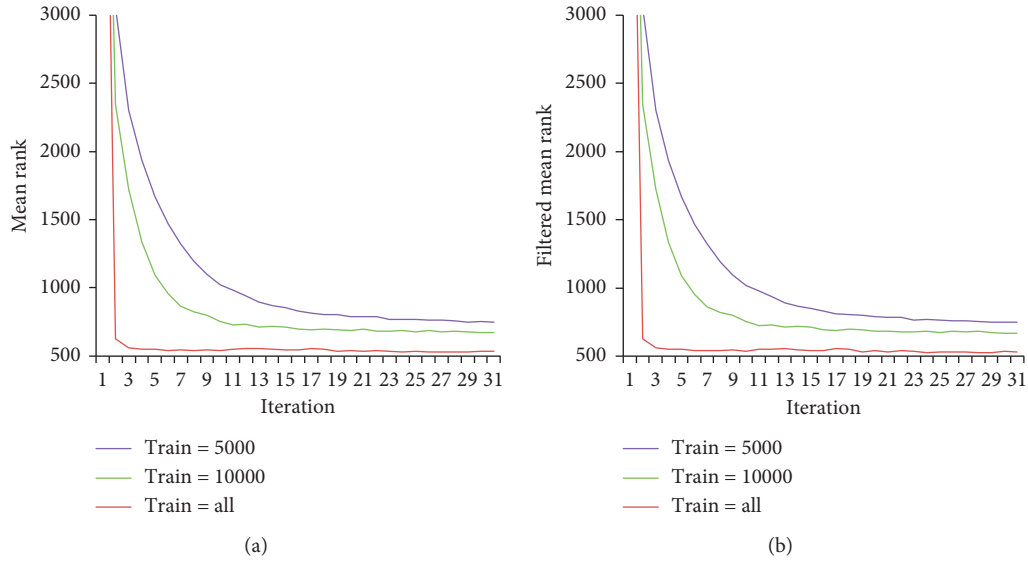


Figure 2: Training process of sub-ProjE model in general scenario. (a) Mean Rank. (b) Filtered Mean Rank. (c) HITs@10. (d) Filtered HITs@10.



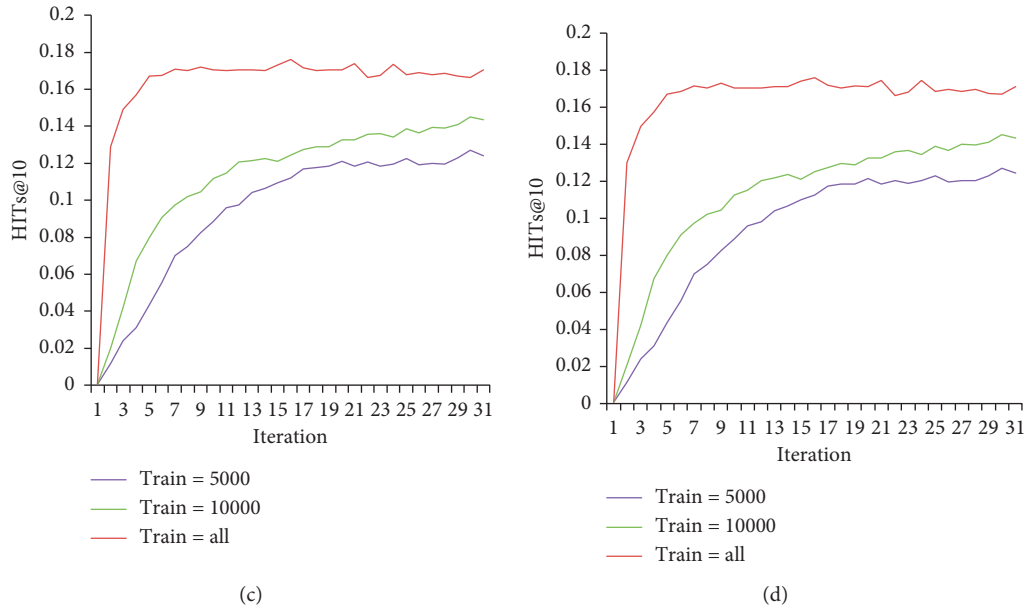


Figure 3: Training process of sub-ProjE model in zero-shot scenario. (a) Mean Rank. (b) Filtered Mean Rank. (c) HITs@10. (d) Filtered HITs@10.

5. Conclusions

This paper proposes a new knowledge representation learning method utilizing unsupervised entity name vectors. The basic idea is to seek the subspace projection of unsupervised entity vectors in knowledge representation tasks. This method allows indirect update of entity vectors that do not appear during the process of training and applicable to the case that only a few labelled data can be obtained. Experiments on Freebase verify the effectiveness of this method. Results show that the performance of this simple method surpasses the best existing knowledge representation learning model in case the training data is less, and furthermore, it can be applied to zero-shot scenarios.

Data Availability

The datasets used in this paper to produce the experimental results are publicly available. FB15k and FB15k-237 can be downloaded from http://openke.thunlp.org. The unsupervised pretrained entity vectors can be downloaded from http://code.google.com/p/word2vec.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant no. 61876217, Suzhou Science and Technology Plan Project under Grant no. SYG201903, Fundamental Computing Education Research Project of Association of Fundamental Computing Education in Chinese Universities under Grant nos. 2018-AFCEC-328 and 2019-AFCEC-288, and Research Funds of Suzhou Vocational Institute of Industrial Technology under Grant no. 2019kyqd001.

References

[1] K. Bollacker, C. Evans, P. Paritosh et al., "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pp. 1247–1250, Vancouver, Canada, June 2008.

[2] G. A. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[3] C. Unger, L. Bühmann, J. Lehmann et al., "Template-based question answering over RDF data," in *Proceedings of the 21st International Conference on World Wide Web*, pp. 639–648, Lyon, France, April 2012.

[4] M. Nickel, K. Murphy, V. Tresp et al., "A review of relational machine learning for knowledge graphs," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 11–33, 2015.

[5] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[6] A. García-Durán, A. Bordes, N. Usunier, and Y. Grandvalet, "Combining two and three-way embedding models for link prediction in knowledge bases," *Journal of Artificial Intelligence Research*, vol. 55, pp. 715–742, 2016.

[7] R. Socher, D. Chen, C. D. Manning et al., "Reasoning with neural tensor networks for knowledge base completion," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 926–934, Lake Tahoe, NV, USA, December 2013.

[8] A. Bordes, N. Usunier, A. Garcia-Duran et al., "Translating embeddings for modeling multi-relational data," in *Proceedings of the Advances in Neural Information Processing*

Systems, pp. 2787–2795, Lake Tahoe, NV, USA, December 2013.

[9] T. Mikolov, I. Sutskever, K. Chen et al., "Distributed representations of words and phrases and their compositionality," in *Proceedings of the Advances in Neural Information Processing Sytems*, pp. 3111–3119, Lake Tahoe, NV, USA, December 2013.

[10] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 650–658, Las Vegas, NV, USA, August 2008.

[11] M. Nickel, V. Tresp, and H. P. Kriegel, "A three-way model for collective learning on multi-relational data," *ICML*, vol. 11, pp. 809–816, 2011.

[12] M. Nickel, V. Tresp, and H. P. Kriegel, "Factorizing yago: scalable machine learning for linked data," in *Proceedings of the 21st International Conference on World Wide Web*, pp. 271–280, Lyon, France, April 2012.

[13] R. Jenatton, N. L. Roux, A. Bordes et al., "A latent factor model for highly multi-relational data," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 3167–3175, San Francisco, CA, USA, Decemeber 2012.

[14] A. Bordes, J. Weston, R. Collobert et al., "Learning structured embeddings of knowledge bases," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, August 2011.

[15] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," *Machine Learning*, vol. 94, no. 2, pp. 233–259, 2014.

[16] D. Chen, R. Socher, C. D. Manning et al., "Learning new facts from knowledge bases with neural tensor networks and semantic word vectors," 2013, http://arxiv.org/abs/1301.3618.

[17] Z. Wang, J. Zhang, J. Feng et al., "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, Québec, Canada, July 2014.

[18] Y. Lin, Z. Liu, M. Sun et al., "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the AAAI*, pp. 2181–2187, Austin, TX, USA, January 2015.

[19] G. Ji, S. He, L. Xu et al., "Knowledge graph embedding via dynamic mapping matrix," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 1, pp. 687–696, Beijing, China, July 2015.

[20] Y. Jia, Y. Wang, H. Lin et al., "Locally adaptive translation for knowledge graph embedding," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, AZ, USA, February 2016.

[21] Y. Lin, Z. Liu, H. Luan et al., "Modeling relation paths for representation learning of knowledge bases," 2015, http://arxiv.org/abs/1506.00379.

[22] G. Ji, K. Liu, S. He et al., "Knowledge graph completion with adaptive sparse transfer matrix," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, AZ, USA, February 2016.

[23] B. Shi and T. Weninger, "ProjE: embedding projection for knowledge graph completion," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, February 2017.

[24] Y. Luo, Q. Wang, B. Wang et al., "Context-dependent knowledge graph embedding," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1656–1661, Lisbon, Portugal, September 2015.

[25] P. Ristoski and H. Paulheim, "RDF2Vec: RDF graph embeddings for data mining," *Lecture Notes in Computer Science*, Springer, Berlin, Germany, pp. 498–514, 2016.

[26] Y. Dong, N. V. Chawla, A. Swami et al., "Metapath2vec: Scalable Representation Learning for Heterogeneous Networks," in *Proceedings of the Knowledge Discovery and Data Mining*, pp. 135–144, Jeju, South Korea, May 2017.

[27] R. Hussein, D. Yang, P. Cudremauroux et al., "Are meta-paths necessary?: revisiting heterogeneous graph embeddings," in *Proceedings of the Conference on Information and Knowledge Management*, pp. 437–446, Torino, Italy, October 2018.

[28] T. Dettmers, P. Minervini, P. Stenetorp et al., "Convolutional 2d knowledge graph embeddings," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, February 2018.

[29] Z. Wang, J. Zhang, J. Feng et al., "Knowledge graph and text jointly embedding," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1591–1601, Doha, Qatar, October 2014.

[30] R. Xie, Z. Liu, J. Jia et al., "Representation learning of knowledge graphs with entity descriptions," in *Proceedings of the AAAI*, pp. 2659–2665, Phoenix, AZ, USA, February 2016.

[31] H. Xiao, M. Huang, and X. Zhu, "SSP: semantic space projection for knowledge graph embedding with text descriptions," 2016, http://arxiv.org/abs/1604.04835.

[32] Y. Bengio, R. Ducharme, P. Vincent et al., "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[33] R. Collobert, J. Weston, L. Bottou et al., "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[34] T. Mikolov, K. Chen, G. S. Corrado et al., "Efficient estimation of word representations in vector space," in *Proceedings of the International conference on learning representations*, Scottsdale, AZ, USA, May 2013.

[35] K. Toutanova and D. Chen, "Observed versus latent features for knowledge base and text inference," in *Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality*, pp. 57–66, Beijing, China, July 2015.

[36] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proceedings of the International conference on learning representations*, San Diego, CA, USA, May 2015.

Hindawi
Scientific Programming
Volume 2020, Article ID 7081653, 15 pages
https://doi.org/10.1155/2020/7081653

Research Article

Multiswarm Multiobjective Particle Swarm Optimization with Simulated Annealing for Extracting Multiple Tests

Toan Bui,[1] Tram Nguyen,[2,3] Huy M. Huynh,[4] Bay Vo,[1] Jerry Chun-Wei Lin,[5] and Tzung-Pei Hong[6,7]

[1]*Faculty of Information Technology, Ho Chi Minh City University of Technology (HUTECH), Ho Chi Minh, Vietnam*
[2]*Faculty of Information Technology, Nong Lam University, Ho Chi Minh, Vietnam*
[3]*Department of Computer Science, Faculty of Electrical Engineering and Computer Science, VŠB-Technical University of Ostrava, Ostrava, Czech Republic*
[4]*Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam*
[5]*Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway*
[6]*Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, Taiwan*
[7]*Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan*

Correspondence should be addressed to Bay Vo; vd.bay@hutech.edu.vn

Received 5 February 2020; Revised 8 May 2020; Accepted 8 July 2020; Published 1 August 2020

Academic Editor: Kifayat Ullah Khan

Education is mandatory, and much research has been invested in this sector. An important aspect of education is how to evaluate the learners' progress. Multiple-choice tests are widely used for this purpose. The tests for learners in the same exam should come in equal difficulties for fair judgment. Thus, this requirement leads to the problem of generating tests with equal difficulties, which is also known as the specific case of generating tests with a single objective. However, in practice, multiple requirements (objectives) are enforced while making tests. For example, teachers may require the generated tests to have the same difficulty and the same test duration. In this paper, we propose the use of Multiswarm Multiobjective Particle Swarm Optimization (MMPSO) for generating k tests with multiple objectives in a single run. Additionally, we also incorporate Simulated Annealing (SA) to improve the diversity of tests and the accuracy of solutions. The experimental results with various criteria show that our approaches are effective and efficient for the problem of generating multiple tests.

1. Introduction

In the education sector, evaluation of students' study progress is important and mandatory. There are many methods such as oral tests or writing tests to evaluate their knowledge and understanding about subjects. Because of the scalability and ease of human resources, writing tests are used more widely for the final checkpoints of assessment (e.g., final term tests), where a large number of students must be considered. Writing tests can be either descriptive tests, in which students have to fully write their answers, or multiple-choice tests, in which students pick one or more choices for each question. Even though descriptive tests are easier to create at first, they consume a great deal of time and effort during the grading stage. Multiple-choice tests, on the other hand, are harder to create at first as they require a large number of questions for security reasons, as in Ting et al. [1]. However, the grading process can be extremely fast, automated by computers, and bias-free from human graders. Recently, many researchers have invested their efforts to make computers automate the process of creating multiple-choice tests using available question banks, as in the work of Cheng et al. [2]. The results were shown to be promising and, thus, make multiple-choice tests more feasible for examinations.

One of the challenges in generating multiple-choice tests is the difficulty of the candidate tests. The tests for all students should have the same difficulty for fairness. However, it can be

seen that generating all tests having the same level of difficulties is an extremely hard task, even in the case of manually choosing questions from a question bank, and the success rate of generating multichoice tests satisfying a given difficulty is low and time-consuming. Therefore, to speed up the process, some authors chose to automatically generate tests with the use of computers and approximate the difficulties of the required difficulties. This is also known as generating tests with a single objective where the level of difficulty is the objective. For example, Bui et al. [3] proposed the use of particle swarm optimization to generate tests with approximating difficulties to the required levels from users. The tests are generated from question banks that consist of various questions with different difficulties. The difficulty value of each question is judged and adapted based on users via previous real-life exams. The work evaluates three random oriented approaches, which are Genetic Algorithms (GAs) by Yildirim [4, 5] and Particle Swarm Optimization (PSO). The experiment result shows that PSO gives the best performance concerning most of the criteria by Bui et al. [3]. Previous works only focused on solving a single objective of the extracting test based on the difficulty level requirement of the user defined. In practice, exam tests can depend on multiple factors such as questions' duration and total testing time. Thus, designing a method that can generate tests with multiple objectives is challenging. Furthermore, the proposed approaches can only extract a single test at each run. To extract multiple tests, the authors have to execute their application multiple times. This method is time-consuming, and duplicate tests can occur because each run is executed separately. Besides, they do not have any information about each other to avoid duplication.

In this paper, we propose a new approach that uses Multiswarm Multiobjective Particle Swarm Optimization (MMPSO) to extract k tests in a single run with multiple objectives. Multiswarms are the same as the multitest in extracting k tests. However, they are based on their search on multiple subswarms instead of one standard swarm that executes their application multiple times to extract multiple tests. The use of diverse subswarms to increase performance when optimizing their tests is studied in Antonio and Chen [6]. Additionally, we use Simulated Annealing (SA) to initialize the first population for PSO to increase the diversities of generated tests. We also aim to improve the results on various criteria such as diversities of solutions and accuracy.

The main contributions of this paper are as follows:

(1) We propose a multiswarm multiobjective approach to deal with the problem of extracting k tests simultaneously.

(2) We propose the use of SA in combining with PSO for extracting tests. SA was selected as it is capable of escaping local optimum solutions.

(3) We propose a parallel version of our serial algorithms. Using parallelism, we can control the overlap of extracted tests to save time.

The rest of this paper is organized as follows. Section 2 describes the related research. The problem of extracting k tests from question banks is explained in Section 3. Correlated studies of normal multiswarm multiobjective PSO and multiswarm multiobjective PSO with SA for the problem of extracting k tests from question banks are presented in Sections 4 and 5. The next section analyzes and discusses the experimental results of this study. Finally, the future research trends, and the conclusions of the paper are provided in Section 6.

2. Related Work

Recently, evolutionary algorithms have been applied to many fields for optimization problems. Some of the most well-known algorithms are Genetic Algorithms (GAs) and Particle Swarm Optimization (PSO). GAs were invented based on the concept of Darwin's theory of evolution, and they seek solutions via progressions of generations. Heuristic information is used for navigating the search space for potential individuals, and this can achieve globally optimal solutions. Since then, there have been many works that used GAs in practice [7–11].

Particle swarm optimization is a swarm-based technique for optimization that is developed by Eberthart and Kennedy [12]. It imitates the behavior of a school of fishes or the flock of birds. PSO optimizes the solutions via the movements of individuals. The foundation of PSO's method of finding optima is based on the following principles proposed by Eberhart and Kennedy: (1) All individuals (particles) in swarms tend to find and move towards possible attractors; (2) each individual remembers the position of the best attractor that it found. In particular, each solution is a particle in a swarm and is denoted by two variables. One is the current location, denoted by present[], and the other is the particle's velocity, denoted by v[]. They are two vectors on the vector space R^n, in which n changes based on the problems. Additionally, each particle has a fitness value that is given by a chosen fitness function. At the beginning of the algorithm, the initial generation (population) is created either in a random manner or by some methods. The movement of each particle individual is affected by two information sources. The first is P_{best}, which is the best-known position of the particle visited in the past movements. The second is G_{best}, which is the best-known position of the whole swarm. In the original work proposed by Eberhart and Kennedy, particles traverse the search space by going after the particles with strong fitness values. Particularly, after disjointed periods, the velocity and position of each individual are updated with the following formulas:

$$\begin{aligned} v^{t+1} &= v^t + c1.\text{rand} * P^t_{\text{best}} - \text{present}^t \\ &\quad + c2.\text{rand} * G^t_{\text{best}} - \text{present}^t, \\ \text{present}^{t+1} &= \text{persent}^t + v^{t+1} \end{aligned} \tag{1}$$

where rand() is a function that returns a random number in the range (0,1) and $c1$, $c2$ are constant weights.

While PSO is mostly used for the continuous value domain, recently, some works have shown that PSO can also be prominently useful for discrete optimization. For

example, Sen and Krishnamoorthy [13, 14] transformed the original PSO into discrete PSO for solving the problem of transmitting information on networks. The work result proves that the proposed discrete PSO outperforms Simulated Annealing (SA).

To further improve the performance for real-life applications, some variants of PSO have been proposed and exploited such as multiswarm PSO. Peng et al. [15] proposed an approach for multiswarm PSO that pairs the velocity update of some swarms with different methods such as the periodically stochastic learning strategy or random mutation learning strategy. The experiments have been run on a set of specific benchmarks. The results showed that the proposed method gives a better quality of solutions and has a higher chance of giving correct solutions than normal PSO. Vafashoar and Meibodi [16] proposed an approach that uses Cellular Learning Automata (CLA) for multiswarm PSO. Each swarm is placed on a cell of the CLA, and each particle's velocity is affected by some other particles. The connected particles are adjusted overtime via periods of learning. The results indicate that the proposed method is quite effective for optimization problems on various datasets. In order to balance the search capabilities between swarms, Xia et al. [17] used multiswarm PSO in combination with various strategies such as the dynamic subswarm number, subswarm regrouping, and purposeful detecting. Nakisa et al. [18] proposed a strategy to improve the speed of convergence of multiswarm PSO for robots' movements in a complex environment with obstacles. Additionally, the authors combine the local search strategy with multiswarm PSO to prevent the robots from converging at the same locations when they try to get to their targets.

In practice, there exist a lot of optimization problems with multiple objectives instead of a single objective. Thus, a lot of work for multiobjective optimization has been proposed. For example, Li and Babak [19] proposed multiobjective PSO combining with an enhanced local search ability and parameter-less sharing. Kapse and Krishnapillai [20] also proposed an adaptive local search method for multiobjective PSO using the time variance search space index to improve the diversity of solutions and convergence. Based on crowded distance sorting, Cheng et al. [21] proposed an improved version, circular crowded sorting, and combined with multiobjective PSO. The approach scatters the individuals of initial populations across the search space in order to be better at gathering the Pareto frontier. The method was proven to improve the search capabilities, the speed of convergence, and diversity of solutions. Similarly, Adel et al. [22] used multiobjective with uniform design instead of traditional random methods to generate the initial population. Based on $R2$ measurement, Alan et al. [23] proposed an approach that used $R2$ as an indicator to navigate swarms through the search space in multiobjective PSO. By combining utopia point-guided search with multiobjective PSO, Kapse and Krishnapillai [24] proposed a strategy that selects the best individuals that are located near the utopia points. The authors also compared their method with other algorithms such as NSGA-II (Nondominated Sorting Genetic Algorithm II) by Deb et al. [25] or CMPSO (Coevolutionary multiswarm PSO) by Zhan et al. [26] on several benchmarks and demonstrated the proposed method's effectiveness. Saxena and Mishra [27] designed a multiobjective PSO algorithm named MOPSO tridist. The algorithm used triangular distance to select leader individuals which cover different regions in Pareto frontier. The authors also included an update strategy for Pbest with respect to their connected leaders. MOPSO tridist was shown to outperform other multiobjective PSOs, and the authors illustrated the algorithm's application with the digital watermarking problem for RBG images. Based on chaotic particle swarm optimization, Liansong and Dazhi [28] designed a multiobjective optimization for chaotic particle swarm optimization and based on comprehensive learning particle swarm optimization, and Xiang and Xueqing [29] proposed an extension, the MSCLPSO algorithm, and incorporated various techniques from other evolutionary algorithms. In order to increase the flexibility of multiobjective PSO, Mokarram and Banan [30] proposed the FC-MOPSO algorithm that can work on a mix-up of constrained, unconstrained, continuous, and/or discrete optimization problems. Recently, Mohamad et al. [31] reviewed and summarized the disadvantages of multiobjective PSO. Based on that, they proposed an algorithm, M-MOPSO. The authors also proposed a strategy based on dynamic search boundaries to help escape the local optima. M-MOPSO was proven to be more efficient when compared with several state-of-the-art algorithms such as Multiobjective Grey Wolf Optimizer (MOGWO), Multiobjective Evolutionary Algorithm based on Decompositions (MOEA/D), and Multiobjective Differential Evolution (MODE).

An extension of multiple objective optimization problems is the dynamic multiple objective optimization problems, in which each objective would change differently depending on the time or environment. To deal with this problem, Liu et al. [32] proposed CMPSODMO which is based on the multiswarm coevolution strategy. The author also combined it with special boundary constraint processing and a velocity update strategy to help with the diversity and convergence speed.

To make it easier for readers, Table 1 summarizes different application domains in which PSO algorithms have been applied for different purposes.

The abovementioned works can be effective and efficient for the optimization problems in Table 1; however, applying them for the problem of generating k test in a single run with multiple objectives is not feasible according to the work of Nguyen et al. [33]. Therefore, in this work, we propose an approach that uses Multiswarm Multiobjective Particle Swarm Optimization (MMPSO) combined with Simulated Annealing (SA) for generating k tests with multiple objectives. Each swarm, in this case, is a test candidate, and it runs on a separate thread. The migration happens randomly by chance. We also aim to improve the accuracy and diversity of solutions.

Table 1: Summarizes different application domains in which PSO algorithms have been applied for different purposes.

Categories	References	Algorithm	Types of optimization problems								Modern optimization techniques
			Unconstrained	Constrained	Continuous	Discrete	Single-objective	Multiobjective	Single-swarm	Multiswarm	
Academia and scientometrics	[22]	Particle swarm optimization Based on multiobjective functions with uniform design (MOPSO-UD)		X	X			X			Multiobjective PSO with uniform design generates the initial population instead of traditional random methods
	[23]	*R*2-based multi/many-objective particle swarm optimization		X	X			X			The proposed approach used *R*2 as an indicator to navigate swarms through the search space in multiobjective PSO
	[21]	An improved version, circular crowded sorting, and combined with multiobjective PSO		X	X			X			The individuals of initial populations across the search space to be better at gathering the Pareto frontier
	[20]	An adaptive local search method for multiobjective PSO		X	X			X			An adaptive local search method for multiobjective PSO using the time variance search space index to improve the diversity of solutions and convergence
	[24]	Combining utopia point-guided search with multiobjective PSO		X	X			X			A strategy that selects the best individuals that are located near the utopia points
	[19]	A novel MOPSO with enhanced local search ability and parameter-less sharing		X	X			X			The proposed approach estimates the density of the particles' neighborhood in the search space. Initially, the proposed method accurately determines the crowding factor of the solutions; in later stages, it effectively guides the entire swarm to converge close to the true Pareto front
	[28]	Chaotic particle swarm optimization		X	X			X			The work improves the diversity of the population and uses simplified mesh reduction and gene exchange to improve the performance of the algorithm
	[32]	A coevolutionary technique based on multiswarm particle swarm optimization		X	X			X	X		The authors combined their proposed algorithm with special boundary constraint processing and a velocity update strategy to help with the diversity and convergence speed
	[31]	Particle swarm optimization algorithm based on dynamic boundary search for constrained optimization		X	X			X			The authors proposed a strategy based on dynamic search boundaries to help escape the local optima
	[30]	A new PSO-based algorithm (FC-MOPSO)	X	X	X	X	X	X			FC-MOPSO algorithm can work on a mix-up of constrained, unconstrained, continuous and/or discrete, single-objective, multiobjective optimization problems algorithm that can work on a mix-up of constrained, unconstrained, continuous, and/or discrete optimization problems
	[15]	A novel particle swarm optimization algorithm with multiple learning strategies (PSO-MLS)		X	X		X			X	The authors proposed an approach for multiswarm PSO that pairs the velocity update of some swarms with different methods such as the periodically stochastic learning strategy or random mutation learning strategy.
	[16]	Cellular Learning Automata (CLA) for multiswarm PSO		X	X		X			X	Each swarm is placed on a cell of the CLA, and each particle's velocity is affected by some other particles. The connected particles are adjusted overtime via periods of learning
	[17]	Improved particle swarm optimization algorithm based on dynamical topology and purposeful detecting.		X	X		X				In order to balance the search capabilities between swarms. The extensive experimental results illustrate the effectiveness and efficiency of the three proposed strategies used in MSPSO
	[29]	Particle swarm optimization with differential evolution (DE) strategy		X	X			X		X	The purpose is to achieve high-performance multiobjective optimization
	[26]	Coevolutionary multiswarm PSO		X	X		X				To modify the velocity update equation to increase search information and diversity solutions to avoid local Pareto front. The results show superior performance in solving optimization problems

Table 1: Continued.

Categories	References	Algorithm	Types of optimization problems								Modern optimization techniques
			Unconstrained	Constrained	Continuous	Discrete	Single-objective	Multiobjective	Single-swarm	Multiswarm	
	[18]	Particle swarm optimization with local search		X		X	X			X	The authors proposed a strategy to improve the speed of convergence of multiswarm PSO for robots' movements in a complex environment with obstacles. Additionally, the authors combine the local search strategy with multiswarm PSO to prevent the robots from converging at the same locations when they try to get to their targets
Application (artificial intelligence)	[27]	Based on new leader selection strategy to improved particle swarm optimization		X	X			X			The algorithm used triangular distance to select leader individuals that cover different regions in the Pareto frontier. The authors also included an update strategy for P_{best} with respect to their connected leaders. MOPSO tridist was shown to outperform other multiobjective PSOs, and the authors illustrated the algorithm's application with the digital watermarking problem for RBG images
	[13, 14]	Discrete PSO		X		X	X				For solving the problem of transmitting information on networks. The work result proves that the proposed discrete PSO outperforms Simulated Annealing (SA)
	[2]	Novel approach of particle swarm optimization (PSO)		X		X		X	X		The dynamic question generation system is built to select tailored questions for each learner from the item bank to satisfy multiple assessment requirements. The experimental results show that the PSO approach is suitable for the selection of near-optimal questions from large-scale item banks
Application (multichoice question test extraction)	[3]	Particle swarm optimization (PSO)		X	X	X	X		X		The authors used particle swarm optimization to generate tests with approximating difficulties to the required levels from users. The experiment result shows that PSO gives the best performance concerning most of the criteria
	[33]	Multiswarm single-objective particle swarm optimization		X	X	X		X	X		The authors use particle swarm optimization to generate multitests with approximating difficulties to the required levels from users. In this parallel stage, migration happens between swarms to exchange information between running threads to improve the convergence and diversities of solutions

3. Problem Statement

3.1. Problem of Generating Multiple Tests. In our previous works [3, 33], we have proposed a PSO-based method to multichoice test generation; however, it was a single-objective approach. In this paper, we introduce a multiobjective approach of multichoice test generation by combining PSO and SA algorithms.

Let $Q = \{q_1, q_2, q_3, \ldots, q_n\}$ be a question bank with n questions. Each question $q_i \in Q$ contains four attributes {QC, SC, QD, OD}. QC is a question identifier code and is used to avoid duplication of any question in the solution. SC denotes a section code of the question and is used to indicate which section the question belonged to. QD denotes a time limit of the question, and OD denotes a real value in the range [0.1, 0.9] that represents an objective difficulty (level) of the question. QC, SC, and QD are discrete positive integer values as in the work of Bui et al. [3] and Nguyen et al. [33].

The problem of generating multiple k tests (or just multiple tests) is to generate k number of tests simultaneously in a single run, e.g., our objective is to generate a set of tests, in which each test $E_i = \{q_{i1}, q_{i2}, q_{i3}, \ldots, q_{im}\}$ ($q_{ij} \in Q$, $1 \le j \le m$, $1 \le i \le k$, $k \le n$) consists of m ($m \le n$) questions. Additionally, those tests must satisfy both the requirements of objective difficulty ODR and testing time duration TR that were given by users. For example, ODR = 0.8 and TR = 45 minutes mean that all the generated tests must have approximately the level of difficulty equal to 0.8 and the test time equal to 45 minutes.

The objective difficulty of a test E_i is defined as $\mathrm{OD}_{E_i} = \sum_{j=1}^{m} q_{ij} \cdot \mathrm{OD}/m$, and the duration of the test E_i is determined by $T_{E_i} = \sum_{j=1}^{m} q_{ij} \cdot \mathrm{QD}$.

Besides the aforementioned requirements, there are additional constraints each generated test must satisfy as follows:

$C1$: each question in a generated test must be unique (i.e., a question cannot appear more than once in a test).

$C2$: in order to make the test more diverse, there exists no case that all questions in a test have the same difficulty value as the required objective difficulty ODR. For example, if ODR = 0.6, then $\exists\, q_{ki} \in T_{E_k}$: $q_{ki} \cdot \mathrm{OD} \neq 0.6$.

$C3$: some questions in a question bank must stay in the same groups because their content is relating to each other. The generated tests must ensure that all the questions in one group appear together. This means if a question of a specific group appears in a test, the remaining questions of the group must also be presented in the same test [3, 33].

$C4$: as users may require generated tests to have several sections, a generated test must ensure that the required numbers of questions are drawn out from question banks for each section.

3.2. Modeling MMPSO for the Problem of Generating Multiple Tests. The model for MMPSO for the problem of generating multiple tests can be represented as follows:

$$\begin{cases} \min f_1(x_1, x_2, x_3, \ldots, x_m), \\ \min f_2(x_1, x_2, x_3, \ldots, x_m), \\ \vdots \\ \min f_k(x_1, x_2, x_3, \ldots, x_m), \end{cases} \tag{2}$$

where $f_1, f_2, \ldots, f_k$ are swarms that represent multiple tests; $x_1, \ldots, x_m$ are the number of questions in the test.

Assume that F is an objective function for multiobjective of the problem; it can be formulated as follows:

$$F = \sum \alpha_i F_i, \tag{3}$$

where α_i is a weight constraint ($\alpha_i \in (0, 1]$) and $\sum \alpha_i = 1$ and F_i is a single-objective function. In this paper, we use an evaluation of the two functions, which are the average levels of difficulty requirements of the tests F_1 and total test duration F_2.

$F_1 = f(q_{kj} \cdot \mathrm{OD}) = (\sum_{j=1}^{m} q_{kj} \cdot \mathrm{OD}/m) - \mathrm{ODR}$, where F_1 satisfies the conditions {$C1$, $C2$, $C3$, $C4$} and m is the total number of questions in the test, $q_{kj} \cdot \mathrm{OD}$ is the difficulty value of each question, and ODR is the required difficulty level.

$F_2 = f(q_{kj} \cdot \mathrm{QD}) = 1 - (\sum_{i=1}^{m} q_{kj} \cdot \mathrm{QD}/\mathrm{TR})$, where F_2 satisfies the conditions {$C1$, $C2$, $C3$, $C4$} and m is the total number of questions in the test, $q_{kj} \cdot \mathrm{QD}$ is the duration for each question, and TR is the required total time of tests.

The objective function F is used as the fitness function in the MMPSO, and the results of the objective function are considered the fitness of the resulting test.

In this case, the better the fitness, the smaller the F becomes. To improve the quality of the test, we also take into account the constraints $C1$, $C2$, $C3$, and $C4$.

For example, provided that we have a question bank as in Table 2, the test extraction requirements are four questions, a difficulty level of 0.6 (ODR = 0.6), a total duration of the test of 300 seconds (TR = 300), and a weight constraint ($\alpha = 0.4$). Table 3 illustrates a candidate solution with its fitness = 0.1 computed by using formula (3).

4. MMPSO in Extracting Multiple Tests

4.1. Process of MMPSO for Extracting Tests. This paper proposes a parallel multiswarm multiobjective PSO (MMPSO) for extracting multiple tests (MMPSO) based on the idea in Bui et al. [3]. It can be described as follows. Creating an initial swarm population is the first step in PSO, in which each particle in a swarm is considered a candidate test; this first population also affects the speed of convergence to optimal solutions. This step randomly picks questions in a question bank. The questions, either stand-alone or staying in groups (constraint $C3$), are drawn out for one section (constraint $C4$) until the required number of questions of the section is reached and the drawing process is repeated for next sections. When the required number of questions of the candidate test and all the constraints are met, the fitness value of the generated test will be computed according to formula (3).

The G_{best} and P_{best} position information is the contained questions. All P_{best} slowly move towards G_{best} by using the

Table 2: The question banks.

QC	01	02	03	04	05	06	07	08	09	10
OD	0.3	0.2	0.8	0.7	0.4	0.6	0.5	0.8	0.2	0.3
QD	100	110	35	40	65	60	60	35	110	100

Table 3: An example of test results.

An individual that satisfies the requirement	QC	05	08	01	04	Fitness (1) (ODR = 0.6; TR = 300; $\alpha = 0.4$)
	OD	0.4	0.8	0.3	0.7	$(0.4 \times \lvert(2.2/4) - 0.6\rvert) + [(1 - 0.4) \times \lvert1 - (240/300)\rvert] = 0.1$
	QD	65	35	100	40	

location information of G_{best}. The movement is the replacement of some questions in the candidate test according to the velocity P_{best}. If the fitness value of a newly found P_{best} of a particle is smaller than the particle's currently best-known P_{best} (i.e., the new position is better than the old), then we assign a newly found position value to P_{best}.

G_{best} moves towards the final optimal solution in random directions. The movement is achieved by replacing its content with some random questions from the question bank. In a similar way to P_{best}, if the new position is no better than the old one, the G_{best} value will not be updated.

The algorithm ends when the fitness value is lower than the fitness threshold ε or the number of movements (iteration loops) surpasses the loop threshold λ. Both of the thresholds are given by users.

4.2. Migration Parallel MMPSO for the Extracting Test (Parallel MMPSO). Based on the idea in Nguyen et al. [33]; we present the migration parallel approach of MMPSO for increasing performance. Each swarm now corresponds to a thread, and the migration happens by chance between swarms. The migration method starts with locking the current thread (swarm) to avoid interference from other threads in.

In the dual-sector model [34], Lewis describes a relationship between two regions, the subsistence sector and the capitalist sector. We can view the two types of economic sectors here as the strong (capitalist) sectors and the weak (subsistence) sectors (while ignoring other related aspects of the economy). Whether a sector is strong or weak depends on the fitness value of G_{best} positions of its swarm. However, when applying those theories, some adjustments are made so that the parallel MMPSO can yield better optimal solutions.

The direction of migration changes when individuals with strong P_{best} (strong individuals) in strong sectors move to weak sectors. The weak sectors' G_{best} may be replaced by the incoming P_{best}, and the fitness value of the weak swarms should make a large lift, as in the work of Nguyen et al. [33].

Backward migration from the weak swarms to strong swarms also happens alongside forwarding migration. For every individual that moves from a strong swarm to a weak swarm, there is always one that moves from the weak swarm back to the strong swarm. This is to ensure that the number of particles and the searching capabilities of the swarms do not significantly decrease.

The foremost condition for migration to happen is that there are changes in the fitness values of the current G_{best} compared to the previous G_{best}.

The probability for migration is denoted as γ, and the unit is a percentage (%).

The number of migrating particles is equal to $\delta \times$ the size of the swarm (i.e., the number of existing particles in the swarm), where δ denotes the percentage of migration.

The migration parallel MMPSO-based approach to extract multiple tests is described in a form of a pseudocode in Algorithm 1.

The particle updates its velocity (V) and positions (P) with the following formulas:

$$V^{t+1} = V^t + r_1 \times V^t_{p\text{best}} + r_2 \times V^t_{g\text{best}}, \tag{4}$$

where $V_{p\text{best}}$ is the velocity of P_{best}, with $V_{p\text{best}}$ determined by $V_{p\text{best}} = \alpha \times m$; $V_{g\text{best}}$ is the velocity of G_{best}, with $V_{g\text{best}}$ determined by $V_{g\text{best}} = \beta \times m$, $\alpha, \beta \in (0, 1)$, r_1, r_2 are random values, and m is the number of questions in the test solutions.

$$P^{t+1} = P^t + V^{t+1}. \tag{5}$$

The process of generating multiple tests at the same time in a single run using migration parallel MMPSO includes two stages. The first stage is generating tests using multi-objective PSO. In this stage, the algorithm proceeds to find tests that satisfy all requirements and constraints using multiple threads. Each thread corresponds to each swarm that runs separately. The second stage is improving and diversifying tests. This stage happens when there is a change in the value of G_{best} of each swarm (for each thread) in the first stage. In this second stage, migration happens between swarms to exchange information between running threads to improve the convergence and diversity of solutions based on the work of Nguyen et al. [33]. The complete flowchart that applies the parallelized migration method to the MMPSO algorithm is shown in Figure 1.

4.3. Migration Parallel MMPSO in Combination with Simulated Annealing. As mentioned above, the initial population affects the convergence speed and diversity of test solutions. The creation of a set of initial solutions (population) is generally performed randomly in PSO. It is one of the drawbacks since the search space is too wide, so the probability of getting stuck in a local optimum solution is

Function Migration_Par_MMPSO: Extracting tests using Migration Parallel MMPSO
For each available thread t **do**
Generate the initial population with random questions;
While stop conditions are not met **do**
Gather G_{best} and all P_{best};
Foreach P_{best}
Move P_{best} towards G_{best} using location information of G_{best},;
Update velocity using equation (4);
Update position using equation (5);
End for
G_{best} moves in a random direction to search for the optimal solution;
If the probability for migration γ is met **then**
Execute function **Migration_MMPSO** with t;
End if
End while
End for
Function Migration _MMPSO: Improving solutions with migration method
Lock the current thread (i.e., block all modifications from other threads to the current thread) to avoid interference from other threads to the current thread during migration procedure.
Select λ, which are the set of stronger individuals for migration except for the G_{best};
Lock other threads so that no unintended changes will happen to them during the migration:
Choose a thread that has a G_{best} weaker than the one in the current thread;
Unlock the other threads except for chosen thread;
Set the status of the chosen thread to "Exchanging";
Move the λ selected individuals to the chosen thread;
Remove those λ selected individuals;
Select the λ weakest individuals in the chosen thread;
Add those λ weakest individuals to the current thread;
Set the status of chosen thread to "Available";
Unlock the current thread and the chosen thread;

Algorithm 1: Pseudocode: migration parallel MMPSO.

also high. In order to improve the initial population, we apply SA in the initial population creation step of migration parallel MMPSO instead of the random method. SA was selected since it is capable of escaping local optimums in Kharrat and Neji [35]. In this study, the process of finding new solutions using SA is improved by moving P_{best} to G_{best} using the received information about the location of G_{best} (which is commonly used in PSO). The MMPSO with SA is described by a pseudocode in Algorithm 2.

5. Experimental Studies

5.1. Experimental Environment and Data. Bui et al. [3] evaluated different algorithms such as the random method, genetic algorithms, and PSO-based algorithm for extracting tests from a question bank of varying sizes. The results of the experiment showed that the PSO-based algorithms are better than others. Hence, the experiment in this paper only evaluated and compared the improved SA parallel MMPSO algorithm with the normal parallel MMPSO algorithm in terms of the diversity of tests and the accuracy of solutions.

All proposed algorithms are implemented in C# and run on 2 computers which are a 2.5 GHz Desktop PC (4-CPUs, 4 GB RAM, Windows 10) and a 2.9 GHz VPS (16-CPUs, 16 GB RAM, Windows Server 2012). The experimental data include 2 question banks. One is with 998 different questions (the small question bank) and the other one is with 12000 different questions (the large question bank). The link to the data is https://drive.google.com/file/d/1_EdCUNyqC9IGziFUIf4mqs0G1qHtQyGI/view. The small question bank consists of multiple sections, and each section has more than 150 questions with different difficulty levels (Figure 2). The large question bank includes 12,000 different questions in which each part has 1000 questions with different difficulty levels (Figure 3). The experimental parameters of MMPSO are presented in Table 4. The results are shown in Tables 5 and 6 and Figures 4 and 5.

Our experiments focus on implementing formula (3) and an evaluation of the two functions, which are the average levels of difficulty requirements of the tests F_1 and total test duration F_2.

5.2. Evaluation Method. In this part, we present the formula for the evaluation of all algorithms about their stability to produce required tests with various weight constraints (α). The main measure is the standard deviation, which is defined as follows:

$$A = \sqrt{\frac{\sum_{i=1}^{z} (y_{i-} \overline{y})^2}{z-1}}, \quad (6)$$

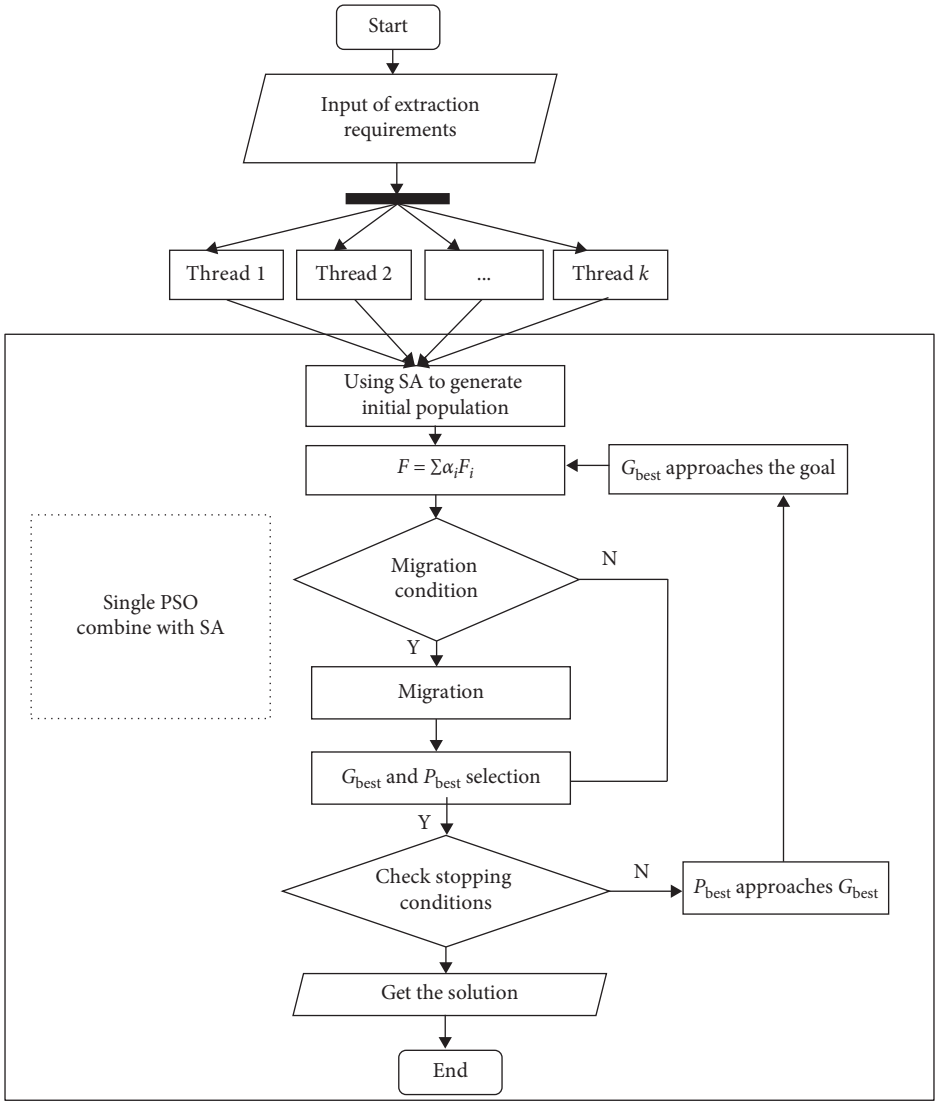


Figure 1: The flowchart of the MMPSO algorithm in migration parallel.

Function Migration_Par_MMPSO_SA: Extracting tests using Migration Parallel MMPSO and SA
For each available thread t **do**
Generate the initial population by using SA as it is capable of escaping local minimums. In this stage, the process of finding new solutions using SA is improved by moving P_{best} to G_{best} using the received information about the location of G_{best} and remove any incorrect solutions;
While stop conditions are not met **do**
Gather G_{best} and all P_{best};
For each P_{best}
Move P_{best} towards G_{best} using location information of G_{best};
Apply velocity update using (4),
Apply position update using (5),
End for
G_{best} moves in a random direction to search for the optimal solution;
If the probability for migration γ is met **then**
Execute function **Migration_MMPSO** with t;
End if
End while
End for

Algorithm 2: Pseudocode: migration parallel MMPSO with SA.

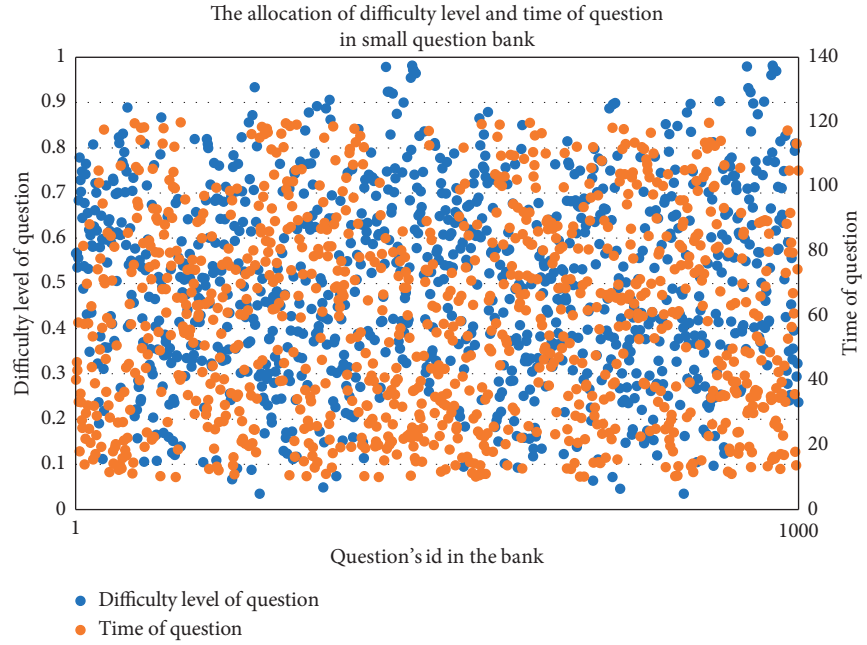


Figure 2: The allocation of the difficulty level and time of question in the small question bank.

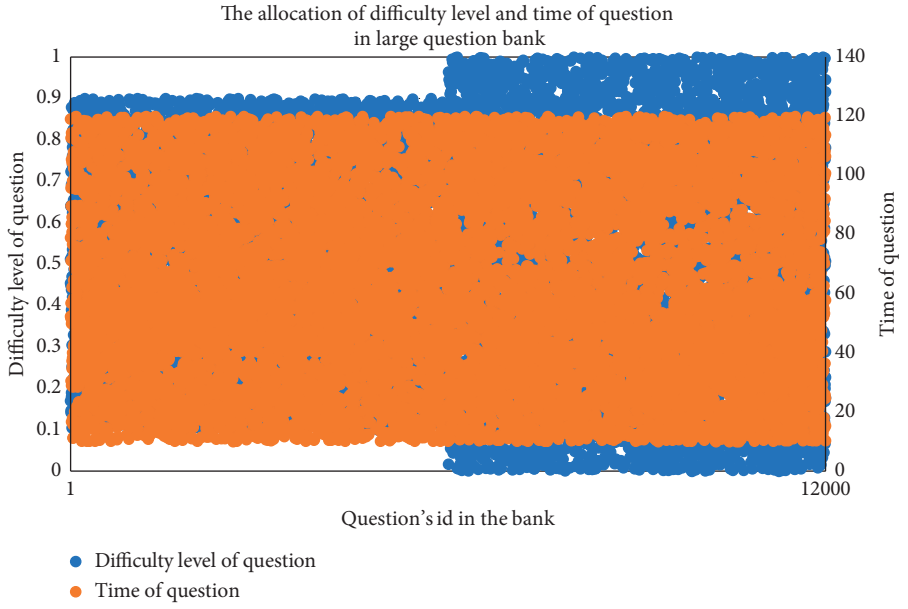


Figure 3: The allocation of the difficulty level and time of question in the large question bank.

Table 4: Experimental parameters.

The required level of difficulty (ODR)	0.5
The total test time (TR)	5400 (seconds)
The value of α	[0.1, 0.9]
The number of required questions in a test	100
The number of questions in each section in the test	10

Table 4: Continued.

The number of simultaneously generated tests in each run	100
The number of questions in the bank	1000 and 12,000
The PSO's parameters	The number of particles in each swarm: 10 Random value $r1$, $r2$ are in [0,1] The percentage of P_{best} individuals which receive position information from G_{best} ($C1$): 5% The percentage of G_{best} which moves to final goals ($C2$): 5% The percentage of **migration** δ: 10% The percentage of **migration probability** γ: 5% The stop condition: either when the tolerance fitness <0.001 or when the number of movement loops >1000
The SA's parameters	Initial temperature: 100 Cooling rate: 0.9 Termination temperature: 0.01 Number of iterations: 100

Table 5: Experimental results in the small question bank.

Algorithms	Weight constraint (α)	Number of runs	Successful times	Average runtime for extracting tests (second)	Average number of iteration loops	Average fitness	Average duplicate (%)	Standard deviation
Parallel multiswarm multiobjective PSO (parallel MMPSO)	0.1	50	11	61.3658	999.75	0.003102	2.43	0.0007071
	0.2	50	445	47.9793	981.03	0.003117	2.64	0.0014707
	0.3	50	425	35.8007	957.53	0.004150	2.73	0.0021772
	0.4	50	530	30.5070	928.10	0.004850	2.80	0.0027973
	0.5	50	774	29.5425	877.65	0.004922	2.85	0.0033383
	0.6	50	1410	22.6973	754.82	0.003965	2.91	0.0034005
	0.7	50	2900	14.9059	470.13	0.002026	2.97	0.0022461
	0.8	50	3005	16.7581	488.31	0.001709	3.01	0.0017271
	0.9	50	3019	28.5975	619.34	0.001358	3.04	0.0009634
Parallel multiswarm multiobjective PSO with SA (parallel MMPSO with SA)	0.1	50	4	142.2539	999.98	0.003080	2.98	0.0006496
	0.2	50	2912	132.3828	900.42	0.001265	3.27	0.0007454
	0.3	50	3681	111.9513	650.42	0.001123	3.38	0.0008364
	0.4	50	3933	100.0204	474.91	0.001085	3.44	0.0009905
	0.5	50	4311	91.7621	318.75	0.000938	3.48	0.0008439
	0.6	50	4776	84.7441	161.23	0.000746	3.53	0.0005124
	0.7	50	4990	81.1127	76.75	0.000666	3.54	0.0002421
	0.8	50	4978	84.6747	131.32	0.000679	3.52	0.0002518
	0.9	50	4937	98.8690	339.89	0.000749	3.41	0.0002338
Migration parallel multiswarm multiobjective PSO (migration parallel MMPSO)	0.1	50	575	51.3890	959.29	0.002138	5.19	0.0008091
	0.2	50	1426	33.2578	837.87	0.002119	5.60	0.0011804
	0.3	50	1518	25.3135	779.76	0.002587	5.82	0.0017130
	0.4	50	1545	21.0524	745.36	0.002977	5.89	0.0021845
	0.5	50	1650	17.9976	710.92	0.003177	5.95	0.0025374
	0.6	50	1751	16.0573	680.97	0.003272	5.92	0.0028531
	0.7	50	2463	12.9467	540.21	0.002243	5.94	0.0022161
	0.8	50	3315	12.7420	402.75	0.001374	5.90	0.0012852
	0.9	50	3631	19.1735	439.27	0.001067	5.85	0.0006259
Migration parallel multiswarm multiobjective PSO with SA (migration parallel MMPSO with SA)	0.1	50	816	139.6821	952.42	0.002183	3.82	0.0009039
	0.2	50	3641	111.4438	638.08	0.001039	5.19	0.0005336
	0.3	50	3958	98.9966	463.53	0.000984	5.36	0.0006349
	0.4	50	4084	92.6536	357.13	0.000973	5.30	0.0007475
	0.5	50	4344	88.3098	255.46	0.000898	5.10	0.0007442
	0.6	50	4703	83.4776	144.00	0.000758	4.90	0.0004939
	0.7	50	4874	81.2981	84.57	0.000697	4.70	0.0003271
	0.8	50	4955	84.2094	106.68	0.000683	4.45	0.0002609
	0.9	50	4937	94.1685	267.30	0.000746	4.19	0.0002345

Table 6: Experimental results in the large question bank.

Algorithms	Weight constraint (α)	Number of runs	Successful times	Average runtime for extracting tests (second)	Average number of iteration loops	Average fitness	Average duplicate (%)	Standard deviation
Parallel multiswarm multiobjective PSO (parallel MMPSO)	0.1	50	2931	23.50	888.85	0.001137	0.95	0.000476
	0.2	50	4999	14.05	484.33	0.000725	1.03	0.000219
	0.3	50	4997	9.56	296.80	0.000689	1.04	0.000234
	0.4	50	4999	5.99	190.55	0.000676	1.04	0.000233
	0.5	50	5000	3.61	121.24	0.000668	1.05	0.000236
	0.6	50	5000	2.77	79.32	0.000663	1.05	0.000235
	0.7	50	5000	3.22	92.33	0.000669	1.05	0.000238
	0.8	50	5000	4.92	173.19	0.000673	1.04	0.000231
	0.9	50	5000	10.98	384.13	0.000738	1.02	0.000213
Parallel multiswarm multiobjective PSO with SA (parallel MMPSO with SA)	0.1	50	3055	99.75	890.40	0.001095	0.96	0.000432
	0.2	50	5000	84.90	469.23	0.000709	1.04	0.000224
	0.3	50	5000	74.43	275.54	0.000686	1.05	0.000230
	0.4	50	5000	73.03	168.91	0.000668	1.06	0.000237
	0.5	50	5000	69.88	99.92	0.000663	1.07	0.000235
	0.6	50	5000	67.34	61.42	0.000662	1.07	0.000236
	0.7	50	5000	53.43	69.02	0.000661	1.07	0.000235
	0.8	50	5000	70.06	132.27	0.000676	1.06	0.000236
	0.9	50	5000	79.58	319.36	0.000734	1.03	0.000211
Migration parallel multiswarm multiobjective PSO (migration parallel MMPSO)	0.1	50	2943	33.52	886.90	0.001144	0.95	0.000482
	0.2	50	4995	19.33	488.60	0.000724	1.02	0.000219
	0.3	50	4998	12.43	295.69	0.000688	1.04	0.000231
	0.4	50	5000	8.57	190.20	0.000667	1.04	0.000238
	0.5	50	4999	6.02	120.88	0.000665	1.05	0.000240
	0.6	50	5000	4.44	78.89	0.000669	1.04	0.000234
	0.7	50	5000	4.98	92.53	0.000668	1.05	0.000234
	0.8	50	5000	7.94	171.98	0.000669	1.04	0.000236
	0.9	50	5000	15.76	383.09	0.000738	1.02	0.000209
Migration parallel multiswarm multiobjective PSO with SA (migration parallel MMPSO with SA)	0.1	50	3122	102.00	888.48	0.001091	0.96	0.000436
	0.2	50	5000	85.68	469.50	0.000716	1.04	0.000222
	0.3	50	5000	77.35	276.03	0.000678	1.05	0.000235
	0.4	50	5000	73.19	167.84	0.000674	1.06	0.000234
	0.5	50	5000	69.62	99.66	0.000665	1.06	0.000238
	0.6	50	5000	67.83	61.33	0.000660	1.07	0.000237
	0.7	50	5000	64.19	69.02	0.000666	1.07	0.000237
	0.8	50	5000	61.46	133.45	0.000673	1.06	0.000234
	0.9	50	5000	71.62	319.68	0.000731	1.03	0.000213

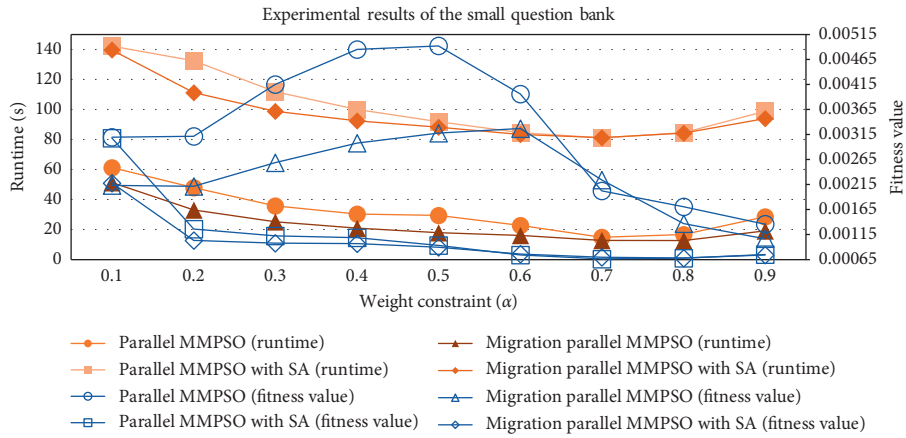


Figure 4: Experimental results of the runtime and fitness value are in Table 4.

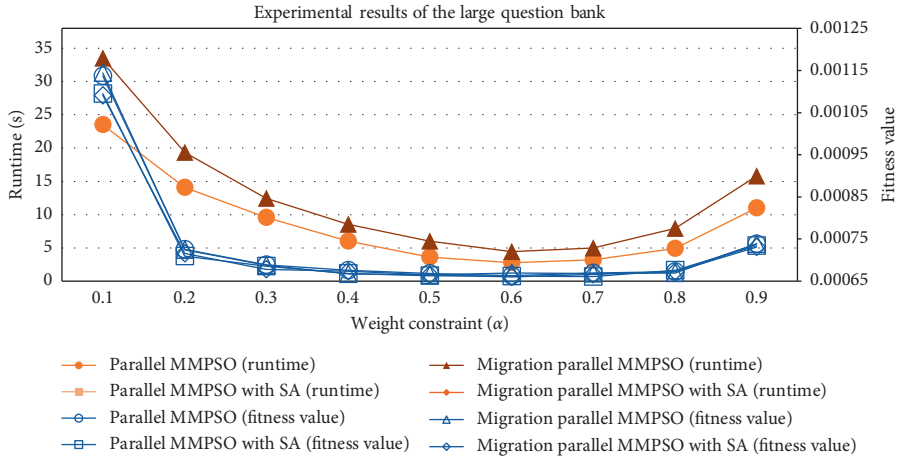


Figure 5: Experimental results of the runtime and fitness value are in Table 5.

where z is the number of experimental runs. $\overline{y}$ is the average fitness of all runs. y_i is a fitness value of run i^{th}.

The standard deviation is used to assess the stability of the algorithms. If its value is low, then the generated tests of each run do not have much difference in the fitness value. The weight constraint α is also being examined as it balances the objective functions. In our cases, a change in α can shift the importance towards the test duration constraint, the test difficulty constraint, or the balance between those two. We can select α to suit what we require, emphasizing more on either the test duration or the test difficulty.

5.3. Experimental Results. The experiments are executed with the parameters following Ridge and Kudenko [36] in Table 3, and the results are presented in Table 5 (run on computer 4-CPUs) and Table 6 (run on computer 16-CPUs). The comparisons of runtime and fitness of the small and large question bank are presented in Figures 4 and 5. Regarding Tables 5 and 6, each run extracts 100 tests simultaneously, and each test has a fitness value. Each run also requires several iteration loops to successfully extract 100 candidate tests. The average runtime for extracting tests is the average runtimes of all 50 experimental runs. The average number of iteration loops is the average of all required loops of all 50 runs. The average fitness is the average of all fitness values of 5000 generated tests. The average duplicate indicates the average number of duplicate questions among 100 generated tests of all 50 runs. The average duplicate is also used to indicate the diversity of tests. The lower the value, the more diverse the tests.

When α is at the lower range [0.1, 0.5], the correctness for difficulty value of each generated test is emphasized more than that of the total test time. Based on the average fitness value, all algorithms appear to have a harder time generating tests at the lower range [0.1, 0.5] compared with at the higher range [0.5, 0.9]. Additionally, when α increases, the runtime starts to decrease, the fitness gets better (i.e., the fitness values get smaller), and the numbers of loops required for generating tests decrease. Apparently, satisfying the requirement for the test difficulty requirement is harder than satisfying the requirement for total test time. The experiment results also show that integrating SA gives a better fitness value without SA. However, runtimes of algorithms with SA are longer as a trade-off for better fitness values.

All algorithms can generate tests with acceptable percentages of duplicate questions among generated tests. The duplicate question proportions between generated tests depend on the sizes of the question bank. For example, if the question bank's size is 100, we need to generate 50 tests in a single run and each test contains 30 questions, and then, some generated tests should contain similar questions of the other generated tests.

Based on the standard deviation in Tables 5 and 6, all MMPSO algorithms with SA are more stable than those without SA since the standard deviation values of those with SA are smaller. In other words, the differences in fitness values between runs are smaller with SA than without SA. The smaller standard deviation values and smaller average fitness values also mean that we less likely need to rerun the MMPSO with SA algorithms many times to get the generated tests that better fit the test requirements. The reason is that the generated tests we obtain at the first run are likely close to the requirements (due to the low fitness value) and the chance that we obtain those generated tests with less fit to requirements is low (due to the low standard deviation value).

6. Conclusions and Future Studies

Generation of question papers through a question bank is an important activity in extracting multichoice tests. The quality of multichoice questions is good (diversity of the level of difficulty of the question and a large number of questions in question bank). In this paper, we propose the use of MMPSO to solve the problem of generating multi-objective multiple k tests in a single run. The objectives of the tests are the required level of difficulty and the required total test time. The experiments evaluate two algorithms, MMPSO with SA and normal MMPSO. The results indicate that MMPSO with SA gives better solutions than normal

MMPSO based on various criteria such as diversities of solutions and numbers of successful attempts.

Future studies may focus on investigating the use of the proposed hybrid approach [37, 38] to solve other NP-hard and combinatorial optimization problems, which focus on fine-tuning the PSO parameters by using some type of adaptive strategies. Additionally, we will extend our problem to provide feedback to instructors from multiple-choice data, such as using fuzzy theory [39], and PSO with SA for mining association rules to compute the difficulty levels of questions.

Data Availability

The data used in this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] F. Ting, J. H. Wei, C. T. Kim, and Q. Tian, "Question classification for E-learning by artificial neural network," in *Proceedings of the Fourth International Conference on Information, Communications and Signal Processing*, pp. 1575–1761, Hefei, China, 2003.

[2] S. C. Cheng, Y. T. Lin, and Y. M. Huang, "Dynamic question generation system for web-based testing using particle swarm optimization," *Expert Systems with Applications*, vol. 36, no. 1, pp. 616–624, 2009.

[3] T. Bui, T. Nguyen, B. Vo, T. Nguyen, W. Pedrycz, and V. Snasel, "Application of particle swarm optimization to create multiple-choice tests," *Journal of Information Science & Engineering*, vol. 34, no. 6, pp. 1405–1423, 2018.

[4] M. Yildirim, "Heuristic optimization methods for generating test from a question bank," *Advances in Artificial Intelligence*, pp. 1218–1229, Springer, Berlin, Germany, 2007.

[5] M. Yildirim, "A genetic algorithm for generating test from a question bank," *Computer Applications in Engineering Education*, vol. 18, no. 2, pp. 298–305, 2010.

[6] B. R. Antonio and S. Chen, "Multi-swarm hybrid for multimodal optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1759–1766, Brisbane, Australia, June 2012.

[7] N. Hou, F. He, Y. Zhou, Y. Chen, and X. Yan, "A parallel genetic algorithm with dispersion correction for HW/SW partitioning on multi-core CPU and many-core GPU," *IEEE Access*, vol. 6, pp. 883–898, 2018.

[8] M. E. C. Bento, D. Dotta, R. Kuiava, and R. A. Ramos, "A procedure to design fault-tolerant wide-area damping controllers," *IEEE Access*, vol. 6, pp. 23383–23405, 2018.

[9] C. Han, L. Wang, Z. Zhang, J. Xie, and Z. Xing, "A multi-objective genetic algorithm based on fitting and interpolation," *IEEE Access*, vol. 6, pp. 22920–22929, 2018.

[10] J. Lu, Z. Chen, Y. Yang, and L. V. Ming, "Online estimation of state of power for lithium-ion batteries in electric vehicles using genetic algorithm," *IEEE Access*, vol. 6, pp. 20868–20880, 2018.

[11] X.-Y. Liu, Y. Liang, S. Wang, Z.-Y. Yang, and H.-S. Ye, "A hybrid genetic algorithm with wrapper-embedded approaches for feature selection," *IEEE Access*, vol. 6, pp. 22863–22874, 2018.

[12] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, 1995.

[13] G. Sen, M. Krishnamoorthy, N. Rangaraj, and V. Narayanan, "Exact approaches for static data segment allocation problem in an information network," *Computers & Operations Research*, vol. 62, pp. 282–295, 2015.

[14] G. Sen and M. Krishnamoorthy, "Discrete particle swarm optimization algorithms for two variants of the static data segment location problem," *Applied Intelligence*, vol. 48, no. 3, pp. 771–790, 2018.

[15] M. Q. Peng, Y. J. Gong, J. J. Li, and Y. B. Lin, "Multi-swarm particle swarm optimization with multiple learning strategies," in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pp. 15-16, New York; NY, USA, 2014.

[16] R. Vafashoar and M. R. Meybodi, "Multi swarm optimization algorithm with adaptive connectivity degree," *Applied Intelligence*, vol. 48, no. 4, pp. 909–941, 2017.

[17] X. Xia, L. Gui, and Z.-H. Zhan, "A multi-swarm particle swarm optimization algorithm based on dynamical topology and purposeful detecting," *Applied Soft Computing*, vol. 67, pp. 126–140, 2018.

[18] B. Nakisa, M. N. Rastgoo, M. F. Nasrudin, M. Zakree, and A. Nazri, "A multi-swarm particle swarm optimization with local search on multi-robot search system," *Journal of Theoretical and Applied Information Technology*, vol. 71, no. 1, pp. 129–136, 2015.

[19] M. Li and F. Babak, "Multi-objective particle swarm optimization with gradient descent search," *International Journal of Swarm Intelligence and Evolutionary Computation*, vol. 4, pp. 1–9, 2014.

[20] S. P. Kapse and S. Krishnapillai, "An improved multi-objective particle swarm optimization based on utopia point guided search," *International Journal of Applied Metaheuristic Computing*, vol. 9, no. 4, pp. 71–96, 2018.

[21] T. Cheng, M. F. P. J. Chen, Z. Yang, and S. Gan, "A novel hybrid teaching learning based multi-objective particle swarm optimization," *Neurocomputing*, vol. 222, pp. 11–25, 2016.

[22] H. A. Adel, L. Songfeng, E. A. A. Yahya, and A. G. A. Arkan, "A particle swarm optimization based on multi objective functions with uniform design," *SSRG International Journal of Computer Science and Engineering*, vol. 3, pp. 20–26, 2016.

[23] D. M. Alan, T. Gregorio, H. B. Z. Jose, and T. L. Edgar, "R2-based multi/many-objective particle swarm optimization," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 1898527, 10 pages, 2016.

[24] S. P. Kapse and S. Krishnapillai, "An improved multi-objective particle swarm optimization algorithm based on adaptive local search," *International Journal of Applied Evolutionary Computation*, vol. 8, no. 2, pp. 1–29, 2017.

[25] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[26] Z. H. Zhan, J. Li, and J. Cao, "Multiple populations for multiple objectives: a coevolutionary technique for solving multiobjective optimization problems," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 445–463, 2013.

[27] N. ShiZhang and K. K. Mishra, "Improved multi-objective particle swarm optimization algorithm for optimizing watermark strength in color image watermarking," *Applied Intelligence*, vol. 47, no. 2, pp. 362–381, 2017.

[28] X. Liansong and P. Dazhi, "Multi-objective optimization based on chaotic particle swarm optimization," *International Journal of Machine Learning and Computing*, vol. 8, no. 3, pp. 229–235, 2018.
[29] Y. Xiang and Z. Xueqing, "Multi-swarm comprehensive learning particle swarm optimization for solving multi-objective optimization problems," *PLoS One*, vol. 12, no. 2, Article ID e, 2017.
[30] V. Mokarram and M. R. Banan, "A new PSO-based algorithm for multi-objective optimization with continuous and discrete design variables," *Structural and Multidisciplinary Optimization*, vol. 57, no. 2, pp. 509–533, 2017.
[31] Z. B. M. Z. Mohd, J. Kanesan, J. H. Chuah, S. Dhanapal, and G. Kendall, "A multi-objective particle swarm optimization algorithm based on dynamic boundary search for constrained optimization," *Applied Soft Computing*, vol. 70, pp. 680–700, 2018.
[32] R. Liu, J. Li, J. Fan, C. Mu, and L. Jiao, "A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization," *European Journal of Operational Research*, vol. 261, no. 3, pp. 1028–1051, 2017.
[33] T. Nguyen, T. Bui, and B. Vo, "Multi-swarm single-objective particle swarm optimization to extract multiple-choice tests," *Vietnam Journal of Computer Science*, vol. 6, no. 2, pp. 147–161, 2019.
[34] D. Hunt, "W. A. Lewis on "economic development with unlimited supplies of labour"" *Economic Theories of Development: An Analysis of Competing Paradigms*, pp. 87–95, Harvester Wheatsheaf, New York, NY, USA, 1989.
[35] A. Kharrat and M. Neji, "A hybrid feature selection for MRI brain tumor classification," *Innovations in Bio-Inspired Computing and Applications*, pp. 329–338, Springer, Berlin, Germany, 2018.
[36] E. Ridge and D. Kudenko, "Tuning an algorithm using design of experiments," *Experimental Methods for the Analysis of Optimization Algorithms*, pp. 265–286, Springer, Berlin, Germany, 2010.
[37] P. Riccardo, I. Umberto, L. Giampaolo, and L. Stefano, "Global/local hybridization of the multi-objective particle swarm optimization with derivative-free multi-objective local search," in *Proceedings of the Congress of the Italian Society of Industrial and Applied Mathematics (SIMAI)*, pp. 198–209, At Milan, Italy, 2017.
[38] S. Sedarous, S. M. El-Gokhy, and E. Sallam, "Multi-swarm multi-objective optimization based on a hybrid strategy," *Alexandria Engineering Journal*, vol. 57, no. 3, pp. 1619–1629, 2018.
[39] H. S. Le and H. Fujita, "Neural-fuzzy with representative sets for prediction of student performance," *Applied Intelligence*, vol. 49, no. 1, pp. 172–187, 2019.

Hindawi
Scientific Programming
Volume 2020, Article ID 7846264, 9 pages
https://doi.org/10.1155/2020/7846264

Research Article

FaceFilter: Face Identification with Deep Learning and Filter Algorithm

Mohammed Alghaili,[1] Zhiyong Li,[1] and Hamdi A. R. Ali[2]

[1]*College of Information Science and Engineering, Hunan University, Changsha 410000, China*
[2]*Computer Science Department, Hajjah University, Hajjah, Yemen*

Correspondence should be addressed to Zhiyong Li; zhiyong.li@hnu.edu.cn

Received 21 November 2019; Accepted 2 April 2020; Published 1 August 2020

Academic Editor: Kifayat Ullah Khan

Although significant advances have been made recently in the field of face recognition, these have some limitations, especially when faces are in different poses or have different levels of illumination, or when the face is blurred. In this study, we present a system that can directly identify an individual under all conditions by extracting the most important features and using them to identify a person. Our method uses a deep convolutional network that is trained to extract the most important features. A filter is then used to select the most significant of these features by finding features greater than zero, storing their indices, and comparing the features of other identities with the same indices as the original image. Finally, the selected features of each identity in the dataset are subtracted from features of the original image to find the minimum number that refers to that identity. This method gives good results, as we only extract the most important features using the filter to recognize the face in different poses. We achieve state-of-the-art face recognition performance using only half of the 128 bytes per face. The system has an accuracy of 99.7% on the Labeled Faces in the Wild dataset and 94.02% on YouTube Faces DB.

1. Introduction

Recently, deep neural networks and especially convolutional neural networks (CNNs) have become the most commonly used method for feature representation and have achieved good results in face recognition problems. Face recognition can be divided into two categories: face verification, where two faces are presented and the system needs to verify whether these two faces belong to the same person, and face identification, where a face image is presented with an unknown identity and the system needs to determine this identity.

Most existing works that have focused on face recognition have achieved a high level of success [1–13]. However, if the pose is significantly changed or the face is presented at an angle, the individual cannot be identified.

Previous approaches to face recognition that are based on the discriminative classification model (face identification) are trained on a dataset of known identities, and an intermediate bottleneck layer is used as a representation for recognition. This approach generalizes a very large representation for each face, but some works have tried to reduce this dimensionality using PCA [10].

Another approach used in FaceNet [14] directly trained its output to obtain 128-D embedding using a triplet-based loss function based on LMNN [9]. These triplets comprise two matching faces and a nonmatching face. The aim of the triplet loss function is to separate positive results from negative ones by a certain distance margin.

In contrast, our approach uses an unsupervised learning technique to obtain 128 bytes per face and then passes these bytes to a filter in order to find the most suitable representation for each face. We then reduce the dimensionality of the representation to half of the 128 bytes, to match the original face with other faces to find the identity. This approach can identify a given face in different poses and can identify other faces that are most similar to the original identity.

Figure 1: Different poses of a single face with different types of illumination; all of these are identified correctly by the system.

As an illustration, Figure 1 shows a picture of a single individual at different angles and in different poses.

The remainder of this paper is organized as follows: Section 2 discusses the most important related work in face recognition. Our method is presented in Section 3, including a description of deep neural networks and our algorithm for handling the features. Sections 4 and 5 present some quantitative results and an evaluation of these.

2. Related Works

Our approach is similar to other recent works [3, 10, 14] in that it learns its representation directly from the face. However, instead of using a vector of features for reidentification, we reduce the vector representation to half of the features extracted for each face. We use a deep convolutional neural network architecture inspired by the NN4 FaceNet [14] and OpenFace [15] networks, but we remove the L2 normalization layer and instead use another fully connected layer.

There are an enormous number of studies of face recognition, and we will briefly discuss the most relevant works.

Huang et al. [16] proposed a convolutional deep belief network based on local convolutional restricted Boltzmann machines to learn a face representation. The learning method was unsupervised learning and the training was on an unlabeled natural image dataset. After that, they transfer the learned representation to a face identification through a classification method such as SVM.

Another attempt for face recognition was proposed by Taigman et al. [17]. This approach called DeepFace and it is one of the earlier large-scale applications of a 3D model for face recognition. They extracted the face representation using a nine-layer DeepFace model which mainly consists of two convolutional layers, three locally connected layers, and two fully connected (FC) layers with more than 120 million parameters using several locally connected layers without weight sharing. Their system was trained on 4.4 M 2D facial images of 4,030 identities and they achieved an accuracy of 97.35% on the benchmark LFW [18] dataset.

Schroff et al. proposed a CNN-based approach used for face recognition and clustering. This approach is called FaceNet [14] which is based on eleven convolutional and three FC layers. They have trained a deep convolutional network on a dataset of 200 M faces and 8 M identities and triplet loss function to directly optimize the embedding instead of an intermediate bottleneck layer as in the previous works. They have used triplets of roughly aligned matching/nonmatching face patches using an online triplet mining method, and they achieved the performance of state-of-the-art face recognition with 128 bytes for each face.

Sun et al. proposed another framework called DeepID [5, 6, 10] for face identification and verification. Their approach utilized an ensemble of shallower and smaller deep convolutional networks than DeepFace, i.e., every DCNN has four convolutional layers and uses 39, 31, and 1 patches, respectively, as an input. Their framework was trained on 202,599 images of 10,177 subjects. Their approach is considered as the first approach that achieved results that surpass human performance for face verification on the LFW dataset.

Parkhi et al. [19] collected a face dataset of 2.6 M 2D faces from 2,622 identities by proposing a new method for crawling the faces from the web. They presented a VGG-Face model consisting of 16 convolutional layers and three fully connected (FC) layers. The authors claimed that they achieved 98.95% accuracy on the LFW [18] dataset.

Deep 3D face recognition results have been represented by Kim et al. [20]. They fine-tuned the VGG-Face network [19] on 3D depth images. After that, they reported their results on three public datasets. They used an augmented dataset of 123,325 depth images to fine-tune VGG-Face. After that, they tested the model on Bosphorus [21], BU3DFE [5], and 3D-TEC (twins) [22] datasets. But their

results do not perform as the state-of-the-art results of the convolutional methods.

3. Method

3.1. Deep Convolutional Networks. We used a deep neural network structure called an NN4 neural network. Before they were input to the network, we resized all images to a size of $96 \times 96 \times 3$. These were used as input to the first convolutional layer, which has 64 kernels of size $7 \times 7 \times 3$ with stride 2. The second convolutional layer has 64 kernels of size $1 \times 1 \times 3$ with stride 2, and in the third convolutional layer, 192 kernels are used with size $3 \times 3 \times 3$ and stride 2. After these layers, an inception architecture was used in which there were six blocks labeled inception 3a, inception 3b, inception 3c, inception 4a, inception 4e, and inception 5a [23].

Since the input of the network was $96 \times 96 \times 3$ and the receptive field was small, the computational requirement was drastically reduced. The total number of parameters was 3,743,925, and the number of trainable parameters was 3,734,613, with 9,312 nontrainable parameters. We trained the network using a stochastic gradient descent (SGD) algorithm with a learning rate starting from 0.05 on a GPU. The model was trained on 202,599 face images of 10,177 subjects. Table 1 shows the network structure. Figure 2 depicts the model diagram while Figure 3 illustrates in detail the structure of inceptions used in this study.

Before training, we used the FaceNet [14] weights as a baseline in our network which used the triplet loss function in its training. Then, we used the Kullback–Leibler (KL) divergence loss functions to train our model as in Variational Feature Learning (VFL) [24] loss function. The difference between our loss function and VFL loss function is that, in VFL, they used the same input and output for two fully connected layers to be used to predict the mean μ and standard deviation σ of a Gaussian distribution. The mean μ and standard deviation σ are used to calculate the loss function which employed the Kullback–Leibler (KL) divergence loss. But in our training, since all input and output for the two fully connected layers are the same, we used one fully connected layer "fc1" in the network to be used to predict the mean μ and standard deviation σ of a Gaussian distribution. The mean μ and standard deviation σ are used to calculate the loss function as follows:

$$\mathrm{KL} = -\frac{1}{2}\sum_{i=1}^{n}\left(1 + \log\left(\sigma_i\right) - \mu_i^2 - \sigma_i\right), \tag{1}$$

where n denotes the output vector size, i.e., 128 in our training.

The network is trained with a softmax classifier for 200 epochs by using an Adam optimizer [25] and learning rate starting from 0.05. The training dataset divided into 70% for the training set and 30% for the validation set.

3.2. Face Reidentification Equations. Each original image x that we want to predict is represented by $f(x) \in R$ as a vector of 128 bytes indexes from 1 to 128. This can be expressed as in (1):

$$f\left(x_o\right) = \left(v^1, v^2, v^3, \ldots, v^n\right), \quad \text{where } n = 128, \tag{2}$$

where x_o is the original image that we want to predict and n is the number of features v in that vector. The vectors of the identities in the dataset will also be extracted, as it is expressed as in (2), and kept in a separate model file:

$$f\left(x_{id}^i\right) = \left(v_{id}^1, v_{id}^2, v_{id}^3, \ldots, v_{id}^n\right), \tag{3}$$

where $n = 128$ and x_{id}^i is the image of a particular identity id and i refers to the number of that identity in the dataset. After extracting the vectors, we will pass the vector of the original image to a filter to extract the most important values that can represent the original image. The filter works as a net to select the highest values among the features in the vector of the original image. It takes the values greater than zero with their corresponding position, i.e., indices of each value:

$$\mathrm{val},\ \mathrm{ind} = \sum_{i}^{n} f\left(v_o^i,\ \mathrm{index}_o^i\right), \quad \text{where } x_o^i > 0, \tag{4}$$

where n is the number of features v_o^i in the vector of the original image, i.e., 128, and i is the index of each feature in the vector. The selected features v_o^i which have values greater than zero will be stored in val while their corresponding indices will be stored in ind. So, we can select all features of each image in the identities of the dataset with the same indices of the selected features of the original image:

$$\mathrm{val}_{id}^i = \sum_{i=1}^{ni\,d} f\left(v_{id}^i,\ \mathrm{index}_{id}^i\right), \quad \text{for each } \mathrm{index}_{id}^i = \mathrm{index}_o^i, \tag{5}$$

where id refers to a particular identity in the set of identities i and $ni\,d$ is the number of features in each image of the identities. The selected features v_{id}^i of the identity id will be chosen if its indices index_{id}^i is equal to the indices of the selected features of the original image index_o^i and will be stored in val_{id}^i. Here, we do not need to select the values greater than zero for each identity in the dataset; rather, we just take the values corresponding to the indices of the largest values in the original image. This step is very important as the features of an eye, for example, may store in a particular index; consequently, we need to take the feature of that eye in each image in the dataset.

To recognize the identity, we will calculate the distance between the filtered values of the original image and the corresponding values of each identity image in the dataset. The lowest distance between the filtered values of the original image and a particular identity image both will have the same identity:

$$\mathrm{iden} = \min\left(\mathrm{val} - \begin{pmatrix} \mathrm{val}_{id}^1 \\ \mathrm{val}_{id}^2 \\ \mathrm{val}_{id}^3 \\ \vdots \\ \mathrm{val}_{id}^i \end{pmatrix}\right), \tag{6}$$

where i refers to the number of identities. It should be noticed that we have only weights of the images of all identities obtained by the model where these weights

Table 1: The structure of the neural network.

Layer	Size in	Size out	Kernel	Feature map	No. of parameters
Input	96 × 96 × 3				
ZeroPadding2D	96 × 96 × 3	102 × 102 × 3			
Conv1	102 × 102 × 3	48 × 48 × 64	7 × 7 × 3, 2	64	9472
Norm	48 × 48 × 64	48 × 48 × 64			
ZeroPadding	48 × 48 × 64	50 × 50 × 64			
Max pool	50 × 50 × 64	24 × 24 × 64	7 × 7 × 3, 2		
Conv2	24 × 24 × 64	24 × 24 × 64	1 × 1 × 3, 2	64	4160
Norm	24 × 24 × 64	24 × 24 × 64			
ZeroPadding	24 × 24 × 64	26 × 26 × 64			
Conv3	26 × 26 × 64	24 × 24 × 192	3 × 3 × 3, 2	192	110784
Norm	24 × 24 × 192	24 × 24 × 192			
ZeroPadding	24 × 24 × 192	26 × 26 × 192			
Max pool	26 × 26 × 192	12 × 12 × 192	3 × 3 × 3, 2		
Inception 3a	12 × 12 × 192	12 × 12 × 256			
Inception 3b	12 × 12 × 256	12 × 12 × 320			
Inception 3c	12 × 12 × 320	6 × 6 × 640			
Inception 4a	6 × 6 × 640	6 × 6 × 640			
Inception 4e	6 × 6 × 640	3 × 3 × 1024			
Inception 5a	3 × 3 × 1024	3 × 3 × 736			
Average pool	3 × 3 × 736	1 × 1 × 736	3 × 3 × 3, 1		
fc1	1 × 1 × 736	128			
Fc	128	1			

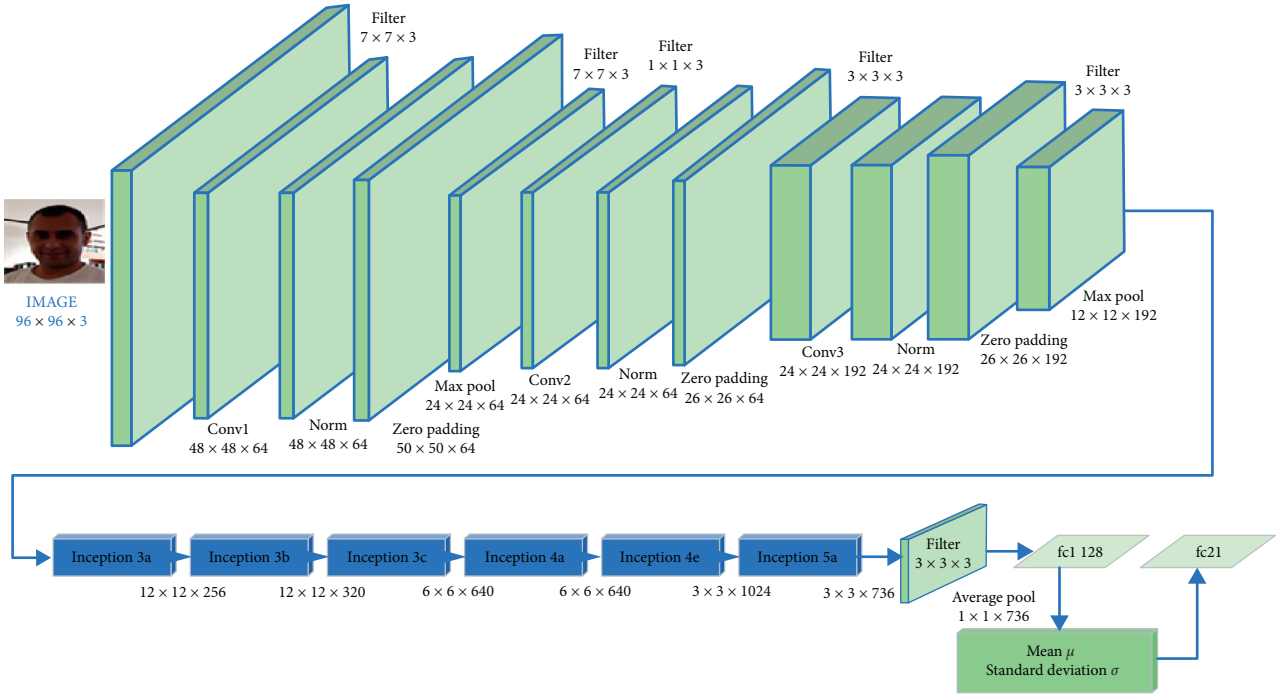


Figure 2: The structure of the model diagram.

have been kept in another model file called acknowledge base.

3.3. Image Aligning with Face Reidentification. Face detection and recognition still have many problems to identify the face especially when the face is aligned to down or to any other angle in the image. This problem can be solved by searching for a face in the image. If the image does not have a face, we will rotate the image step by step from 0 to 360°, where each step is rotated for 14°, until we find a face in that image and pass it as a new image. Therefore, the total number of steps is 25. In case we could not find a face in the image after rotating it, we will pass the image without

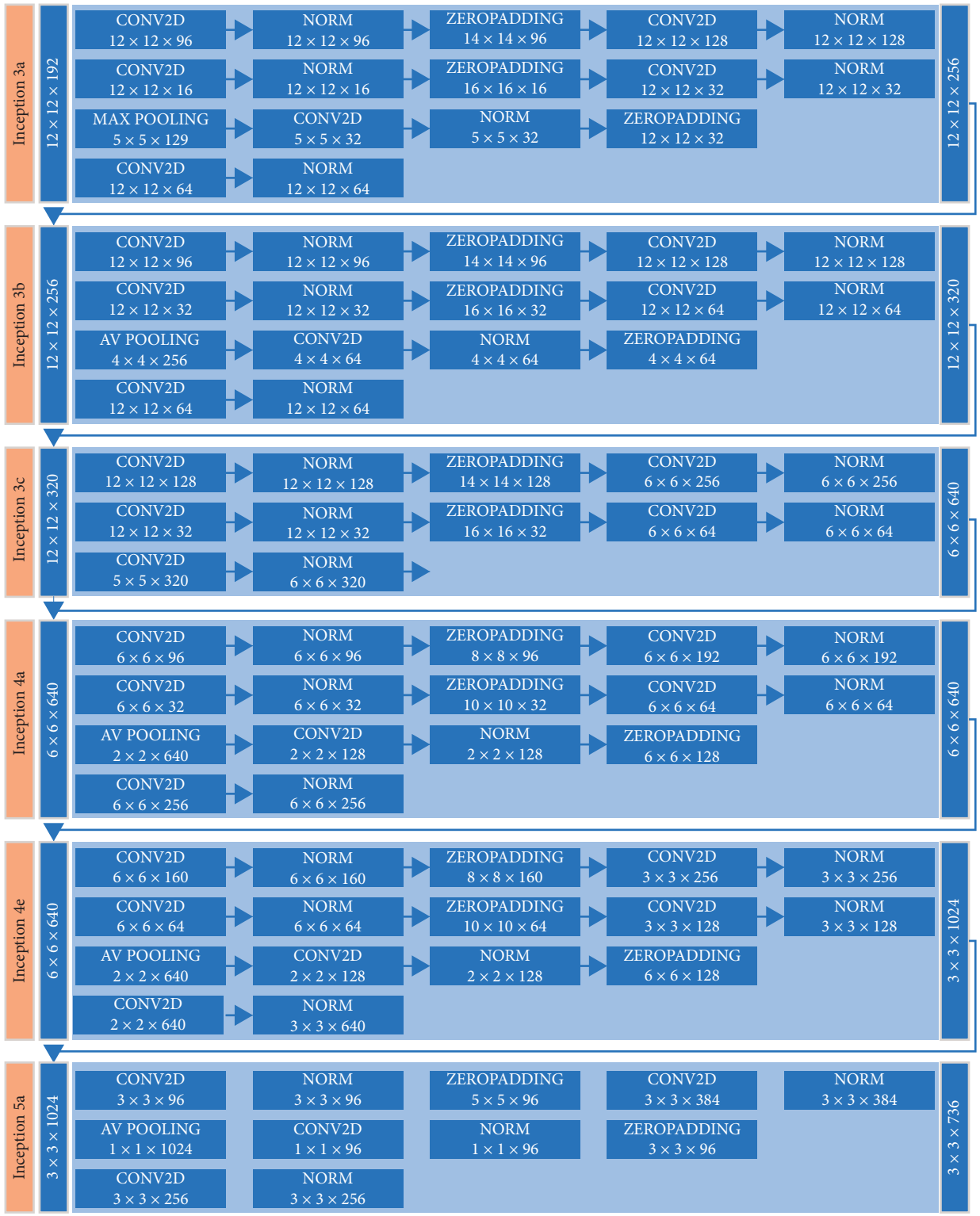


Figure 3: The structure of the inceptions used in the model.

rotating because there may be a face in the image where the face is in different poses and cannot be detected. Figure 4 shows an image with a face that the face detector cannot detect it, but after the rotations, we find a face while Figure 5 shows a face that cannot be detected after 360 rotations, so the original image will not be changed.

4. Evaluation

We used a neural network to extract the features of the faces. Feature extraction takes 128 bytes for each face and then finds the weights greater than zero from the original image with their corresponding indices and finds the other weights

Figure 4: The first image is the original image. The detector could not detect the face, so we rotated it. After two steps of rotations (rotate 1 and rotate 2), the detector still could not detect the face. After 16 steps of rotations, the detector detected the face successfully; therefore, the original image will be the image in (rotate 16). The identity of the face in rotate 16 has been recognized effectively by the system.

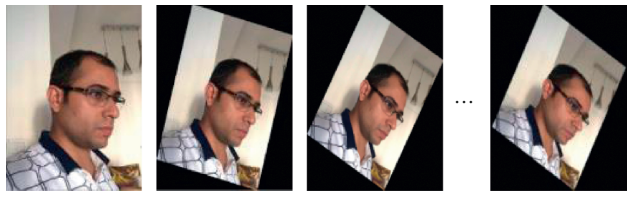


Figure 5: The first image is the original image. The detector could not detect the face, so we rotated it. After two steps of rotations (rotate 1, rotate 2), the detector still could not detect the face. After 25 steps of rotations (rotate 25), the detector still could not detect the face, and therefore, the original image will remain the first image. The identity of the face in rotate 16 has been recognized effectively by the system.

of the identities with the corresponding indices of the original image. The process of selecting weights larger than zero with their corresponding indices is called a filter process where the dimensional of the vector will be reduced to half of 128 bytes. After that, the distance of the filtered bytes of the original image with the bytes of each identity in the same indices of the original image is calculated to find the minimum number. The minimum number will refer to the identity of the original image. We evaluated the network on the Labeled Faces in the Wild and YTF [26] datasets.These two datasets have been used in most previous works which got a state-of-the-art results in their evaluation process. We achieved good results on these two datasets.

In the evaluation process, we extracted the features of each image in the dataset where each image has 128 features and stored them in a separate file. Then, we divided the weights into blocks by dividing the total weights by 128 to find the number of identities ind as in (7). Each block will contain 128 weights and will be treated as a single block for a single identity:

Table 2: Face verification of different methods on LFW.

Method	Accuracy (%)
DeepFace [17]	97.35
FaceNet [14]	99.63
High-dim LBP [27]	95.17
TL Joint Bayesian [28]	96.33
GaussianFace [29, 30]	98.52
DeepID [5]	97.45
DeepID2 [6]	99.15
DeepID2+ [10]	99.47
DeepID3 [30]	99.53
Our method	**99.70**

Table 3: Face verification of different methods on YTF.

Method	Accuracy (%)
Face reidentification [12]	90.41
DeepFace [17]	91.4
Face representation [11]	92.24
Deeply learned face [10]	93.2
FaceNet [14]	95.12
Our method	**94.02**

$$ind = \text{total} \times \frac{\text{weights}}{128}. \tag{7}$$

For the original image that we want to identify it, we extracted its 128 features using our model and passed these features to the filter to find the most important features for representation and reduce the dimensionality to the half. After choosing the positive values of each feature of the original image and taking their corresponding indexes, we will extract the features of each block from the features of the dataset according to the indexes of the positive features of the original image as in the following equation:

$$\text{TW} = \sum_{i1=0}^{128} W_{i\,dn}^{o_{i2}}, \quad \text{if } o_{i2} = i\,dn_i, \tag{8}$$

where o_{i2} is the indexes of the filter weights of the original image and the values of $i2$ are indexed from 0 to half of 128. W is the weight of the identity $i\,dn$. Finally, we applied (6) to identify the image.

5. Experiments

5.1. Dataset. We have used Celeb Faces Attributes Dataset (CelebA) as the training faces in our training. It consists of 10,177 identities and 202,599 faces. Before training, we extract the face of each image in the dataset using a face detector and then we resized it to the input size of our neural networks which is $96 \times 96 \times 3$ pixels.

LFW and YTF datasets used in the evaluation process. LFW is a database of face photographs for studying the problem of unconstrained face recognition. This database contains 13,233 images of 5,749 people detected and centered by the Viola–Jones face detector and collected from the web. YTF is a database of face videos designed to be used for

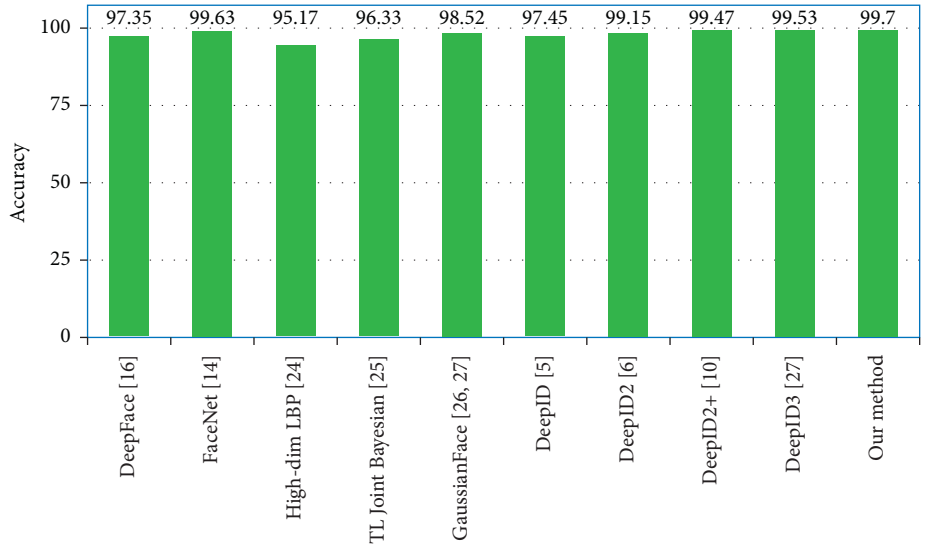


Figure 6: Chart illustrates the classification accuracy for different methods on LFW.

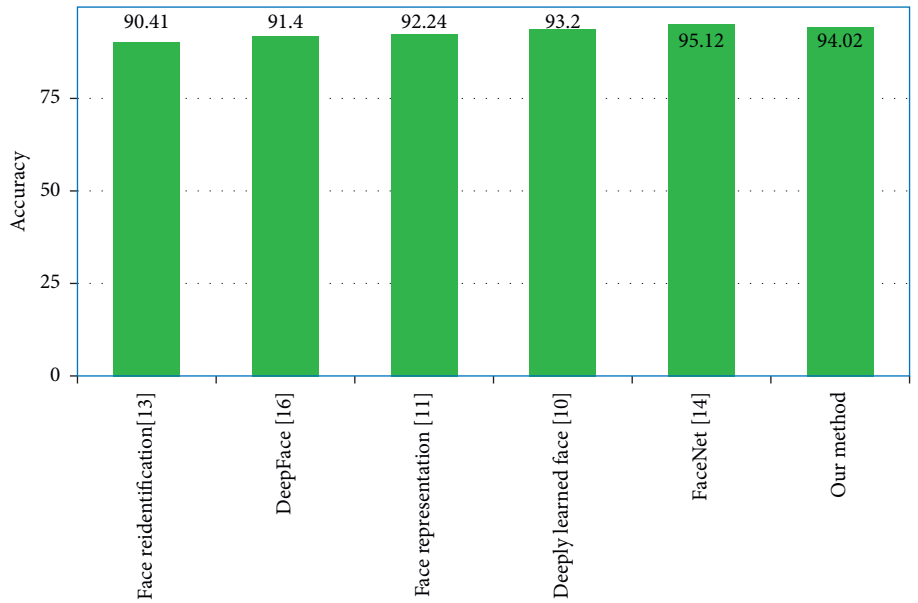


Figure 7: Chart illustrates the classification accuracy for different methods on YTF.

studying the problem of unconstrained face recognition in videos. This dataset contains 3,425 videos of 1,595 different people downloaded from YouTube. The shortest video contains 48 frames while the longest video contains 6,070 frames.

5.2. Various Dimensionalities. Various embedding dimensionalities were explored in previous studies [14], and accordingly, the dimension 128 has been selected as it gives the best accuracy. The comparison between four embedding dimensionalities, 64, 128, 256, and 512, shows that the difference in the performance is small. In this study, we explored the best dimensionality, i.e., 128, before and after applying the filter. After applying the filter to the dimension of 128, the dimensionality has reduced to the half of 128 with higher accuracy of the dimension of 128 by using our new algorithm.

5.3. Acknowledge Base Identities. In order to increase the number of identities without looking at the picture of any identity in the dataset again, acknowledge base model has been created to save the features of each identity. The features of any new identity will be saved in the acknowledge base model. This acknowledge base model will be used to know any other unseen face picture to predict the identity.

5.4. Effect of Face Detection. Most of the face detection frameworks have shown good results in face detection, whereas

there are still some limitations. Many faces have not been detected incorrectly using the most widely framework used for face detection. This limitation can affect negatively the results. Therefore, face detection still needs some improvements.

5.5. Performance on LFW and YTF. During the evaluation, the feature of every identity is extracted and kept in the acknowledge base. Any other extraction for any identity will be added to the acknowledge base with its corresponding label of that identity. Every time in the evaluation step, we took 200k images for test and kept their features with their corresponding labels in the acknowledge base with any previous features extracted for any identity. That means the acknowledge base model can store the features of all images in the dataset and it can find the single identity of any face among all these identities. We achieved a classification accuracy of 99.70% on the LFW dataset and 94.02% on the YTF dataset. Table 2 and Table 3 show the classification accuracy with some methods as compared to our classification accuracy on LFW and YTF. Figures 6 and 7 demonstrate a comparison chart for previous studies with LFW and YTF.

6. Conclusions

Deep neural network is used in this paper for face reidentification. The filter technique is used to select the most important features from the features extracted by the model. This method can identify the face in different poses and different levels of illumination. The rotation technique for 360° is used for the images that have the face in different angles, while this kind of rotation cannot be done in the augmentation method in deep learning.

We noticed that deep learning is very important to extract the features, but with well-prepared mathematical operations on the extracted features from the deep learning, it can increase the accuracy of the model.

Data Availability

The model, extraction of weight code, features saved in acknowledge base, and the equations for evaluation code are available in the following URL: https://drive.google.com/open?id=1pXMkhAOx9zV4n8ynmer2xlF5lLeQZ3Rz.

Disclosure

The funding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; and in the decision to publish the results.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

Mohammed Al-Ghaili performed programming and wrote the manuscript. Zhiyong Li supervised the study. Hamdi A.R. Ali proofread the article.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (nos. 61672215 and 61976086), National Key R&D Program of China (no. 2018YFB1308604), and Hunan Science and Technology Innovation Project (no. 2017XK2102).

References

[1] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Deep learning identity-preserving face space," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1489–1496, IEEE, Sydney, NSW, Australia, December 2013.

[2] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Hybrid deep learning for face verification," in *Proceedings of the International Conference on Computer Vision*, pp. 113–120, IEEE, Sydney, NSW, Australia, December 2013.

[3] W. Liu, Y. Wen, Z. Yu et al., "Sphereface: deep hypersphere embedding for face recognition," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[4] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Recover canonical-view faces in the wild with deep neural networks," pp. 1404–3543, 2014, https://arxiv.org/abs/1404.3543.

[5] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 1891–1898, IEEE, Columbus, OH, USA, June 2014.

[6] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," *Advances in Neural Information Processing Systems*, pp. 1988–1996, 2014, https://arxiv.org/abs/1406.4773.

[7] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, "Web-scale training for face identification," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 2746–2754, IEEE, Boston, MA, USA, June 2015.

[8] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Deep learning and disentangling face representation by multi-view perceptron," in *Proceedings of the NIPS*, Quebec, Canada, December 2015.

[9] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbour classification," *Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.

[10] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 2892–2900, IEEE, Boston, MA, USA, June 2015.

[11] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," pp. 1411–7923, 2014, https://arxiv.org/abs/1411.7923.

[12] Y. Wang, J. Shen, S. Petridis, and M. Pantic, "A real-time and unsupervised face re-identification system for human-robot interaction," *Pattern Recognition Letters*, vol. 128, 2018.

[13] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the International Conference on Computer Vision*, pp. 3730–3738, IEEE, Santiago, Chile, December 2015.

[14] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: a unified embedding for face recognition and clustering," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 815–823, IEEE, Boston, MA, USA, June 2015.

[15] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: a general-purpose face recognition library with mobile applications," Tech. Rep. CMU-CS-16-118, CMU School of Computer Science, Pittsburgh, PA, USA, 2016.

[16] G. B. Huang, H. Lee, and E. Learned-Miller, "Learning hierarchical representations for face verification with convolutional deep belief networks," in *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, pp. 2518–2525, IEEE, Providence, RI, USA, June 2012.

[17] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, "Deepface: closing the gap to human-level performance in face verification," in *Proceedings of the conference on computer vision and pattern recognition*, pp. 1701–1708, IEEE, Columbus, OH, USA, June 2014.

[18] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: a database for studying face recognition in unconstrained environments," in *Proceedings of the Workshop on Faces in "Real-Life" Images: Detection, Alignment, and Recognition*, University of Massachusetts, Amherst, MA, USA, October 2018.

[19] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proceedings of the BMVC*, vol. 6, University of Oxford, Oxford, UK, September 2015.

[20] D. Kim, M. Hernandez, J. Choi, and G. Medioni, "Deep 3D face identification," in *Proceedings of the International Joint Conference on Biometrics (IJCB)*, pp. 1703–10714, IEEE, Denver, CO, USA, October 2017.

[21] A. Savran, N. Alyüz, H. Dibeklioğlu et al., "Bosphorus database for 3D face analysis," in *Proceedings of the European Workshop On Biometrics and Identity Management*, pp. 47–56, Springer, Roskilde, Denmark, May 2008, Lecture Notes in Computer Science.

[22] V. Vijayan, K. W. Bowyer, P. J. Flynn et al., "Twins 3D face recognition challenge," in *Proceedings of the International Joint Conference on Biometrics (IJCB)*, pp. 1–7, IEEE, Washington, DC, USA, October 2011.

[23] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 1–9, IEEE, Boston, MA, USA, June 2015.

[24] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013, https://arxiv.org/abs/1312.6114ICLR.

[25] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," *Clinical Orthopaedics and Related Research*, https://arxiv.org/abs/1412.6980, 2014.

[26] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 529–534, IEEE, Providence, RI, USA, June 2011.

[27] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: high-dimensional feature and its efficient compression for face verification," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 3025–3032, IEEE, Portland, OR, USA, June 2013.

[28] X. Cao, D. Wipf, F. Wen, G. Duan, and J. Sun, "A practical transfer learning algorithm for face verification," in *Proceedings of the International Conference on Computer Vision*, pp. 3208–3215, IEEE, Sydney, NSW, Australia, December 2013.

[29] C. Lu and X. Tang, "Surpassing human-level face verification performance on LFW with GaussianFace," *Association for the Advancement of Artificial Intelligence*, pp. 3811–3819, 2014, https://arxiv.org/abs/1404.3840.

[30] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: face recognition with very deep neural networks," pp. 3811–3819, 2015, https://arxiv.org/abs/1502.00873.

Hindawi
Scientific Programming
Volume 2020, Article ID 4748606, 13 pages
https://doi.org/10.1155/2020/4748606

Research Article

Data-Driven Decision-Support System for Speaker Identification Using E-Vector System

He Ma,[1] Yi Zuo ,[1,2,3,4] Tieshan Li,[1,2] and C. L. Philip Chen[1,2]

[1]*Navigation College, Dalian Maritime University, Dalian 116026, China*
[2]*Maritime Big Data & Artificial Intelligent Application Center, Dalian Maritime University, Dalian 116026, China*
[3]*Collaborative Innovation Center for Transport Studies, Dalian Maritime University, Dalian 116026, China*
[4]*The Research Institute for Socionetwork Strategies, Kansai University, Osaka 5648680, Japan*

Correspondence should be addressed to Yi Zuo; zuo@dlmu.edu.cn

Received 12 November 2019; Revised 24 February 2020; Accepted 12 June 2020; Published 29 June 2020

Academic Editor: Rahman Ali

Recently, biometric authorizations using fingerprint, voiceprint, and facial features have garnered considerable attention from the public with the development of recognition techniques and popularization of the smartphone. Among such biometrics, voiceprint has a personal identity as high as that of fingerprint and also uses a noncontact mode to recognize similar faces. Speech signal-processing is one of the keys to accuracy in voice recognition. Most voice-identification systems still employ the mel-scale frequency cepstrum coefficient (MFCC) as the key vocal feature. The quality and accuracy of the MFCC are dependent on the prepared phrase, which belongs to text-dependent speaker identification. In contrast, several new features, such as d-vector, provide a black-box process in vocal feature learning. To address these aspects, a novel data-driven approach for vocal feature extraction based on a decision-support system (DSS) is proposed in this study. Each speech signal can be transformed into a vector representing the vocal features using this DSS. The establishment of this DSS involves three steps: (i) voice data preprocessing, (ii) hierarchical cluster analysis for the inverse discrete cosine transform cepstrum coefficient, and (iii) learning the E-vector through minimization of the Euclidean metric. We compare experiments to verify the E-vectors extracted by this DSS with other vocal features measures and apply them to both text-dependent and text-independent datasets. In the experiments containing one utterance of each speaker, the average accuracy of the E-vector is improved by approximately 1.5% over the MFCC. In the experiments containing multiple utterances of each speaker, the average micro-F1 score of the E-vector is also improved by approximately 2.1% over the MFCC. The results of the E-vector show remarkable advantages when applied to both the Texas Instruments/Massachusetts Institute of Technology corpus and LibriSpeech corpus. These improvements of the E-vector contribute to the capabilities of speaker identification and also enhance its usability for more real-world identification tasks.

1. Introduction

Over the last decades, recognition technologies based on biometrics such as fingerprint, facial features, voiceprint, and iris scans have been widely used in target identification for access security, system identity, private confirmation, etc. In terms of technical and practical usage, recognition through iris scans is the most secured and accurate and is applied to meet the requirements of military standards [1]. For mass requirements, fingerprint recognition is one of the most popular and mature identity-recognition technologies [2]. As fingerprint acquisition and recognition need specific devices [3], it has been increasingly replaced by face recognition in recent years, often preferred for its noncontact mode [4]. Facial data can be collected more easily than iris and fingerprint data, as most smartphones already have an inbuilt camera. However, the accuracy of face recognition is dependent on the recognition conditions, such as environmental brightness and camera angle [5]. Similar to face recognition, voice recognition is also a noncontact mode technique. The voiceprint can be easily collected using a microphone and other voice receivers, and its quality

requirements are less dependent on environmental factors than those of face recognition [6]. Similar to fingerprints, voiceprints also contain unique biometric features and are superior to facial features on recognition accuracy. Nevertheless, voice recognition has many merits compared with other recognition technologies. Feature extraction from voiceprint data is the main technical bottleneck, and its real-world applications are fewer than those of fingerprint and face recognition in daily life. In contrast to face recognition, which employs image processing methods, a voiceprint is composed of classic mechanical waves that need signal processing methods to transform voice signals from time-domain representation into frequency-domain representation. Such vocal features are difficult to implement, but quite efficient for speaker identification owing to the biometric variances and personal characteristics between different voiceprints. In most methods of speaker identification (SI), there are two main processes: one is to extract the vocal features, and the other is to learn the identification model based on these features. The vocal features not only adequately represent the common properties of the same speaker, but also separate the different speakers as far apart as possible. Therefore, an effective extraction of vocal features can determine the performance of SI models, such that these models can definitely identify the target speaker from multiple speakers' utterances.

The general process of SI can be regarded as a decision-making support process that decides the identity of the corresponding speakers by their utterances. In the field of automatic speech recognition (ASR), most methods of SI are constructed by extracting the vocal features, which is also one of the most important applications of the decision-support system (DSS) for SI tasks. The linear prediction coefficient (LPC) was extracted by a linear combination of the exiting speech, which was the first proposed speech feature in 1967 [7]. Since the vocal feature named mel-scale frequency cepstrum coefficient (MFCC) was proposed in 1980 [8], it has been extensively applied in SI systems. The perceptive linear prediction coefficient was extracted by putting speech signals into the auditory model based on LPC in 2011 [9]. In 2012, a histogram frequency-domain transformation on the discrete cosine transform (DCT) cepstrum coefficient (HDCC) was carried out based on the idea of the MFCC feature extraction [10]. Subsequently, Kim and Stern applied a power-law nonlinear transformation instead of the traditional log nonlinear transformation of MFCC by auditory processing, and they proposed a new feature called power-normalized cepstral coefficients (PNCC) in 2016 [11]. In comparison with the traditional vocal features, an SI model based on the identity vector (i-vector) was also proposed in a data-driven approach [12], which is a popular topic in the field of ASR. Furthermore, Variani et al. applied a deep neural network to generate the d-vector, which is a similar feature to the i-vector [13]. Based on this d-vector, an end-to-end SI approach was also proposed [14].

The existing methods for the extraction of vocal features mostly used model-based approaches, such as MFCC, PNCC, and LPC. In contrast, several new vocal features such as d-vector are based on data-driven approaches. However, these approaches are "black box" in vocal feature learning [14]. Consequently, in this study, a novel method using the data-driven approach of hierarchical cluster analysis for SI is proposed. There are three main contributions: (1) a novel vocal feature extraction method is proposed based on a data-driven approach of hierarchical cluster analysis; (2) the Euclidean metric is used as a measure to generate an adaptive feature vector called the "E-vector"; (3) DSS is established based on the E-vector to provide decision-making support services for SI tasks. In the data-driven hierarchical clustering approach, various personal phonetic features are considered to learn and extract the vocal feature vector. The distances between different cepstral coefficients of the same speaker are measured using the Euclidean metric, and the E-vector is generated through a hierarchical clustering approach by minimizing the Euclidean metric. In the comparative experiments of single utterance SI, the E-vector method improves the identification accuracy by approximately 3% over the MFCC and 5% over the HDCC, where the DSS of SI is based on the Gaussian mixture model (GMM). In the comparative experiments of multiple utterances SI, the micro-F1 score of the E-vector is better than the MFCC and HDCC, where the DSS of SI is based on both the GMM and hidden Markov model (HMM).

The remainder of this paper is organized as follows: the problem statement and E-vector are introduced in Section 2. The conducted comparative experiments to evaluate the performance of the E-vector are described in Section 3. Finally, a conclusion is provided in Section 4.

2. Materials and Methods

2.1. Problem Statement

2.1.1. Model-Based Extraction of Vocal Features. The existing vocal features used in SI are mostly based on model-based approaches, such as MFCC, HDCC, and PNCC. MFCC is a widely used speech feature first proposed in the 1980s. MFCC applies a discrete Fourier transform method to transform the time-domain signal into a frequency-domain signal. In an MFCC transformation, we use the following equation to translate the frequency-domain signal into mel-frequency:

$$\mathrm{mel}(f) = 2595 * \ln\left(1 + \frac{f}{700}\right), \tag{1}$$

where f is the original frequency, and $\mathrm{mel}(f)$ represents mel-frequency. Subsequently, the amplitude based on mel-frequency is calculated by a series of triangular filters, as in Figure 1(a). Finally, MFCC is obtained by making a cepstrum analysis of the signal using the mel-frequency and triangular filters [8]. HDCC is a new feature proposed with the influence of MFCC. The HDCC creates a two-term span of histogram bins: 50–500 Hz with a span of 50 Hz each and 600–1000 Hz with a 100 Hz span of each as shown in Figure 1(b). After DCT cepstrum coefficients of each bin are obtained from histogram analysis, we can extract the HDCC for each bin [10]. PNCC has similar parts of the first two steps of MFCC in its initial process. Next, PNCC obtains the

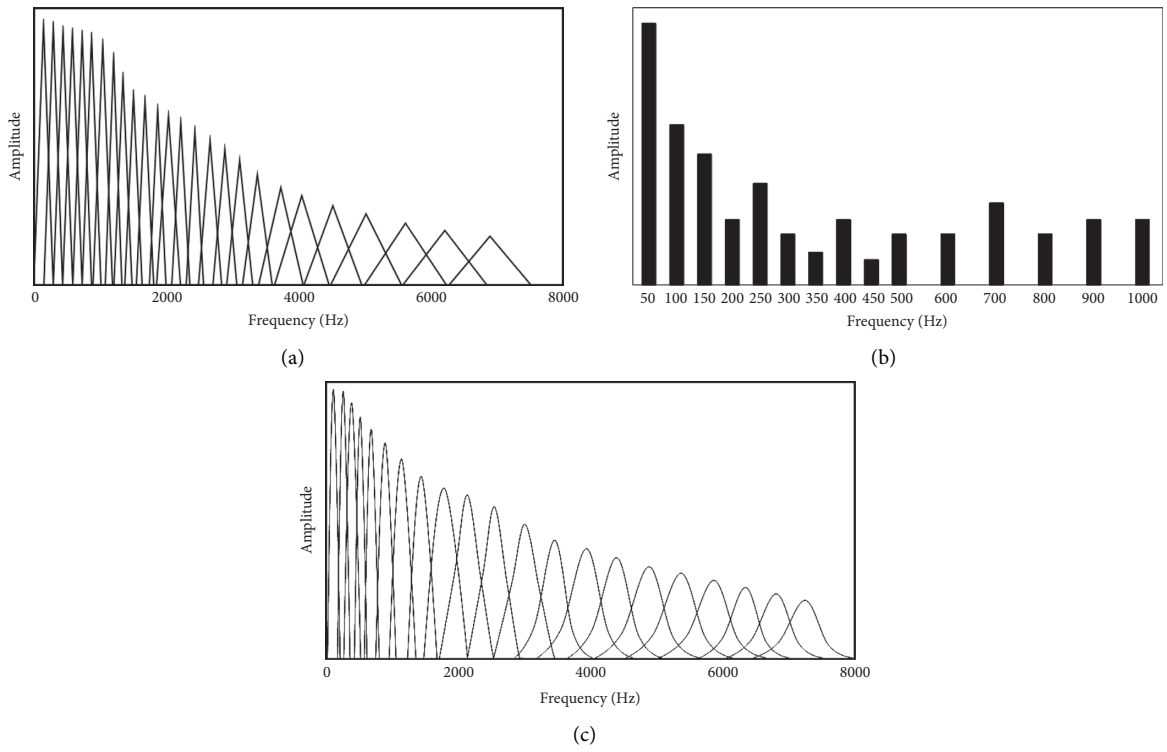


Figure 1: (a) MFCC filters. (b) HDCC filters. (c) PNCC filters.

short-time spectral power using the squared gammatone summation. As shown in Figure 1(c), gammatone filters are power-law nonlinear transformations, different from the traditional log nonlinearity used in the MFCC. Finally, smoothing-weight processing is used on each frame, and spectral subtraction is applied to realize noise suppression [11].

2.1.2. Data-Driven Extraction of Vocal Features. The existing research using data-driven approaches for SI mostly focused on clustering different speakers via their feature similarity. For instance, the Alibaba group proposed a speech recognition method based on a clustering method in 2017 [15]. They obtained the feature vector based on cluster analysis of training data. Then, the feature vector model was established for speech recognition [15]. Nevertheless, there are few researches on applying data-driven approach to extract vocal feature. The i-vector, d-vector, and end-to-end SI approach were proposed based on data-driven approach; however, they are black box (method without a transparent working process) [12–14]. Actually, vocal data has relevant regularities accounting for the speaker's personal phonetic features. Therefore, in this paper, a novel vocal feature, E-vector, extracted using a data-driven approach is proposed. The method learns the SI models based on the E-vector realized as a DSS.

2.2. Determining the Decision Objective. In decision-making process, there are generally four steps, as shown in Figure 2 (a). At the beginning, decision objective (DO) should be determined after finding out the problem. Then, the scheme will be designed based on the decision environment. Next, the scheme will be evaluated in order to carry the scheme. In an SI task, the SI process can be regarded as a multilabel classification task. The number of speakers is the number of classes; the labels are the utterance of each speaker. The DO is achieving the classification of all speakers by identifying all speakers' identities based on vocal features, as shown in Figure 2 (b).

2.3. E-Vector System for Speaker Identification. In this section, we introduce the E-vector system for SI-DSS. It is shown in Figure 3(a) that the SI-DSS based on E-vector system is established in three steps: (i) data preprocessing, (ii) cluster analysis, and (iii) learning models. When a continuous speech signal is put into the E-vector system, data preprocessing is applied to obtain the inverse discrete cosine transform (IDCT) cepstrum coefficient; then the clustering method is used to analyze the IDCT cepstrum coefficient, and finally, GMM and HMM are applied to classify the speakers. The following charts show the detailed introduction of the E-vector system.

2.3.1. Step 1: Data Preprocessing. The competency of data preprocessing is storing speech data in the form of the IDCT cepstrum coefficient. The original speech data is in the form of a continual signal wave, and the spectrogram is generally used to describe the continual signal wave. In this study, the spectrogram is extracted by the following three steps:

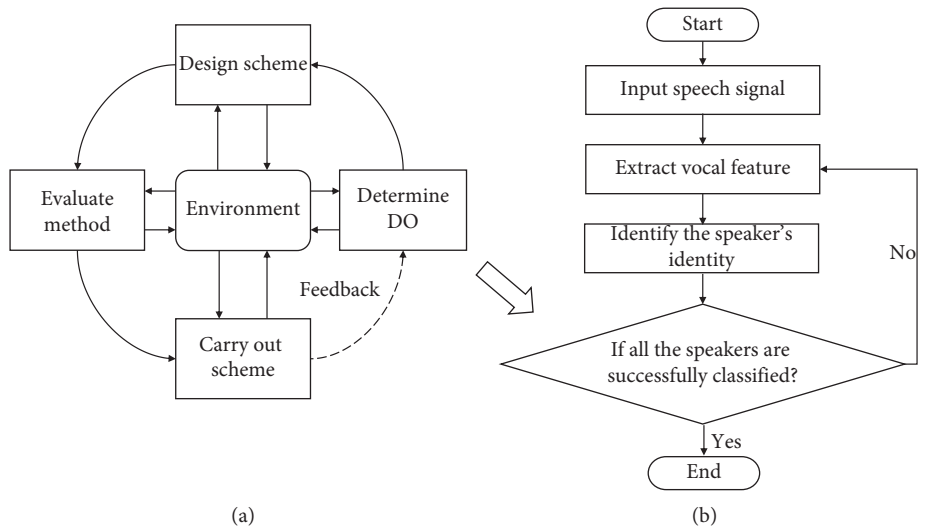


Figure 2: The relationship between DSS and SI. (a) DSS. (b) The SI system based on DSS.

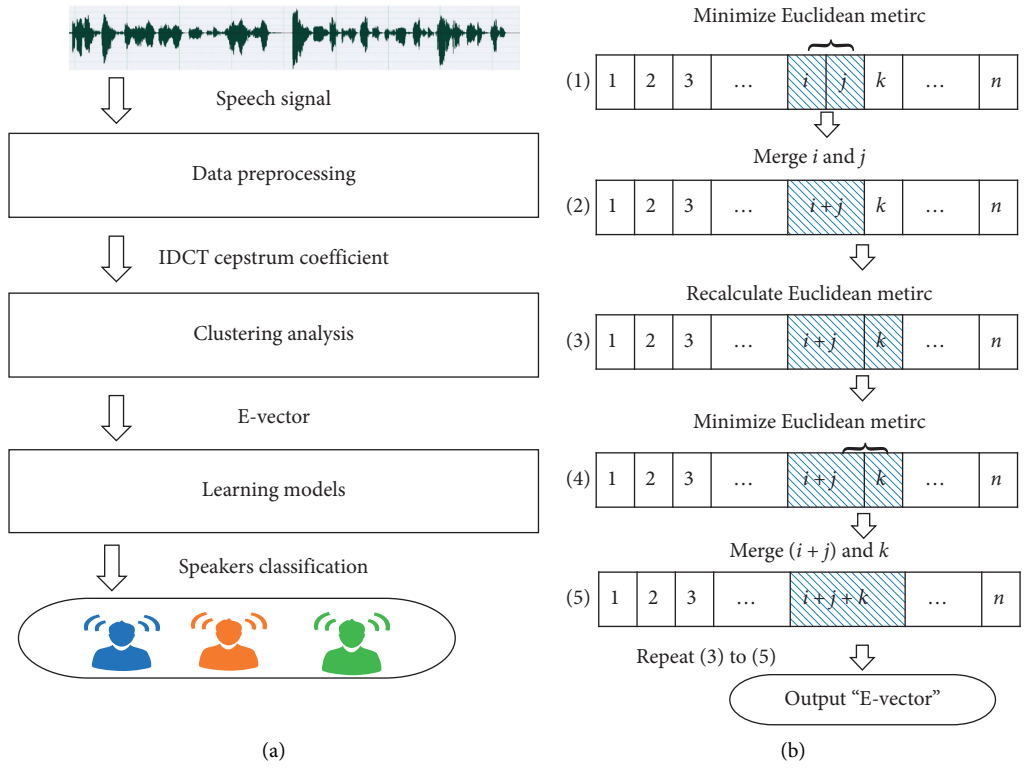
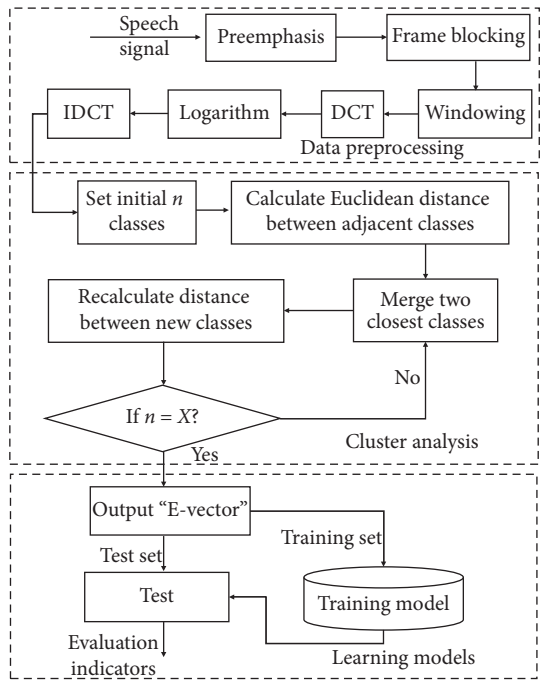


Figure 3: Continued.



(c)

Figure 3: Overview of E-vector system for SI-DSS. (a) Flowchart of SI-DSS. (b) Flowchart of E-vector procedure. (c) Flowchart of SI system.

(1) The first step aims at making the speech signal wave more significant. High-pass filtering process shown in the following equation is used to preemphasize the input signal wave [7]:

$$H(z) = 1 - \mu z^{-1}. \tag{2}$$

Here, z is the input speech signal, H (z) is the output preemphasis speech signal, and the value of μ is 0.97 in this study.

(2) In the second step, the preweighted speech signal is segmented into small blocks to get a frame signal (frames of 20 ms in this study).

(3) The third step is adding a Hamming window, W, to the framed signal. The Hamming window function is defined as

$$W(n, 0.46) = 0.54 - a \times \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \ll N-1, \tag{3}$$

where N is the number of each frame. It can make the voice data more periodic for analyzing each frame signal.

Then, homomorphic signal processing is applied to obtain the IDCT cepstrum coefficient. The processing involves three steps:

(1) In the first step, the DCT is applied to obtain the multiplicative signal as equation (4) from all frames of the speech signal. The process of DCT is defined as

$$C(a) = \sum_{m=0}^{N-1} S(b)\cos\left(\frac{\pi n(b-0.5)}{M}\right), \quad a = 1, 2, \ldots, L. \tag{4}$$

Here, $S(b)$ is the input speech signal and M is its points; $C(a)$ is the output signal, and a is the points of transformation:

$$C(a) = S_h(b) \times S_l(b). \tag{5}$$

Here, $S_h(b)$ is the high-frequency signal; $S_l(b)$ represents the low-frequency signal.

(2) The second step is calculating the logarithmic energy of the output signal to convert the multiplicative signal into an additive signal as follows:

$$\log C(a) = \log\left(S_h(b)\right) + \log\left(S_l(b)\right). \tag{6}$$

(3) The third step is applying the IDCT to obtain the cepstrum coefficient as follows:

$$c(a) = s_h(b) \times s_l(b). \tag{7}$$

Here, $c(a)$ is the IDCT cepstrum coefficient, $s_h(b)$ is the output high-frequency signal, and $s_l(b)$ is the output low-frequency signal.

2.3.2. Step 2: Cluster Analysis for IDCT Cepstrum Coefficient. The obtained IDCT cepstrum coefficient is a data matrix, and the length of a row is proportional to the time duration of the input vocal signal, and the length of a column is proportional to the number of the speakers in the input signal. The analysis process of the IDCT cepstrum coefficients consists of five steps:

(1) If the input vocal signal contains m speakers' speech, the IDCT cepstrum coefficient can be described by set A. The speech of speaker p ($p = 1, 2, \ldots, m$) can be described as in (9), where n is the number of A's columns. For such a data matrix, the cluster method can be applied to analyzing using the data-driven approach. An improved hierarchical cluster method is proposed to analyze the IDCT cepstrum coefficient as only adjacent columns can be grouped:

$$A = \{A_1; A_2; \ldots; A_m\}, \tag{8}$$

$$A_p = \{a_1^P, a_2^P, \ldots, a_n^P\}. \tag{9}$$

(2) Each column of set A is regarded as a class, so there are n classes, as shown in Figure 3(b).

(3) Calculate the distances of adjacent classes and define the similarity values as a set. Here, the Euclidean distance measure [16] is used to calculate the distance. The smaller the value of the distance, the greater the similarity. The Euclidean distance l_i ($l_i \in S$) of a_i and a_{i+1} can be described as in (10). Thus, the set S can be composed of $n - 1$ number of distance values. It can be described as in (11):

$$l_i = \mathrm{Dis}(a_i, a_{i+1}) = \sqrt{\sum_{j=1}^{mn} (a_{i,j} - a_{i+1,j})}, \tag{10}$$

$$S = \{l_1, l_2, \ldots, l_{n-1}\}. \tag{11}$$

(4) Compare all values in S; if the Euclidean distance l_i of a_i and a_{i+1} is the minimum one, group a_i and a_{i+1} into a class. Update the classes of set A.

(5) Iterate step (2) to step (4) until the number n of classes in set A is equal to X (X is determined by identification accuracy) as shown as Figure 3(c). Then, the classes in set A constitute the E-vector.

2.3.3. Step 3: Learning SI Models. The identification process matches the input feature with the model feature set by the degree of similarity. In this study, the model feature set is established using the GMM and HMM based on the E-vector feature Algorithm 1. The HMM achieves the identification task by searching for the sequence most likely to produce a particular output sequence in the implicit state; the process consists of six steps:

(1) Define a vocal feature set $A = \{a_1, a_2, \ldots, a_X\}$ for the model, where a_X is the number X class of the vocal feature set A.

(2) Accumulate certain vocal features with their labels in each class a_i in the vocal feature set A to establish the training feature set.

(3) Obtain the best model λ_i for a_i, based on the training set, as shown in Figure 3(c).

(4) Produce an unknown observation sequence O for input feature.

(5) Estimate the probability of the input feature $\Pr(O/\lambda_i)(i = 1, 2, \ldots, X)$. The input feature belongs to the class with the maximal probability.

(6) Calculate evaluation indicators, and the value of X is determined by the state with the highest accuracy.

3. Results and Discussion

3.1. Experimental Design

3.1.1. Datasets of Vocal Corpus. In the experiments, we used two vocal datasets. One was the Texas Instruments/Massachusetts Institute of Technology (TIMIT) corpus, and the other one was the LibriSpeech corpus. The TIMIT corpus was applied as the representative of a text-dependent experiment, which contained 6300 sentences spoken by 630 persons [17]. The LibriSpeech corpus is a variety of audio datasets that consists of text and voice. Thus, the LibriSpeech corpus was used as the representative of a text-independent experiment [18]. Two groups of experiments were conducted with the TIMIT corpus and LibriSpeech corpus. Table 1 shows the number of speakers used in the experiments with the TIMIT and LibriSpeech corpus.

3.1.2. Evaluation Indicators. In the research of identity identification, several evaluation indicators were applied to evaluate the algorithm's performance. The false rejection rate (FRR) is the proportion of cases mistaking the matched voiceprint as the unmatched voiceprint. It refers to the proportion of cases in which the same voiceprint is mistakenly considered as a different voiceprint when testing the voiceprint recognition on the standard voiceprint database:

$$\mathrm{FRR} = 1 - \mathrm{recall}. \tag{12}$$

In this study, we applied the measures accuracy, precision, recall, and micro-F1 score to evaluate the performance of the E-vector against other features. If the number of speakers is m, TP_i is the true positive number of "i" ($0 \le i \le m$) speakers, FP_i is the false positive number of "i," TN_i is the true negative number of "i," and FN_i is the false negative number of "i."

The accuracy is calculated as follows:

$$\mathrm{accuracy} = \frac{\sum_{i=0}^{m} (\mathrm{TP}_i + \mathrm{TN}_i)}{\sum_{i=0}^{m} (\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{TN}_i + \mathrm{FN}_i)}. \tag{13}$$

Input: z (continuous speech signal); frame 20 ms; Step 10 ms; $n = X$
Output: E-vector $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_\mathbf{X}\}$; accuracy; micro-F1
(1) Initialization: $c\ (a) \longrightarrow$IDCT cepstrum coefficient
(2) Data preprocessing: $c\ (a) =$ data preprocessing (z)
(3) Cluster: Set S (S$\leftarrow\varnothing$), Name $c\ (a) \longrightarrow \mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_\mathbf{i}, \ldots, \mathbf{a}_\mathbf{n})$.
(4) **For** each $\mathbf{a}_\mathbf{i}$ in A do
(5) $\quad \mathbf{l}_\mathbf{i} = \mathbf{Dis}(\mathbf{a}_\mathbf{i}, \mathbf{a}_{\mathbf{i+1}})$
(6) $\quad$ Put all $\mathbf{l}_\mathbf{i}$ into S
(7) $\quad$ **If** $\mathbf{l}_\mathbf{i} = \mathbf{arcmin}(\mathbf{S})$
(8) $\quad \mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_\mathbf{i} + \mathbf{a}_{\mathbf{i+1}}, \mathbf{a}_{\mathbf{i+2}} \ldots, \mathbf{a}_\mathbf{n})$.
(9) $\quad\quad$ **If** $n = X$
(10) $\quad\quad\quad$ break;
(11) $\quad\quad$ **else**
(12) $\quad\quad\quad$ continue;
(13) $\quad\quad$ **end**
(14) $\quad$ **end**
(15) **end**
(16) Learn models: put A into GMM, HMM $\longrightarrow$ accuracy, micro-F1

Algorithm 1: E-vector system for speaker identification.

Table 1: Datasets of TIMIT and LibriSpeech corpuses.

	TIMIT				LibriSpeech			
Set no.	T1	T2	T3	T4	L1	L2	L3	L4
Number of speakers	100	300	500	630	10	20	30	40

The precision is calculated as follows:

$$\text{precision} = \frac{\sum_{i=0}^{m} \text{TP}}{\sum_{i=0}^{m} (\text{TP}_i + \text{FP})}. \tag{14}$$

The recall represents the percentage actually true in the positive set, and it is described as follows:

$$\text{recall} = \frac{\sum_{i=0}^{m} \text{TP}_\text{i}}{\sum_{i=0}^{m} (\text{TP}_i + \text{FN}_i)}. \tag{15}$$

The formula of micro-F1 is described as follows:

$$\text{micro} - F1 = 2 \times \frac{recall \times precision}{\text{recall} + \text{precision}}. \tag{16}$$

3.2. Comparison Experiments

3.2.1. Optimization of E-Vector Dimension. In order to decide the optimal dimension of the E-vector based on hierarchical cluster analysis, we selected 630 people of the TIMIT corpus (i.e., T4) and 40 people of the LibriSpeech corpus (i.e., L4) and measured the training accuracy. The proposed E-vector features with different dimensions of 15, 25, and 35 were used with GMM and HMM for the SI task. The results in Table 2 show that 15 dimensions of the E-vector obtain the highest training accuracy. In the experiments with the TIMIT corpus, the voice signals of 630 people were selected for the experiments. The voice signals of 40 people were selected for the experiments with the LibriSpeech corpus. It is shown in Table 2 that the E-vector obtains the same highest accuracy when it consists of 15 and 35 dimensions. We chose the smaller dimension, 15, as the dimension for the E-vector.

Table 2: Identification accuracy comparison.

Model	Set no.	E-vector (15)	E-vector (25)	E-vector (35)
GMM	T4	1.000	0.970	1.000
	L4	1.000	1.000	1.000
HMM	T4	0.930	0.930	0.930
	L4	0.950	0.950	0.950

3.2.2. Single-Utterance Comparison Experiments. We first tested the 15-dimensional E-vector, 13-dimensional MFCC, and 15-dimensional HDCC for SI with an input speech signal containing one utterance of each speaker. The different numbers of speakers in the TIMIT corpus and LibriSpeech corpus are identified using the GMM and HMM. The accuracy results are shown in Table 3. The best performances for each test on each corpus are shown in boldface type.

The following is shown: (1) In the TIMIT corpus, the E-vector performs best with accuracy 1.000 when using the GMM, and the results of MFCC are relatively worse than those of the E-vector; HDCC is inferior to MFCC and E-vector with approximately 10% gap, as shown in Figure 4(a). When the recognition model is HMM, as in Figure 4(c), the results of MFCC and E-vector are both approximately 0.850. It can be found that all these characteristic parameters have good performance with a recognition accuracy of over 0.75. (2) In the LibriSpeech

Table 3: Single-utterance experiment identification accuracy comparison.

Model	Feature	T1	T2	T3	T4	L1	L2	L3	L4
GMM	MFCC	**1.000**	0.990	1.000	0.995	1.000	0.950	0.970	0.975
	HDCC	0.940	0.920	0.926	0.924	**1.000**	**1.000**	0.970	0.950
	E-vector	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
HMM	MFCC	0.930	0.866	**0.874**	**0.860**	**1.000**	0.900	**0.930**	**0.950**
	HDCC	0.833	0.830	0.776	0.760	**1.000**	**0.950**	**0.930**	0.925
	E-vector	**0.940**	**0.870**	0.780	**0.860**	**1.000**	**0.950**	**0.930**	**0.950**

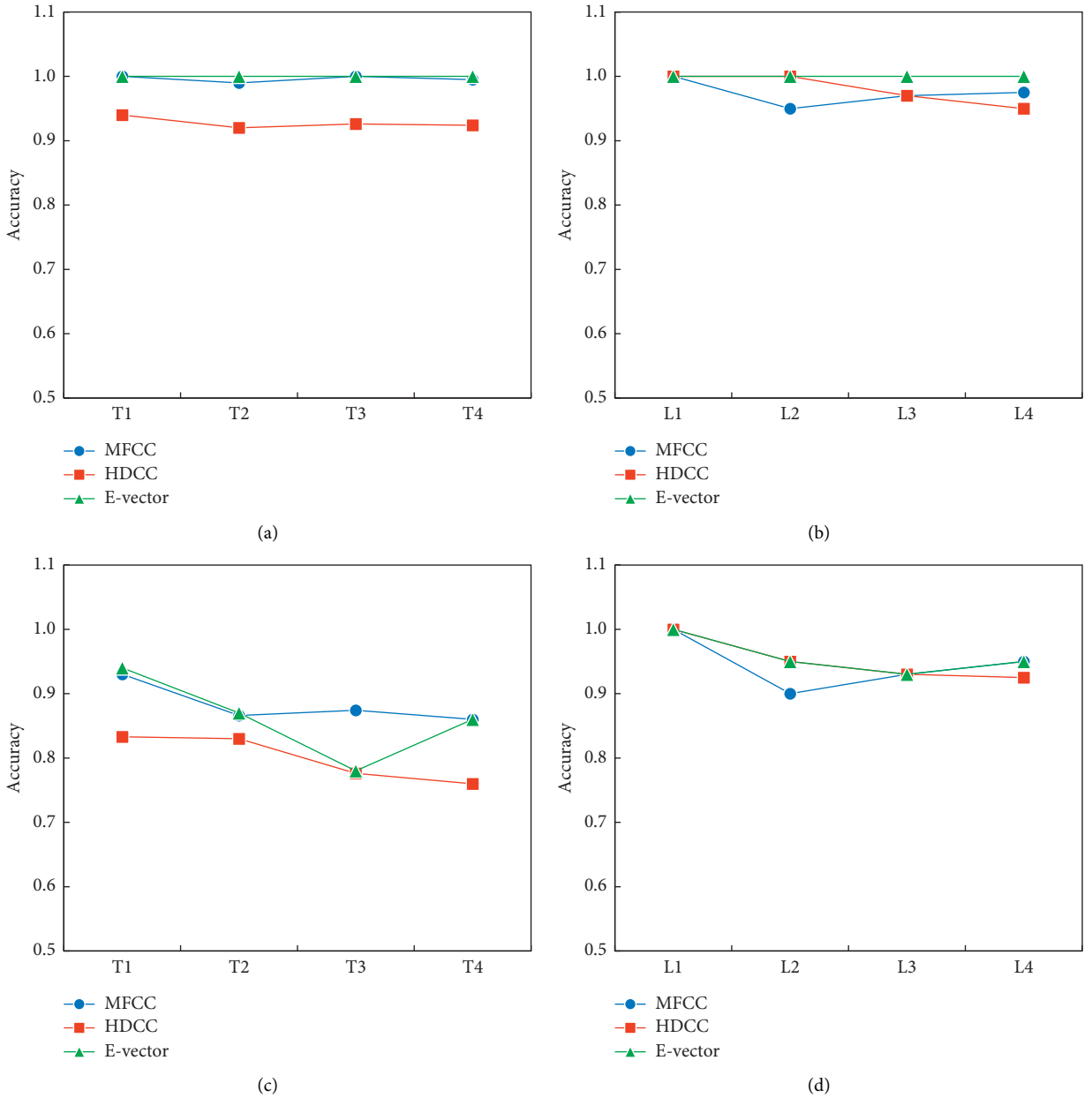


Figure 4: Single-utterance experiment accuracy comparison. (a) TIMIT corpus using GMM. (b) LibriSpeech corpus using GMM. (c) TIMIT corpus using HMM. (d) LibriSpeech corpus using HMM.

Table 4: Three-utterance experiment micro-F1 score comparison.

Model	Feature	T1	T2	T3	T4	L1	L2	L3	L4
GMM	MFCC	**0.990**	0.971	0.965	0.973	**1.000**	**1.000**	**1.000**	**1.000**
	HDCC	0.897	0.793	0.778	0.765	**1.000**	**1.000**	0.978	0.992
	E-vector	0.977	**0.978**	**0.975**	**0.976**	**1.000**	**1.000**	**1.000**	**1.000**
HMM	MFCC	0.940	0.903	0.902	0.895	1.000	0.933	0.925	0.900
	HDCC	0.877	0.827	0.815	0.807	**1.000**	**0.950**	0.956	0.958
	E-vector	**0.943**	**0.920**	**0.905**	**0.899**	**1.000**	**0.950**	**0.967**	**0.975**

Table 5: Five-utterance experiment micro-F1 score comparison.

Model	Feature	T1	T2	T3	T4	L1	L2	L3	L4
GMM	MFCC	0.962	0.948	0.939	0.928	**1.000**	0.990	0.987	0.990
	HDCC	0.706	0.699	0.659	0.630	0.970	**1.000**	0.960	0.970
	E-vector	**0.968**	**0.959**	**0.954**	**0.948**	**1.000**	**1.000**	**0.987**	**0.990**
HMM	MFCC	0.916	0.889	0.886	0.860	**1.000**	0.940	**0.953**	0.940
	HDCC	0.816	0.772	0.753	0.735	**1.000**	0.890	**0.953**	0.955
	E-vector	**0.912**	**0.895**	**0.887**	**0.876**	**1.000**	**0.950**	0.920	**0.965**

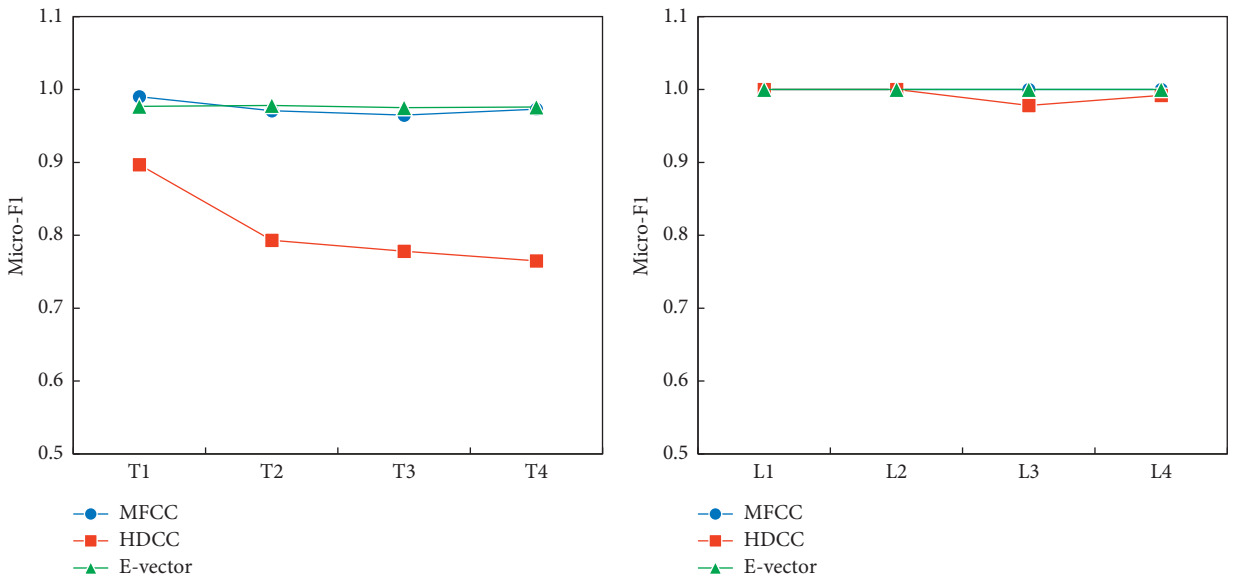


Figure 5: Continued.

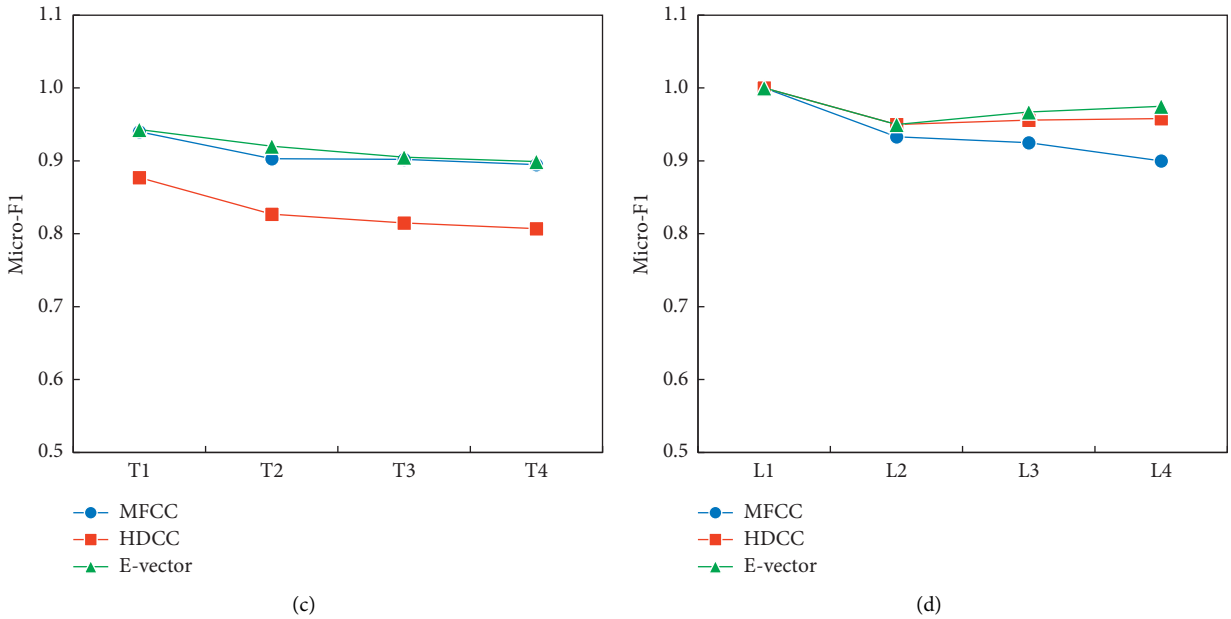


Figure 5: Three-utterance experiment micro-F1 score comparison. (a) TIMIT corpus using GMM. (b) LibriSpeech corpus using GMM. (c) TIMIT corpus using HMM. (d) LibriSpeech corpus using HMM.

corpus, the E-vector also performs best with accuracy 1 when using the GMM; MFCC and HDCC are inferior to E-vector as shown in Figure 4(b). The MFCC is almost similar to the HDCC as shown in Figure 4(d), where the accuracy is approximately 0.93 using HMM. The identification result of E-vector shows remarkable advantages in single-utterance comparison experiments.

3.2.3. Multiple-Utterance Comparison Experiments. In order to further verify the effectiveness of E-vector, we conducted the experiments with the input signal speech containing multiple utterances with GMM and HMM. We firstly used signals containing three utterances from each speaker, and the experiment results are shown in Table 4. Subsequently, we added the utterance of each speaker and used the signals with five utterances of each speaker, and the results are shown in Table 5.

In Figure 5, the identification results with the input signal containing three utterances are shown. (1) When GMM is used as the SI model, the micro-F1 scores of E-vector are a little higher than MFCC, approximately 1%, in both TIMIT and LibriSpeech corpuses. The micro-F1 score of HDCC is less than MFCC and E-vector by approximately 20% (see Figure 5(a)). (2) When HMM is the SI model, the micro-F1 score of E-vector is almost equal to MFCC. HDCC is inferior to others by approximately 10% (see Figure 5(c)) and it is almost equal to MFCC and E-vector in the LibriSpeech (see Figures 5(b) and 5(d)).

In Figure 6, the identification results with the input signal containing five utterances are described. When using GMM as the SI model, the micro-F1 scores of E-vector and MFCC are almost equal, as shown in Figures 6(a) and 6(b). When using HMM as the SI model, the results of the MFCC and E-vector are almost the same level, as shown in Figures 6(c) and 6(d). The micro-F1 score of HDCC is less than MFCC and E-vector by approximately 20% (see Figure 6(a)), and it is a little inferior to others (see Figure 6(b)). In the LibriSpeech corpus, we can find in Figures 6(b) and 6(d) that both MFCC and E-vector show good performances with the score of micro-F1 over 0.96.

Since mirco-F1 score is a collaborative measure of precision and recall, it can better denote the identification performance and stability. Therefore, we calculated the average value of micro-F1 score (Avg. micro-F1) and standard deviation of the micro-F1 score (Std. Dev. micro-F1) in the experiments of multiple utterances (three utterances and five utterances) and compared the results of E-vector with MFCC and HDCC based on different models and corpus databases (see Table 6 and Figure 7).

In Figure 7(a), we can found that the both Avg. micro-F1 and Std. Dev. micro-F1 of E-vector were superior to MFCC and much better than HDCC. However, Figure 7(b) shows that E-vector obtained a similar level of Std. Dev. micro-F1 as MFCC. E-vector still outperformed MFCC and HDCC in comparison of Avg. micro-F1. Subsequently, we can also obtain the same results in Figures 7(c) and 7(d). Particularly, in case of TIMIT corpus (see Figure 7(c)), the Avg. micro-F1 of E-vector can be improved by 0.65% and 21.40% against MFCC and HDCC, and Std. Dev. micro-F1 of E-vector can be improved by 5.41% and 21.40% against the other two vocal features. In a word, the above investigations revealed that the average proportion of mistaking the matched utterance as the unmatched utterance of E-vector is less than MFCC and HDCC; namely, FRR of E-vector is lower in multiple-utterance SI tasks.

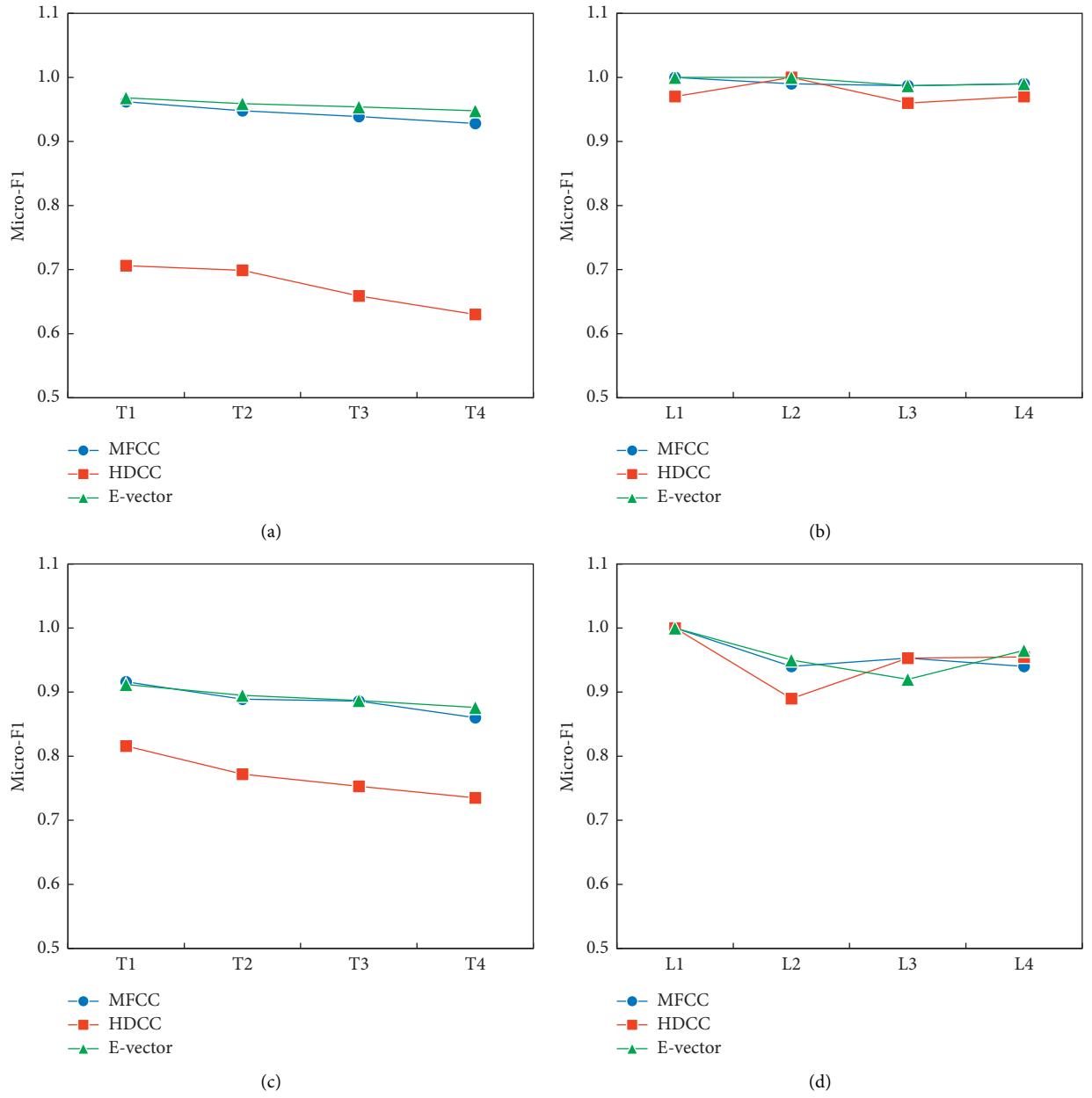


Figure 6: Five-utterance experiment micro-F1 score comparison. (a) TIMIT corpus using GMM. (b) LibriSpeech corpus using GMM. (c) TIMIT corpus using HMM. (d) LibriSpeech corpus using HMM.

Table 6: Comparison of Avg. micro-F1and Std. Dev. micro-F1 SD of multiple-utterance identification.

Micro-F1	Different models					
	GMM			HMM		
	E-vector	MFCC	HDCC	E-vector	MFCC	HDCC
Avg.	↑0.982	0.978	0.862	↑0.935	0.924	0.879
Std. Dev.	↓0.018	0.023	0.139	↓0.039	↓0.039	0.090
	Different corpus databases					
	TIMIT			LibriSpeech		
	E-vector	MFCC	HDCC	E-vector	MFCC	HDCC
Avg.	↑0.936	0.930	0.771	↑0.982	0.972	0.971
Std. Dev.	↓0.035	0.037	0.071	↓0.025	0.034	0.029

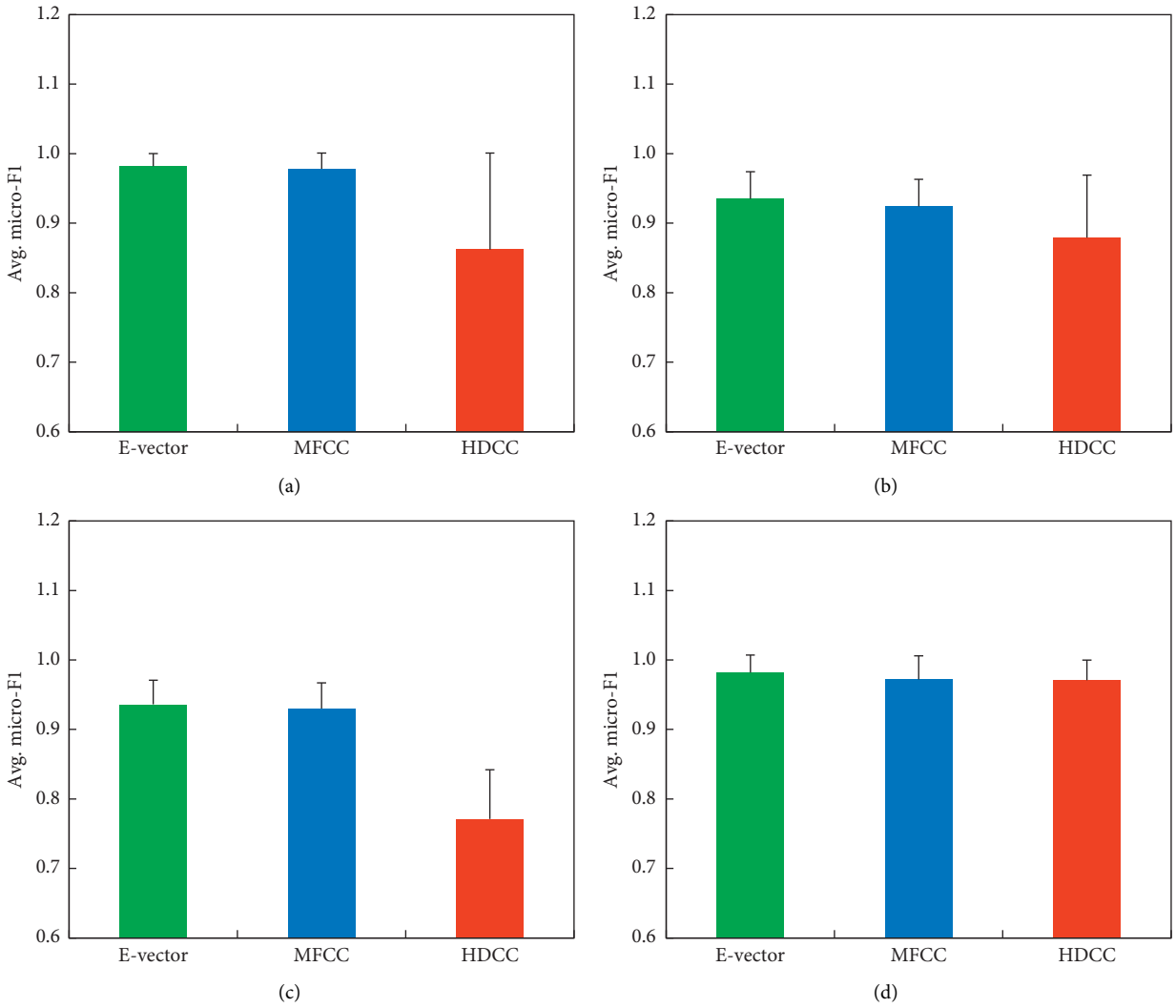


Figure 7: Stability comparison of E-vector, MFCC, and HDCC in different situations. (a) GMM-based SI using TIMIT and LibriSpeech corpus. (b) HMM-based SI using TIMIT and LibriSpeech corpus. (c) SI of TIMIT corpus using GMM and HMM. (d) SI of LibriSpeech corpus using GMM and HMM.

4. Conclusions

In this paper, we proposed a novel data-driven approach for vocal feature extraction based on DSS. Our method learns the E-vector with the minimization of the Euclidean metric using hierarchical analysis for the IDCT cepstrum coefficient, which is obtained by voice data preprocessing. Several different graphs of the experiments illustrate the effectiveness of our method in challenging SI tasks.

As a vocal feature extraction method, the generalization of E-vector is also significant. Our results show that E-vector has perfect identification performances in both one- and multiple-utterance experiments in different corpus databases, with an approximately 1.5% superiority to MFCC at best. It is also shown that our method is suitable to both GMM and HMM, with an approximately 2.1% average micro-F1 score superiority to MFCC at best. These advantages of the proposed method contribute to the capabilities of voice-feature extraction and enhance its usability for more real-world identification tasks. In our future work, we plan to investigate cosine similarity and correlation coefficient calculation methods to extract more optimized feature vectors.

Data Availability

The data and codes used in this article are given as follows: TIMIT corpus: http://academictorrents.com/details/34e2b78745138186976cbc27939b1b34d18bd5b3; LibriSpeech corpus: http://www.openslr.org/12/; E-vector source code: https://github.com/XiaoHe68/voice2vector.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

He Ma and Yi Zuo contributed equally to this work.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (under Grant nos. 61751202, U1813203, 61803064, 51939001, and 61976033), the Science and Technology Innovation Funds of Dalian (under Grant no. 2018J11CY022), the Liaoning Revitalization Talents Program (under Grant nos. XLYC1807046 and XLYC1908018), the Natural Science Foundation of Liaoning Province (under Grant nos. 2019-ZD-0151 and 2020-HYLH-26), and the Fundamental Research Funds for the Central Universities (under Grant no. 3132019345).

References

[1] D. Zhao, W. Luo, L. Ran, and L. Yue, "Negative iris recognition," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 1, pp. 112–125, 2015.

[2] T.-Y. Jea and V. Govindaraju, "A minutia-based partial fingerprint recognition system," *Pattern Recognition*, vol. 38, no. 10, pp. 1672–1684, 2005.

[3] R. Cappelli, M. Ferrara, and D. Maltoni, "Minutia cylinder-code: a new representation and matching technique for fingerprint recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2128–2141, 2010.

[4] K. Cao and A. K. Jain, "Automated latent fingerprint recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 4, pp. 788–800, 2019.

[5] V. Bruce and A. Young, "Understanding face recognition," *British Journal of Psychology*, vol. 77, no. 3, pp. 305–327, 2011.

[6] Z. Liu, Z. Wu, T. Li, J. Li, and C. Shen, "GMM and CNN hybrid method for short utterance speaker recognition," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3244–3252, 2018.

[7] B. S. Atal and S. L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *The Journal of the Acoustical Society of America*, vol. 50, pp. 637–655, 2005.

[8] L. Muda, M. Begam, and I. Elamvazuthi, "Voice recognition algorithms using Mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques," *Journal of Computing*, vol. 2, 2010.

[9] S. Cai, X. Li, X. Zou et al., "Power normalized perceptional linear predictive feature for robust automatic speech recognition," in *Proceedings of INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, Institute of Noise Control Engineering, Osaka Japan, pp. 3022–3027, 2011.

[10] S. Al-Rawahy, A. Hossen, and U. Heute, "Text-independent speaker identification system based on the histogram of DCT-cepstrum coefficients," *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 16, no. 3, pp. 141–161, 2012.

[11] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 7, pp. 1315–1329, 2016.

[12] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[13] E. Variani, X. Lei, E. McDermott, I. Lopez Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4052–4056, IEEE, Florence, Italy, May 2014.

[14] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5115–5119, IEEE, Shanghai, China, May 2016.

[15] Alibaba Group Holding Limited and Grand Cayman (KY), "Method of speech recognition and device thereof: CN 107564513 A," CN Patent 107564513 A, 2018.

[16] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[17] V. Zue, S. Seneff, and J. Glass, "Speech database development at MIT: timit and beyond," *Speech Communication*, vol. 9, no. 4, pp. 351–356, 1990.

[18] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, IEEE, Brisbane, Australia, April 2015.

Hindawi
Scientific Programming
Volume 2020, Article ID 8930387, 10 pages
https://doi.org/10.1155/2020/8930387

Research Article

A Grey Target Group Decision Method with Dual Hesitant Fuzzy Information considering Decision-Maker's Loss Aversion

Yufeng Zhou,[1,2] **Yufeng Li**,[3] **and Zhi Li**[1]

[1]*Chongqing Engineering Technology Research Center for Information Management in Development, Chongqing Technology and Business University, Chongqing, China*
[2]*Postdoctoral Research Station of Management Science and Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 211106, China*
[3]*Research Center for Economy of Upper Researches of the Yangtze River, Chongqing Technology and Business University, Chongqing, China*

Correspondence should be addressed to Yufeng Zhou; xtuzyf@qq.com

Received 19 September 2019; Revised 19 January 2020; Accepted 8 June 2020; Published 29 June 2020

Academic Editor: Kifayat Ullah Khan

The uncertainty, complexity, and behavioral preference are widely existing in real-world decision-making problems. In this paper, different from the previous grey target decision method, we propose a novel grey target group decision method considering decision-maker's loss aversion with positive and negative clouts under the dual hesitant fuzzy environment. Firstly, defining the dual hesitant fuzzy ideal optimization scheme as the positive clout and the ideal inferior scheme as the negative clout, positive and negative target-eye distances are measured by the normalized Hamming distances from the DHFEs to the positive clout and the negative clout. Then, a new comprehensive target-eye distance is proposed to evaluate alternatives between the positive and the negative clout. A nonlinear optimization model is established to obtain the optimal initial attribute weights with the goal of minimizing the comprehensive target-eye distance. Then, a grey target group decision method with dual hesitant fuzzy information considering decision-maker's loss aversion and variable weights is proposed. Finally, a numeral example is given to verify the effectiveness and practicality of the proposed model and method.

1. Introduction

Multiattribute decision-making (MADM) is an important issue of modern decision-making science. Recently, MADM has become a hotspot in the decision-making area, and a lot of academic achievements have been made in the related domain [1–3]. Multiattribute group decision-making (MAGDM) can use the wisdom of the group to promote scientific decision-making. Pattern matching optimization is an effective method, which is applied to many areas in MAGDM, such as tracking and detection [4, 5], computer engineering [6], physical sciences [7], health-related issues [8], natural sciences, and industrial academic areas [9]. In addition, other methods, such as TOPSIS [10, 11], probabilistic linguistic term sets [12], data-capturing devices [13–17], the state-of-the-art features [18–22], the hidden Markov model (HMM), modified HMM, embedded HMM, Gaussian mixture modal (GMM), and the support vector machine (SVM) are also widely used in the field of MAGDM [23–26]. Grey target decision is one of the main methods using grey system theory to solve the problem of uncertain MADM and MAGDM under incomplete information condition. The main idea of the grey target decision is to achieve a uniform dimension of the Euclidean space through measuring and transforming index sets in the absence of the standard mode, i.e., the grey target. All the alternatives are distributed on the grey target. A target-eye of the grey target is found as the standard mode. Then, all the decision points in the grey target are compared with the target-eye to obtain relative target-eye distances, and the dominant position of each decision point is sorted by the target-eye distances.

In recent years, many scholars have carried out theoretical research on the classic grey target decision methods from different perspectives. Gupta et al. [27] presented an extended TOPSIS method under the interval-valued intuitionistic fuzzy environment. Luo and Wang [28] studied the multiattribute grey target decision method for the attribute value within the three-parameter interval grey number. Wang et al. [29] studied the multiattribute grey target decision method based on soft set theory. Qian et al. [30] built a grey target decision model based on interval grey number panel data for multiindex dynamic problems. Ye [31] proposed an extended TOPSIS method with interval-valued intuitionistic fuzzy numbers for virtual enterprise partner selection. Fu et al. [32] presented a kind of multiattribute grey target decision model based on the positive-negative target center aimed at the complexity and uncertainty of the actual decision environment. Zeng et al. [33] proposed a grey target decision-making model based on the cobweb area and gave an application of choosing the software development pattern.

Fuzzy sets (FSs) are also an effective method to deal with the uncertain problem of decision-making. FSs are used to deal with vagueness and fuzziness in real decision-making problems, and several extensions have been developed. Rodriguez et al. [34] studied the hesitant fuzzy linguistic term set on MADM. Traditional hesitant fuzzy decision-making methods just provide the membership degrees and without considering the importance of the nonmembership degrees. In fact, the nonmembership plays the same important role in describing the vague decision-making information, which indicates the possible degrees of one element does not belong to a fixed set. To assess the attribute values more precisely, Zhu et al. [35] developed the dual hesitant fuzzy set (DHFS), taking into account much more information given by decision-makers, in which the membership degree and the nonmembership degree are in the form of sets of values in [0, 1]. DHFS can avoid information distortion and losing effectively in describing the vague decision-making information. Then, the aggregation operators for aggregating dual hesitant fuzzy elements (DHFEs) have attracted the attention from researchers [36–40]. Regarding the MADM problems with dual hesitant fuzzy information, many methods have been proposed. Ren and Wei [41] developed a prioritized multiattribute decision-making method to solve dual hesitant fuzzy decision problems. Then, they [42] further proposed a dual hesitant fuzzy VIKOR method for multicriteria group decision-making based on the fuzzy measure and the new comparison method. Wei et al. [43] investigated the MADM problem based on the geometric aggregation operators with interval-valued dual hesitant fuzzy linguistic information. Lu and Wei [44] investigated the MADM problem based on the arithmetic and geometric aggregation operators with dual hesitant fuzzy uncertain linguistic information.

Based on the above analysis, we can conclude that the current studies on the DHFS are mainly based on some aggregation operators which are limited to tackle the complex MAGDM problem under the dual hesitant fuzzy environment. Therefore, other classical uncertain decision methods such as the grey target decision method should be further studied for real decision-making problems under dual hesitant fuzzy information. Therefore, the grey target decision method based on positive and negative clouts proposed is extended to solve the MAGDM problems with dual hesitant fuzzy information. It is the major motivation of our study. To the best of our knowledge, the existing grey target decision-making method has not yet been accommodated to deal with the dual hesitant fuzzy information provided by decision-makers. In addition, the current research on the grey target decision method has also not considered the decision-maker's loss aversion. With the development of behavioral decision theory, the MAGDM method has been widely used on the basis of prospect theory. Therefore, the proposed model considers decision-maker's loss aversion and variable weights of attributes based on the prospect theory. The contributions of this paper can be summarized as follows. (i) We proposed a novel grey target group decision method with positive and negative clouts under the dual hesitant fuzzy environment. (ii) We introduce the prospect theory to the grey target group decision.

The remainder of this paper is structured as follows. Section 2 presents the preliminaries of DHFS and DHFEs. Section 3 is the process of establishing the mathematical models. A numerical example is presented in Section 4 followed by conclusions in Section 5.

2. Preliminaries

In this section, some essential concepts are reviewed as follows [36].

Definition 1. Let X be a given set, DHFS; D is a function mapping to X:

$$D = \{\langle x, h(x), g(x)\rangle \mid x \in X\}. \tag{1}$$

In expression (1), $h(x)$ and $g(x)$ are two sets of numbers within the interval [0, 1]. $h(x)$ and $g(x)$, respectively, denote the possible membership degrees and nonmembership degrees under the following conditions:

$$0 \le r, \eta \le 1, 0 \le r^+ + \eta^+ \le 1, \tag{2}$$

where $r \in h(x)$, $\eta \in \mathrm{g}(x)$, $r^+ \in h^+(x) = \cup_{r \in h(x)} \max\{r\}$, and $\eta^+ \in g^+(x) = \cup_{\eta \in g(x)} \max\{\eta\}$ for all $x \in X$. The pair $d(x) = \langle h(x), g(x)\rangle$ is called the DHFE, which can be denoted simply by $d = \langle h, g\rangle$.

Obviously, if there is only one element in both $h(x)$ and $\mathrm{g}(x)$, the DHFE reduces to an intuitionistic fuzzy number.

Definition 2 (see [34]). Let $d_1 = \langle h_1, g_1\rangle$ and $d_2 = \langle h_2, g_2\rangle$, respectively, denote two DHFEs; then, the operational laws can be defined as

$$
\begin{aligned}
d_1 \oplus d_2 &= \langle U_{r_1\in h_1, r_2\in h_2, \eta_1\in g_1, \eta_2\in g_2}\{\{r_1 + r_2 - r_1 r_2\}, \{\eta_1\eta_2\}\}\rangle,\\
d_1 \otimes d_2 &= \langle U_{r_1\in h_1, r_2\in h_2, \eta_1\in g_1, \eta_2\in g_2}\{\{r_1 r_2\}, \{\eta_1 + \eta_2 - \eta_1\eta_2\}\}\rangle,\\
\lambda d_1 &= \langle U_{r_1\in h_1, \eta_1\in g_1}\left\{\left\{1-(1-r_1)^{\lambda}\right\}, \left\{(\eta_1)^{\lambda}\right\}\right\}\rangle, \quad \lambda > 0,\\
d_1^{\lambda} &= \langle U_{r_1\in h_1, \eta_1\in g_1}\left\{\left\{(r_1)^{\lambda}\right\}, \left\{1-(1-\eta_1)^{\lambda}\right\}\right\}\rangle, \quad \lambda > 0.
\end{aligned} \tag{3}
$$

Definition 3 (see [43]). Let $d_i = \langle h_i, g_i\rangle$ $(i = 1, 2, \cdots, n)$ denote a collection of DHFEs; then, the score function $S(d_i)$ and the accuracy function $P(d_i)$ of d_i $(i = 1, 2, \cdots, n)$ can be defined as

$$
\begin{aligned}
S(d_i) &= \frac{1}{\#h}\sum_{r\in h} r - \frac{1}{\#g}\sum_{\eta\in g}\eta,\\
P(d_i) &= \frac{1}{\#h}\sum_{r\in h} r + \frac{1}{\#g}\sum_{\eta\in g}\eta,
\end{aligned} \tag{4}
$$

where $\#h$ and $\#g$ are the number of values in $h(di)$ and $g(di)$, respectively.

Theorem 1 (see [45]). *Let $d_1 = <h_1, g_1>$ and $d_2 = <h_2, g_2>$ denote two DHFEs, and they can be compared according to the following rulesfollowing rules:*

$$
\begin{aligned}
&\text{if } S(d_1) > S(d_2), \text{ then}, d_1 \succ d_2,\\
&\text{if } S(d_1) = S(d_2), \text{ then :}\\
&\text{if } P(d_1) > P(d_2), \text{ then } d_1 \succ d_2,\\
&\text{if } P(d_1) = P(d_2), \text{ then } d_1 = d_2.
\end{aligned} \tag{5}
$$

Definition 4 (see [46]). Let $d_1 = <h_1, g_1>$ and $d_2 = <h_2, g_2>$ be two DHFEs. The normalized Hamming distance is defined as

$$
d(d_1, d_2) = \frac{1}{2}\left(\left|\frac{1}{\#h_1}\sum_{r_1\in h_1} r_1 - \frac{1}{\#h_2}\sum_{r_2\in h_2} r_2\right| + \left|\frac{1}{\#g_1}\sum_{\eta_1\in g_1}\eta_1 - \frac{1}{\#g_2}\sum_{\eta_2\in g_2}\eta_2\right|\right), \tag{6}
$$

where $\#h_i$ and $\#g_i$ $(i = 1, 2)$ are the number of values in h_i and g_i, respectively.

3. A Grey Target Decision Method for MAGDM Problems with Dual Hesitant Fuzzy Information and Decision-Maker's Loss Aversion

In this section, a grey target decision method based on positive and negative clouts is developed to solve the MAGDM problem under the dual hesitant fuzzy environment and uncertain weight information.

Let $A = \{A_1, A_2, \cdots, A_m\}$ be a finite set of m alternatives and $C = \{C_1, C_2, \cdots, C_n\}$ be the set of n attributes. Let DM $= \{\mathrm{DM}_1, \mathrm{DM}_2, \cdots, \mathrm{DM}_k\}$ be the set of decision-makers whose weight vector is $\lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_k\}^T$ such that $\lambda_f \in [0, 1]$ and $\sum_{f=1}^{k}\lambda_f = 1$. Suppose that $Df = (d_{ij}^f)_{m\times n}$ is a dual hesitant fuzzy decision matrix given by DM_f, where $d_{ij}^f = <h_{ij}^f, g_{ij}^f>$ denotes the DHFEs for the evaluation of the attribute C_j $(j = 1, 2, \cdots, n)$ with respect to the alternative A_i $(i = 1, 2, \cdots, m)$, with $h_{ij}^f = \cup_{r_{ij}^f\in h_{ij}^f}(r_{ij}^f)$ and $g_{ij}^f = \cup_{\eta_{ij}^f\in g_{ij}^f}(\eta_{ij}^f)$. Then, to determine the most desirable alternatives, a novel method is proposed based on the grey target decision method with positive and negative clouts.

Step 1. Aggregate all individual dual hesitant fuzzy decision matrices $Df = (d_{ij}^f)_{m\times n}$ $(i = 1, 2, \cdots, m,\ j = 1, 2, \cdots, n,\ f = 1, 2, \cdots, k)$ into a collective dual hesitant fuzzy decision matrix by the following equation [24]:

$$
d_{ij} = \left\langle \bigcup_{r_{ij}^f\in h_{ij}^f, \eta_{ij}^f\in g_{ij}^f} \left\{ \left\{ 1 - \prod_{f=1}^{k}\left(1 - r_{ij}^f\right)^{\lambda_f} \right\}, \left\{ \prod_{f=1}^{k}\left(\eta_{ij}^f\right)^{\lambda_f} \right\} \right\} \right\rangle. \tag{7}
$$

Step 2. Identifying positive and negative clouts, and defining the dual hesitant fuzzy ideal optimization scheme as the positive clout r^+ and the ideal inferior scheme as the negative clout r^-. Positive and negative clouts in the collective dual hesitant fuzzy decision matrix D can be computed according to equations (8) and (9), respectively:

$$
r^+ = \left(r_{i1}^+, r_{i2}^+, \cdots, r_{in}^+\right), \quad (i = 1, 2, \cdots, m), \tag{8}
$$

$$
r^- = \left(r_{i1}^-, r_{i2}^-, \cdots, r_{in}^-\right), \quad (i = 1, 2, \cdots, m). \tag{9}
$$

For the benefit attributes, $r_{ij}^+ = \max_{1\le i\le m} d_{ij}$ and $r_{ij}^- = \min_{1\le i\le m} d_{ij}$, the attribute values d_{ij} can be compared by the score function $S(d_{ij})$ and the accuracy function $P(d_{ij})$ according to Theorem 1.

Step 3. Calculating positive and negative target-eye distances based on r^+ and r^-, and the positive and negative target-eye distances of each alternative are calculated using the following equations:

$$\zeta_{ij}^{+} = \sqrt{\omega_1 d_{i1}^{+2} + \omega_2 d_{i2}^{+2} + \cdots + \omega_n d_{in}^{+2}}, \quad (i = 1, 2, \cdots, m), \tag{10}$$

$$\zeta_{ij}^{-} = \sqrt{\omega_1 d_{i1}^{-2} + \omega_2 d_{i2}^{-2} + \cdots + \omega_n d_{in}^{-2}}, \quad (i = 1, 2, \cdots, m), \tag{11}$$

where $\{\omega_1, \omega_2, \cdots, \omega_n\}$ denote the attribute weight vector. $d_{ij}^{+} = d(d_{ij}, r_{ij}^{+})$ and $d_{ij}^{-} = d(d_{ij}, r_{ij}^{-})$ denote the normalized Hamming distances between the DHFEs d_{ij} to r_{ij}^{+} and r_{ij}^{-}, respectively.

Step 4. Calculating comprehensive target-eye distance: the evaluation vector of each alternative is always between the positive and negative clouts. Luo and Wang [28] proved that the optimal solution can be obtained based on r_{io}^{*} which is on the line between positive and negative clouts. Obviously, the larger r_{io}^{*}, the better the corresponding scheme. The projection r_{io}^{*} is the comprehensive target-eye distance, and r_{io}^{*} can be calculated by the following equation:

$$r_{io}^{*} = \left| \frac{\left(\zeta_{ij}^{+}\right)^2 + \left(\zeta^0\right)^2 - \left(\zeta_{ij}^{-}\right)^2}{\zeta^0} \right|, \tag{12}$$

where ζ^0 is the distance between positive clout ζ_{ij}^{+} and negative clout ζ_{ij}^{-}, $\zeta^0 = d(\zeta_{ij}^{+}, \zeta_{ij}^{-})$, which can be calculated according to equation (6).

In traditional grey target decision methods, r_{io}^{*} can be directly calculated when the attribute weights are known. Then, the alternatives can be sorted by r_{io}^{*}. In the case of unknown attribute weight, the method that how to establish the model to solve the optimal initial weight vector is to be given as follows.

Step 5. Optimization of initial attribute weights ω^0: if the weight sequence $\omega = (\omega_1, \omega_2, \cdots, \omega_n)$ is unknown, the sequence is the grey connotative sequence, and the grey entropy can be defined as

$$H_{\otimes}(\omega) = -\sum_{s=1}^{n} \omega_s \ln \omega_s. \tag{13}$$

According to the maximum entropy principle, $\omega_s(1, 2, \cdots, n)$ should be adjusted to minimize the sequence uncertainty of $\omega^0 = (\omega_1, \omega_2, \cdots, \omega_n)$ and maximize $H_{\otimes}(\omega)$. At the same time, considering the proximity between the effect measure of each scheme and the positive clout, the distance between the positive and negative clouts is minimized, and the optimization model is proposed as follows:

$$\begin{cases} \min \sum_{i=1}^{m} \sum_{j=1}^{n} r_{io}^{*}, \\ \max H_{\otimes}(\omega) = -\sum_{s=1}^{n} \omega_s \ln \omega_s, \\ \sum_{s=1}^{n} \omega_s = 1, \\ 0 \le \omega_s \le 1, \quad \forall s = 1, 2, \cdots, n. \end{cases} \tag{14}$$

Let $f_1(\omega) = \sum_{i=1}^{m} \sum_{j=1}^{n} r_{io}^{*}$ and $f_2(\omega) = \sum_{s=1}^{n} \omega_s \ln \omega_s$; $f_1^{\max}(\omega)$ and $f_1^{\min}(\omega)$ are the maximum and minimum values of $f_1(\omega)$, respectively; $f_2^{\max}(\omega)$ and $f_2^{\min}(\omega)$ are the maximum and minimum values of $f_2(\omega)$, respectively. The two targets are dimensionless. Then, the above multi-objective optimization problems can be transformed into the following single-objective optimization problem:

$$\begin{cases} \min \lambda \left[\dfrac{f_1(\omega) - f_1^{\min}(\omega)}{f_1^{\max}(\omega) - f_1^{\min}(\omega)} \right] + (1 - \lambda) \left[\dfrac{f_2(\omega) - f_2^{\min}(\omega)}{f_2^{\max}(\omega) - f_2^{\min}(\omega)} \right], \\ \sum_{s=1}^{n} \omega_s = 1, \\ 0 \le \omega_s \le 1, \quad \forall s = 1, 2, \cdots, n, \\ 0 \le \lambda \le 1. \end{cases} \tag{15}$$

In view of the fair competition of optimizing objective functions, usually set $\lambda = 0.5$ [28]. The optimal initial weight vector can be obtained by using available mathematical programing software. The optimal solution ω^* is the initial weight ω^0.

Step 6. Calculating the variable weights: firstly, the relative closeness of alternatives should be calculated by the following equation:

$$z_{ij} = \frac{d\left(d_{ij}, r_{ij}^{+}\right)}{d\left(d_{ij}, r_{ij}^{+}\right) + d\left(d_{ij}, r_{ij}^{-}\right)}. \tag{16}$$

Then, let z_j be the relative closeness of a positive ideal solution and $S(z_j)$ be the state variable weight vector.

$$S_i\left(z_j\right) = \frac{\partial B\left(z_j\right)}{\partial z_{ij}}. \tag{17}$$

In equation (17), $B(z_j)$ is an equilibrium function.

The variable weights can be calculated by the following equation:

$$w\left(z_j\right) = \frac{\omega^0 \otimes S\left(z_j\right)}{\sum_i z_{ij}}. \tag{18}$$

Step 7. Calculating the loss values of attribute weights: taking the initial weight as the reference point, the weight loss matrix of the variable weight vector relative to the initial weight is established as F:

$$F = \left[F\left(w_i\left(z_j\right)\right)\right]_{m\times n} = \begin{cases} w_i\left(z_j\right) - w_i^0, & \text{if } w_i\left(z_j\right) \geq w_i^0, \\ -\left(w_i^0 - w_i\left(z_j\right)\right), & \text{if } w_i\left(z_j\right) \leq w_i^0, \end{cases} \tag{19}$$

Step 8. Establishing the foreground theory matrix: considering the decision-maker's different risk attitudes towards the gain and loss of weight, the prospect theory matrix is established as

$$V\left(w_i\left(z_j\right)\right) = \begin{cases} F\left(w_i\left(z_j\right)\right)^{\partial}, & \text{if } w_i\left(z_j\right) \geq w_i^0, \\ -\theta\left(-F\left(w_i\left(z_j\right)\right)\right)^{\beta}, & \text{if } w_i\left(z_j\right) \leq w_i^0, \end{cases} \tag{20}$$

Step 9. Ranking the alternatives according to comprehensive target-eye distances, R_{io}^* is calculated based on the variable weight vector and the prospect theory matrix. The most desirable alternative which has a higher value of R_{io}^* should be selected:

$$R_{io}^* = \sum_i V\left(w_i\left(z_j\right)\right) * \left|r_{io}^*\right|. \tag{21}$$

In summary, the decision process can be described as Figure 1.

4. An Example and the Analysis

4.1. A Numerical Example. In this section, a numerical example is given to present the application of the MAGDM method and to demonstrate its feasibility and effectiveness in a realistic scenario. An investment firm would like to invest four possible alternatives within the volume of total investment. The problem could be resolved through selecting the best options.

Three decision-makers $\{\mathrm{DM}_1, \mathrm{DM}_2, \mathrm{DM}_3\}$ (the weight vector is $\lambda = (0.25, 0.40, 0.35)^T$) are invited to evaluate the alternatives based on the following three attributes: $C_j\,(j = 1, 2, 3)$, where C_1 denotes the market share analysis, C_2 denotes the market growth analysis, and C_3 denotes the benefit analysis. In this case, all attributes are positive. The attribute weight vector is unknown. The four possible alternatives $A_i\,(i = 1, 2, 3, 4)$ are evaluated by using the DHFEs under the above three attributes; then, the dual hesitant fuzzy decision matrices $Df = (d_{ij}^f)_{4\times 3}$ $(i = 1, 2, 3, 4;\ j = 1, 2, 3;$ and $f = 1, 2, 3)$ are constructed as shown in Tables 1–3.

(1) Aggregating all individual dual hesitant fuzzy decision matrices $Df = (d_{ij}^f)_{4\times 3}$ $(i = 1, 2, 3, 4;\ j = 1, 2, 3;$ and $f = 1, 2, 3)$ into the collective dual hesitant fuzzy decision matrix $D = (d_{ij})_{4\times 3}$ by equation (7):

$d_{11} = \{[0.4113, 0.4767, 0.5160, 0.5201, 0.5734, 0.6054, 0.4375, 0.5000, 0.5376, 0.5415, 0.5924, 0.6230, 0.4680, 0.5271, 0.5627, 0.5663, 0.6145, 0.6435], [0.2213, 0.2551, 0.2603, 0.3000, 0.2378, 0.2741, 0.2797, 0.3224]\}.$

$d_{12} = \{[0.4150, 0.4589, 0.4561, 0.4970, 0.5026, 0.5399, 0.4622, 0.5026, 0.5000, 0.5376, 0.5427, 0.5771], [\ 0.1569, 0.2305, 0.1845, 0.2711, 0.1737, 0.2551, 0.2042, 0.3000]\}.$

$d_{13} = \{[0.3568, 0.3966, 0.4419, 0.4267, 0.4622, 0.5026, 0.4671, 0.5000, 0.5376, 0.3917, 0.4293, 0.4722, 0.4578, 0.4914, 0.5296, 0.4960, 0.5271, 0.5627], [0.2280, 0.2521, 0.2558, 0.2828, 0.2711, 0.2998, 0.3041, 0.3364]\}.$

$d_{21} = \{[0.4000, 0.4371, 0.4898, 0.5214, 0.5453, 0.5734, 0.4267, 0.4622, 0.5126, 0.5427, 0.5655, 0.5924], [0.2138, 0.2312, 0.2821, 0.3050, 0.2297, 0.2484, 0.3031, 0.3278]\}.$

$d_{22} = \{[0.4358, 0.5214, 0.5573, 0.4971, 0.5734, 0.6054, 0.4609, 0.5427, 0.5771, 0.5195, 0.5924, 0.6230], [0.1803, 0.2297, 0.2797, 0.3565, 0.1906, 0.2429, 0.2958, 0.3770]\}.$

$d_{23} = \{[0.4006, 0.4799, 0.4518, 0.5243, 0.4671, 0.5376, 0.5126, 0.5771, 0.4960, 0.5627, 0.5390, 0.6000], [0.2305, 0.2549, 0.3041, 0.3364, 0.2741, 0.3031, 0.3617, 0.4000]\}.$

$d_{31} = \{[0.4106, 0.4671, 0.5000, 0.4609, 0.5126, 0.5427, 0.4813, 0.5309, 0.5599, 0.5256, 0.5710, 0.5975], [0.2711, 0.2998, 0.3041, 0.3364, 0.3000, 0.3318, 0.3366, 0.3722]\}.$

$d_{32} = \{[0.3143, 0.3566, 0.4006, 0.4377, 0.5114, 0.5416, 0.3618, 0.4013, 0.4422, 0.4767, 0.5453, 0.5734, 0.3903, 0.4280, 0.4671, 0.5000, 0.5655, 0.5924], [0.1677, 0.1933, 0.2213, 0.2551, 0.1803, 0.2077, 0.2378, 0.2741]\}.$

$d_{33} = \{[0.4013, 0.4993, 0.4767, 0.5624, 0.4280, 0.5216, 0.5000, 0.5819, 0.4965, 0.5790, 0.5599, 0.6320], [0.2000, 0.2305, 0.2352, 0.2711, 0.2213, 0.2551, 0.2603, 0.3000]\}.$

$d_{41} = \{[0.3364, 0.4371, 0.4794, 0.3831, 0.4767, 0.5160, 0.4004, 0.4914, 0.5296, 0.4426, 0.5271, 0.5627, 0.4958, 0.5723, 0.6044, 0.5313, 0.6024, 0.6322], [0.1978, 0.2521, 0.2219, 0.2828, 0.2352, 0.2998, 0.2639, 0.3364]\}.$

$d_{42} = \{[0.4419, 0.4954, 0.5376, 0.5819, 0.5771, 0.6176, 0.4722, 0.5228, 0.5627, 0.6045, 0.6000, 0.6383], [0.1569, 0.2305, 0.2071, 0.3041, 0.1866, 0.2741, 0.2462, 0.3617]\}.$

$d_{43} = \{[0.4578, 0.5296, 0.6309, 0.4960, 0.5627, 0.6569, 0.4955, 0.5622, 0.6565, 0.5309, 0.5930, 0.6807], [0.1741, 0.2219, 0.1904, 0.2426, 0.2291, 0.2921, 0.2505, 0.3193]\}.$

(2) Identifying the dual hesitant fuzzy positive clout r^+ and negative clout r^- of alternatives in the collective dual hesitant fuzzy decision matrix D by equations (8) and (9), respectively:

$r^+ = (<\{0.4113, 0.4767, 0.5160, 0.5201, 0.5734, 0.6054, 0.4375, 0.5000, 0.5376, 0.5415, 0.5924,$

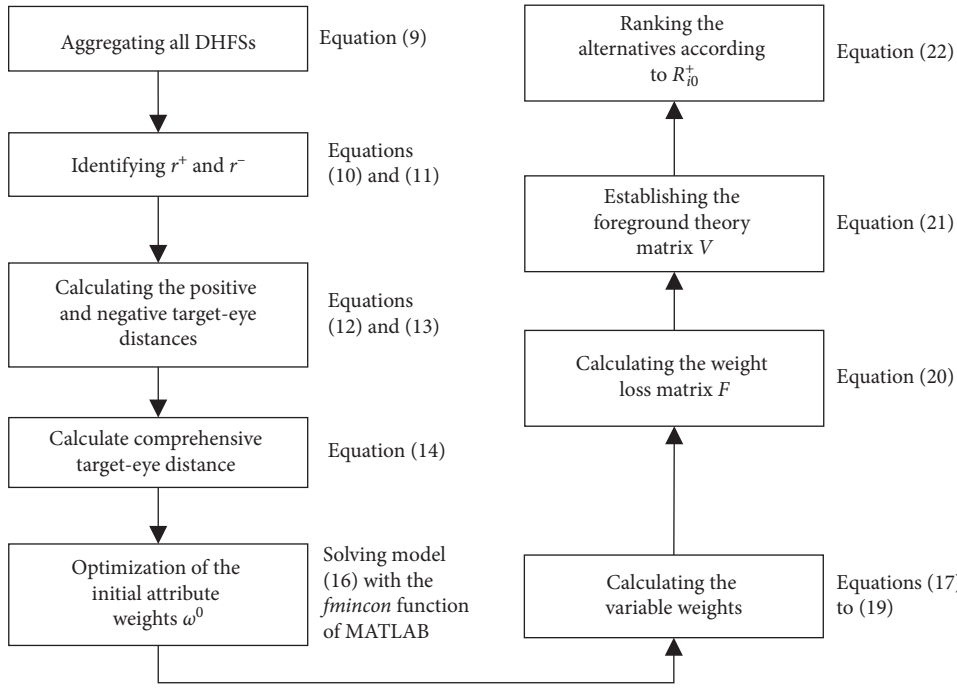


Figure 1: Decision flow of the proposed method.

Table 1: Dual hesitant fuzzy decision matrix D_1 given by DM_1.

	C1	C2	C3
A1	A1	{[0.4 0.5 0.6], [0.3 0.4]}	{[0.3 0.5], [0.2 0.3]}
A2	A2	{[0.4 0.5], [0.3 0.4]}	{[0.4 0.5], [0.4 0.5]}
A3	A3	{[0.5 0.7], [0.2 0.3]}	{[0.2 0.4 0.5], [0.3 0.4]}
A4	A4	{[0.4 0.6 0.8], [0.1 0.2]}	{[0.5 0.6], [0.2 0.4]}

Data source: Yang and Ju [45].

Table 2: Dual hesitant fuzzy decision matrix D_2 given by DM_2.

	C1	C2	C3
A1	A1	{[0.5 0.7], [0.2 0.3]}	{[0.4 0.5 0.6], [0.2 0.3]}
A2	A2	{[0.4 0.6 0.7], [0.1 0.2]}	{[0.6 0.7], [0.1 0.3]}
A3	A3	{[0.5 0.6], [0.3 0.4]}	{[0.3 0.5 0.7], [0.1 0.2]}
A4	A4	{[0.4 0.5], [0.3 0.4]}	{[0.2 0.5 0.6], [0.2 0.4]}

Data source: Yang and Ju [45].

Table 3: Dual hesitant fuzzy decision matrix D_3 given by DM_2.

	C1	C2	C3
A1	{[0.3 0.5 0.6], [0.2 0.3}	{[0.5, 0.6], [0.1 0.2]}	{[0.4 0.5 0.6], [0.3 0.4]}
A2	{[0.4 0.5], [0.4 0.5]}	{[0.2 0.5 0.6], [0.2 0.4]}	{[0.4 0.6], [0.3 0.4]}
A3	{[0.2 0.4 0.5], [0.3 0.4]}	{[0.4 0.5], [0.2 0.3]}	{[0.6 0.7], [0.2 0.3]}
A4	{[0.2 0.5 0.6], [0.2 0.4]}	{[0.6 0.7], [0.1 0.3]}	{[0.4 0.6 0.8], [0.1 0.2]}

Data source: Yang and Ju [45].

0.6230, 0.4680, 0.5271, 0.5627, 0.5663, 0.6145, 0.6435}, {0.2213, 0.2551, 0.2603, 0.3000, 0.2378, 0.2741, 0.2797, 0.3224}>, <{0.4419, 0.4954, 0.5376, 0.5819, 0.5771, 0.6176, 0.4722, 0.5228, 0.5627, 0.6045, 0.6000, 0.6383}, {0.1569, 0.2305, 0.2071, 0.3041, 0.1866, 0.2741, 0.2462, 0.3617}>, <{0.4578, 0.5296, 0.6309, 0.4960, 0.5627, 0.6569, 0.4955, 0.5622, 0.6565, 0.5309, 0.5930, 0.6807}, {0.1741, 0.2219, 0.1904, 0.2426, 0.2291, 0.2921, 0.2505, 0.3193}>).

$r^- = (<\{0.4106, 0.4671, 0.5000, 0.4609, 0.5126, 0.5427, 0.4813, 0.5309, 0.5599, 0.5256, 0.5710, 0.5975\}, \{0.2711, 0.2998, 0.3041, 0.3364, 0.3000, 0.3318, 0.3366, 0.3722\}>, <\{0.3143, 0.3566, 0.4006, 0.4377, 0.5114, 0.5416, 0.3618, 0.4013, 0.4422, 0.4767, 0.5453, 0.5734, 0.3903, 0.4280, 0.4671, 0.5000, 0.5655, 0.5924\}, \{0.1677, 0.1933, 0.2213, 0.2551, 0.1803, 0.2077, 0.2378, 0.2741\}>, <\{0.3568, 0.3966, 0.4419, 0.4267, 0.4622, 0.5026, 0.4671, 0.5000, 0.5376, 0.3917, 0.4293, 0.4722, 0.4578, 0.4914, 0.5296, 0.4960, 0.5271, 0.5627\}, \{0.2280, 0.2521, 0.2558, 0.2828, 0.2711, 0.2998, 0.3041, 0.3364\}>).$

(3) Calculating the positive and negative target-eye distances of each alternative from r^+ and r^- by equations (10) and (11), respectively:

$$\zeta_{ij}^+ = \begin{pmatrix} \sqrt{0 * \omega_1 + 0.0016\omega_2 + 0.0049\omega_3}, \sqrt{3.1108e-04\omega_1 + 3.1204e-04\omega_2 + 0.0040\omega_3}, \\ \sqrt{0.0015\omega_1 + 0.0037\omega_2 + 8.3702e-04\omega_3}, \sqrt{5.3528e-04\omega_1 + 0*\omega_2 + 0*\omega_3} \end{pmatrix},$$
$$\zeta_{ij}^- = \begin{pmatrix} \sqrt{0.0383\omega_1 + 0.0213\omega_2 + 0*\omega_3}, \sqrt{0.0295\omega_1 + 0.0663\omega_2 + 0.0362\omega_3}, \\ \sqrt{0*\omega_1 + 0*\omega_2 + 0.0413\omega_3}, \sqrt{0.0350\omega_1 + 0.0608\omega_2 + 0.0702\omega_3} \end{pmatrix}. \tag{22}$$

(4) Calculate comprehensive target-eye distance r_{io}^* by equation (12):

$$r_{io}^* = \frac{\left(\zeta_{ij}^+\right)^2 + \left(\zeta^0\right)^2 - \left(\zeta_{ij}^-\right)^2}{\zeta^0}, \tag{23}$$

where ζ^0 can be calculated by equation (6), $\zeta^0 = \sqrt{0.0015 * \omega_1 + 0.0037\omega_2 + 0.0049\omega_3}$.

(5) Optimization of the initial attribute weights ω^0: the *fmincon* function of MATLAB is used to solve the model, and the optimal weight can be obtained, $\omega^0 = (\omega_1, \omega_2, \omega_3) = [0.4823\ 0.3699\ 0.1478]$.

(6) Calculating the variable weights: the relative closeness of alternatives can be calculated by equation (16):

$$z_{ij} = \begin{bmatrix} 0 & 0.6490 & 1.0000 \\ 0.3744 & 0.2104 & 0.6367 \\ 1.0000 & 1.0000 & 0.4121 \\ 0.3982 & 0 & 0 \end{bmatrix}. \tag{24}$$

With reference to [46, 47], as well as taking p as the reference point, the S-shaped utility curve combined with the preference of decision-makers is constructed as follows:

$$u(t) = \frac{1}{6}t^3 - \frac{1}{2}pt^2 + \left(\frac{5}{6} + \frac{1}{2}p\right)t. \tag{25}$$

Chinese people's decision-making preference is mostly an S-shaped utility curve [44] if $p = 0.5$, which is positioned as the following:

$$u(t) = \frac{1}{6}t^3 - \frac{1}{4}t^2 + \frac{13}{12}t, \tag{26}$$

$$\text{that is, } \omega_i\left(z_j\right) = \frac{\omega_i^0\left((1/2)z_{ij}^2 - (1/2)z_{ij} + (13/12)\right)}{\sum_{l=1}^n \omega_l^0\left((1/2)z_{lj}^2 - (1/2)z_{lj} + (13/12)\right)}. \tag{27}$$

In equation (25), $i = 1, 2, \cdots, 3$ and $j = 1, 2, \cdots, 4$.

Then, the variable weights W can be calculated by equation (18):

$$W = \begin{bmatrix} 0.5018 & 0.3444 & 0.1538 \\ 0.4760 & 0.3779 & 0.1461 \\ 0.4904 & 0.3761 & 0.1335 \\ 0.4531 & 0.3907 & 0.1561 \end{bmatrix}. \tag{28}$$

(7) Calculating the weight loss matrix F by equation (19):

$$F = \begin{bmatrix} 0.0195 & -0.0255 & 0.0060 \\ -0.0063 & 0.0080 & -0.0017 \\ 0.0081 & 0.0062 & -0.0143 \\ -0.0292 & 0.0208 & 0.0083 \end{bmatrix}. \tag{29}$$

(8) Establishing the foreground theory matrix V by equation (20):

$$V = \begin{bmatrix} 0.0085 & -0.0533 & 0.0020 \\ -0.0128 & 0.0029 & -0.0034 \\ 0.0029 & 0.0021 & -0.0296 \\ -0.0612 & 0.0092 & 0.0030 \end{bmatrix}. \tag{30}$$

Table 4: Decision results of different reference points.

Utility functions	Reference points	Alternatives				Ranking of alternatives
		ξ_1	ξ_2	ξ_3	ξ_4	
Risky utility function	$p = 0$	−0.0179	−0.0101	−0.0005	−0.0430	$A_3 \succ A_2 \succ A_1 \succ A_4$
	$p = 0.25$	−0.0505	−0.0265	−0.0014	−0.0321	$A_3 \succ A_2 \succ A_4 \succ A_1$
S-shaped utility functions	$p = 0.5$	−0.0427	−0.0515	−0.0406	−0.1100	$A_3 \succ A_1 \succ A_2 \succ A_4$
	$p = 0.75$	−0.0406	−0.0254	−0.0001	−0.0720	$A_3 \succ A_2 \succ A_1 \succ A_4$
Conservative utility function	$p = 1$	−0.0652	−0.0392	−0.0008	−0.0967	$A_3 \succ A_2 \succ A_1 \succ A_4$

Table 5: Decision results of different methods.

Methods	Ranking of alternatives
Gupta et al. [27]	$A_4 \succ A_2 \succ A_1 \succ A_3$
Wei et al. [43]	$A_4 \succ A_2 \succ A_1 \succ A_3$
This article: not considering decision-maker's loss aversion	$A_4 \succ A_2 \succ A_1 \succ A_3$
This article: considering decision-maker's loss aversion ($p = 0.5$)	$A_3 \succ A_1 \succ A_2 \succ A_4$
This article: considering decision-maker's loss aversion ($p = 0.75$)	$A_3 \succ A_2 \succ A_1 \succ A_4$

(9) Ranking the alternatives according to R_{io}^*.

R_{io}^* is calculated by equation (21), and $R_{io}^* = [-0.0427 - 0.0515 - 0.0406 - 0.1100]$.

The alternatives can be ranked according to R_{io}^*; thus, $A_3 \succ A_1 \succ A_2 \succ A_4$. Therefore, the most desirable alternative is A_3.

4.2. The Sensitivity Analysis. In order to illustrate the influence of reference points on decision results, the sensitivity analysis results are shown in Table 4 by changing the values of p and repeating Steps (6) to (9). As can be seen from Table 4, in all the utility functions, the optimal alternative is the same, but the ranking of evaluation results at different reference points is different. Therefore, this method fully reflects the flexibility of the process and the limited rationality of the decision-maker.

4.3. Comparison and Analysis with Previous Methods. Compared with the grey target decision method proposed in [27] and the grey relational analysis (GRA) method based on DHFS proposed in [43], the comparison results are shown in Table 5. At the same time, an ablation study is given to analyze the impact of decision-maker's loss aversion on results. We calculate the results of the grey target decision method with positive and negative clouts without considering the loss aversion factor of decision-makers. The tackle method is described as follows. Calculating Step 1 to Step 5 in Section 3, and then substituting the weight calculated in Step 5 into Step 4 to get r_{io}^*; the alternatives can be sorted by r_{io}^*. The results show that, without considering the decision-maker's loss aversion, the decision results obtained by the three methods are the same. The reliability of the grey target decision method with positive and negative clouts and the DHFS proposed in this article is proved. After considering the decision-maker's loss aversion, the sorting results changed. Then, the decision results reflect the limited rationality and preference of decision-makers.

5. Conclusions

In this paper, a novel MAGDM problem based on the grey target decision method with positive and negative clouts under the dual hesitant fuzzy environment is proposed. An optimization model is established, which is employed to determine the initial weight vector of attributes. The optimization model is constructed with considering the proximity between the effect measures of alternatives, the positive ideal value, and the uncertainty of the attribute weight. The alternatives are sorted according to the comprehensive target-eye distance. Finally, a numerical example is presented to illustrate the application of the proposed method. The method provides a scientific and practical decision support to solve the MAGDM problem with double hesitation fuzzy numbers and decision-maker's loss aversion. This study is limited in that it cannot solve some new DHFSs, such as the probabilistic dual hesitant fuzzy set (PDHFS). Further research can introduce multiple new DHFSs to study the grey target group decision method.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was supported by the Social Sciences Foundation of Chongqing Municipality (no. 2016BS034), the National Natural Science Foundation of China (no. 71702015), the National Social Sciences Foundation of China (no. 18BGL007), the China Postdoctoral Science Foundation (no. 2017M611810), the Research Platform Open Project in CTBU (no. KFJJ2018079), Humanities and Social Sciences Research Program of Chongqing Education Commission

(no.18SKGH063), Science and Technology Research Projects of Chongqing Education Commission (no. KJQN201900812), and Technology Foresight and System Innovation Project of Chongqing Municipality (no. cstc2019jsyj-zzysbax0066).

References

[1] I. Deli and Y. Şubaş, "A ranking method of single valued neutrosophic numbers and its applications to multi-attribute decision making problems," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 4, pp. 1309–1322, 2017.

[2] H.-B. Yan, T. Ma, and V.-N. Huynh, "On qualitative multi-attribute group decision making and its consensus measure: a probability based perspective," *Omega*, vol. 70, pp. 94–117, 2017.

[3] S. Singh and M. Dasgupta, "Evaluation of research on CO 2 trans-critical work recovery expander using multi attribute decision making methods," *Renewable and Sustainable Energy Reviews*, vol. 59, pp. 119–129, 2016.

[4] W. Zhao, L. Yan, and Y. Zhang, "Geometric-constrained multi-view image matching method based on semi-global optimization," *Geo-spatial Information Science*, vol. 21, no. 2, pp. 115–126, 2016.

[5] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy, "Adaptive mobile activity recognition system with evolving data streams," *Neurocomputing*, vol. 150, pp. 304–317, 2015.

[6] M. S. Bakli, M. A. Sakr, and T. H. A. Soliman, "A spatio-temporal algebra in Hadoop for moving objects," *Geo-Spatial Information Science*, vol. 21, no. 2, pp. 102–114, 2018.

[7] M. M. Awad, "Forest mapping: a comparison between hyperspectral and multispectral images and technologies," *Journal of Forestry Research*, vol. 29, no. 5, pp. 1395–1405, 2017.

[8] A. Jalal, M. A. K. Quaid, and A. S. Hasan, "Wearable sensor-based human behavior understanding and recognition in daily life for smart environments," in *Proceedings of the 2018 International Conference on Frontiers of Information Technology (FIT)*, Islamabad, Pakistan, December 2018.

[9] A. Jalal, S. Kamal, and D. Kim, "A depth video sensor-based life-logging human activity recognition system for elderly care in smart indoor environments," *Sensors*, vol. 14, no. 7, pp. 11735–11759, 2014.

[10] P. Biswas, S. Pramanik, and B. C. Giri, "TOPSIS method for multi-attribute group decision-making under single-valued neutrosophic environment," *Neural Computing and Applications*, vol. 27, no. 3, pp. 727–737, 2016.

[11] X. Zhang and J. Su, "A combined fuzzy DEMATEL and TOPSIS approach for estimating participants in knowledge-intensive crowdsourcing," *Computers & Industrial Engineering*, vol. 137, Article ID 106085, 2019.

[12] Q. Pang, H. Wang, and Z. Xu, "Probabilistic linguistic term sets in multi-attribute group decision making," *Information Sciences*, vol. 369, pp. 128–143, 2016.

[13] A. Jalal, Y.-H. Kim, Y.-J. Kim, S. Kamal, and D. Kim, "Robust human activity recognition from depth video using spatio-temporal multi-fused features," *Pattern Recognition*, vol. 61, pp. 295–308, 2017.

[14] L. M. G. Fonseca, L. M. Namikawa, and E. F. Castejon, "Digital image processing in remote sensing," in *Proceedings of the 2009 Tutorials of the XXII Brazilian Symposium on Computer Graphics and Image Processing*, Rio de Janeiro, Brazil, October 2009.

[15] S. Kamal, A. Jalal, and D. Kim, "Depth images-based human detection, tracking and activity recognition using spatiotemporal features and modified HMM," *Journal of Electrical Engineering and Technology*, vol. 11, no. 6, pp. 1857–1862, 2016.

[16] K. Buys, C. Cagniart, A. Baksheev, T. De Laet, J. De Schutter, and C. Pantofaru, "An adaptable system for RGB-D based human body detection and pose estimation," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 39–52, 2014.

[17] A. Prochazka, M. Kolinova, J. Fiala, P. Hampl, and K. Hlavaty, "Satellite image processing and air pollution detection," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, Istanbul, Turkey, June 2000.

[18] M. M. U. Rathore, A. Ahmad, A. Paul, and J. Wu, "Real-time continuous feature extraction in large size satellite images," *Journal of Systems Architecture*, vol. 64, pp. 122–132, 2016.

[19] A. Farooq, A. Jalal, and S. Kamal, "Dense RGB-D map-based human tracking and activity recognition using skin joints features and self-organizing map," *KSII Transactions on Internet & Information Systems*, vol. 9, no. 5, 2015.

[20] Q. Huang, J. Yang, and Y. Qiao, "Person re-identification across multi-camera system based on local descriptors," in *Proceedings of the 2012 Sixth International Conference on Distributed Smart Cameras (ICDSC)*, IEEE, Hong Kong, China, pp. 1–6, 2012, October.

[21] S. Kamal and A. Jalal, "A hybrid feature extraction approach for human detection, tracking and activity recognition using depth sensors," *Arabian Journal for Science and Engineering*, vol. 41, no. 3, pp. 1043–1051, 2015.

[22] H. Yoshimoto, N. Date, and S. Yonemoto, "Vision-based real-time motion capture system using multiple cameras," in *Proceedings of the IEEE International Conference on Multi-sensor Fusion and Integration for Intelligent Systems, MFI2003*, Tokyo, Japan, August 2003.

[23] L. Piyathilaka and S. Kodagoda, "Gaussian mixture based HMM for human daily activity recognition using 3D skeleton features," in *Proceedings of the 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, Melbourne, Australia, June 2013.

[24] A. Jalal, M. A. Quaid, and M. A. Sidduqi, "A Triaxial acceleration-based human motion detection for ambient smart home system," in *Proceedings of the 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pp. 353–358, IEEE, Islamabad, Pakistan, 2019 January.

[25] A. Jalal, S. Kamal, and D. Kim, "Individual detection-tracking-recognition using depth activity images," in *Proceedings of the 2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Goyang, Republic of Korea, October 2015.

[26] H. Wu, W. Pan, X. Xiong, and S. Xu, "Human activity recognition based on the combined SVM&HMM," in *Proceedings of the 2014 IEEE International Conference on Information and Automation (ICIA)*, Hailar, China, July 2014.

[27] P. Gupta, M. K. Mehlawat, N. Grover, and W. Pedrycz, "Multi-attribute group decision making based on extended TOPSIS method under interval-valued intuitionistic fuzzy environment," *Applied Soft Computing*, vol. 69, pp. 554–567, 2018.

[28] D. Luo and X. Wang, "The multi-attribute grey target decision method for attribute value within three-parameter interval grey number," *Applied Mathematical Modelling*, vol. 36, no. 5, pp. 1957–1963, 2012.

[29] Y. Wang, X. Shi, J. Sun, and W. Qian, "A grey interval relational degree-based dynamic multiattribute decision making method and its application in investment decision making," *Mathematical Problems in Engineering*, vol. 2014, Article ID 307586, 6 pages, 2014.

[30] W. Qian, X. Yang, and J. Li, "Grey target decision model based on interval grey number type panel data and its application," *Journal of Grey System*, vol. 30, no. 1, pp. 69–80, 2018.

[31] F. Ye, "An extended TOPSIS method with interval-valued intuitionistic fuzzy numbers for virtual enterprise partner selection," *Expert Systems with Applications*, vol. 37, no. 10, pp. 7050–7055, 2010.

[32] S. Fu, G. Sun, and Y. Xiao, "Multi-attribute decision-making model of grey target for information system evaluation," *Computer Science and Information Technology*, vol. 4, no. 1, pp. 15–20, 2016.

[33] B. Zeng, C. Li, and S.-F. Liu, "A novel grey target decision-making model based on cobweb area and its application for choosing the software development pattern," *Scientia Iranica*, vol. 23, no. 1, pp. 361–373, 2016.

[34] R. M. Rodriguez, L. Martinez, and F. Herrera, "Hesitant fuzzy linguistic term sets for decision making," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 109–119, 2012.

[35] B. Zhu, Z. Xu, and M. Xia, "Dual hesitant fuzzy sets," *Journal of Applied Mathematics*, vol. 2012, Article ID 879629, 13 pages, 2012.

[36] H. Wang, X. Zhao, and G. Wei, "Dual hesitant fuzzy aggregation operators in multiple attribute decision making," *Journal of Intelligent & Fuzzy Systems*, vol. 26, no. 5, pp. 2281–2290, 2014.

[37] L. Wang, Q. Shen, and L. Zhu, "Dual hesitant fuzzy power aggregation operators based on Archimedean t-conorm and t-norm and their application to multiple attribute group decision making," *Applied Soft Computing*, vol. 38, pp. 23–50, 2016.

[38] Y. Xing, R. Zhang, M. Xia, and J. Wang, "Generalized point aggregation operators for dual hesitant fuzzy information," *Journal of Intelligent & Fuzzy Systems*, vol. 33, no. 1, pp. 515–527, 2017.

[39] D. Yu, W. Zhang, and G. Huang, "Dual hesitant fuzzy aggregation operators," *Technological and Economic Development of Economy*, vol. 22, no. 2, pp. 194–209, 2016.

[40] J. Yang, J. Su, and L. Song, "Selection of manufacturing enterprise innovation design Project based on consumer's green preferences," *Sustainability*, vol. 11, no. 5, p. 1375, 2019.

[41] Z. Ren and C. Wei, "A multi-attribute decision-making method with prioritization relationship and dual hesitant fuzzy decision information," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 3, pp. 755–763, 2017.

[42] Z. Ren, Z. Xu, and H. Wang, "Dual hesitant fuzzy VIKOR method for multi-criteria group decision making based on fuzzy measure and new comparison method," *Information Sciences*, vol. 388-389, pp. 1–16, 2017.

[43] G.-W. Wei, X.-R. Xu, and D.-X. Deng, "Interval-valued dual hesitant fuzzy linguistic geometric aggregation operators in multiple attribute decision making," *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 20, no. 4, pp. 189–196, 2016.

[44] M. Lu and G.-W. Wei, "Models for multiple attribute decision making with dual hesitant fuzzy uncertain linguistic information," *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 20, no. 4, pp. 217–227, 2016.

[45] S. Yang and Y. Ju, "A GRA method for investment alternative selection under dual hesitant fuzzy environment with incomplete weight information," *Journal of Intelligent & Fuzzy Systems*, vol. 28, no. 4, pp. 1533–1543, 2015.

[46] G. F. Yu, D. F. Li, J. Wu, and Y. F. Ye, "Heterogeneous multi-attribute variable weight decision making method considering decision maker's loss aversion," *Chinese Journal of Management Science*, vol. 26, no. 9, pp. 141–147, 2018.

[47] Y. F. Zhou and N. Chen, "The LAP under facility disruptions during early post-earthquake rescue using PSO-GA hybrid algorithm," *Fresenius Environmental Bulletin*, vol. 28, no. 12, pp. 9906–9914, 2019.

Hindawi
Scientific Programming
Volume 2020, Article ID 3810261, 20 pages
https://doi.org/10.1155/2020/3810261

Research Article

Deep Learning Structure for Cross-Domain Sentiment Classification Based on Improved Cross Entropy and Weight

Rong Fei,[1] **Quanzhu Yao**,[1] **Yuanbo Zhu**,[2] **Qingzheng Xu**,[3] **Aimin Li**,[1] **Haozheng Wu**,[1] **and Bo Hu**[4]

[1]*College of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China*
[2]*China Railway First Survey and Design Institute, Abu Dhabi 710043, China*
[3]*College of Information and Communication, National University of Defense Technology, Changsha, Hunan 710106, China*
[4]*Beijing Huadian Youkong Technology Co., Ltd., Beijing 100091, China*

Correspondence should be addressed to Rong Fei; annyfei@xaut.edu.cn and Quanzhu Yao; qzyao@xaut.edu.cn

Received 10 January 2020; Revised 22 May 2020; Accepted 3 June 2020; Published 29 June 2020

Academic Editor: Rahman Ali

Within the sentiment classification field, the convolutional neural network (CNN) and long short-term memory (LSTM) are praised for their classification and prediction performance, but their accuracy, loss rate, and time are not ideal. To this purpose, a deep learning structure combining the improved cross entropy and weight for word is proposed for solving cross-domain sentiment classification, which focuses on achieving better text sentiment classification by optimizing and improving recurrent neural network (RNN) and CNN. Firstly, we use the idea of hinge loss function (hinge loss) and the triplet loss function (triplet loss) to improve the cross entropy loss. The improved cross entropy loss function is combined with the CNN model and LSTM network which are tested in the two classification problems. Then, the LSTM binary-optimize (LSTM-BO) model and CNN binary-optimize (CNN-BO) model are proposed, which are more effective in fitting the predicted errors and preventing overfitting. Finally, considering the characteristics of the processing text of the recurrent neural network, the influence of input words for the final classification is analysed, which can obtain the importance of each word to the classification results. The experiment results show that within the same time, the proposed weight-recurrent neural network (W-RNN) model gives higher weight to words with stronger emotional tendency to reduce the loss of emotional information, which improves the accuracy of classification.

1. Introduction

Analysis of text emotional tendency, as an important research focus in the analysis of Internet public opinion, is mainly used to analyse and process subjective information, such as attitude, emotion, viewpoint, and tendency, in text. Sentiment analysis was first proposed by Pang et al. [1] for the positive or negative classification of movie reviews and Turney [2] for the positive or negative classification of cars and movies in 2002. Subsequent studies on sentiment analysis have been widely carried out for hotels, restaurants, product reviews, Weibo tweets, and other fields. Additional developments include positive or negative polarized classification methods [3], five classifications including ratings [4], and eight classifications including specific emotions [5].

Traditional sentiment analysis algorithms are mostly based on shallow machine learning, such as the maximum entropy model [6], conditional random field [7], support vector machine [8], and so on. With the increasing popularity of artificial intelligence, data-driven models have gradually become a focus on research of sentiment analysis models.

Deep learning algorithms have been widely used in the fields of speech, image, and natural language processing with their strong feature extraction and excellent information expression capabilities and have achieved better results than traditional models. In 1988, Rumelhart proposed the backpropagation neural network (BPNN) [6], which is a multilayer feedforward neural network (FNN) that uses the error backpropagation algorithm to adjust weights. It is the

most widely used NN model. LeCun et al. [9] used various deep neural networks to train language models at large-scale corpus level and constructed a probabilistic language model based on deep neural network, which solves common natural language processing tasks such as sentiment classification and part-of-speech tagging. Chen et al. proposed a deep learning method of learning potentially complex and irregular probability distributions, which can accurately estimate the values of cumulative distribution function (CDF) and probability density function (PDF) [10].

In the sentiment analysis task, the deep learning algorithm has also been widely used. At the same time, some people use convolutional networks to solve problems with the field of natural language processing and have achieved excellent results in tasks such as semantic analysis, query retrieval, and text classification. Since text is sequence data, there is a close relationship between words and characters as well. In 2006, Hinton [11] proposed a method for extracting features to the maximum extent and efficient learning, which has become a hotspot in deep learning research. Due to the excellent performance of deep learning in many fields, many researchers have begun to use deep learning for text sentiment analysis. Due to the long-term dependence of the cyclic neural network on the processing of long text tasks and the temporal information about words with the text, the LSTM is used [12, 13] for text emotion classification. Kennedy and Inkpen [14] considered the polarity transfer relationship of words in the text and determined the affective tendency by word counting based on the seed word set. Kim compared multiple deep learning models on multiple datasets and found that the experimental results of CNN were better than those of other methods [15]. Tang [16] considered the importance of user information and product information for sentiment classification, combined word vectors, user vectors, and product vectors at the input layer, and then used CNN for modelling and softmax for classification; the results were higher than those of the benchmark system at the sentence level and phrase level.

The difference between the predicted value and the real value of the model is usually evaluated by loss function, which generally tends to be the objective function in the classification or regression algorithms [17]. The smaller the loss function, the more its model can reflect the real data [18]. The closeness between the actual and the expected output is determined by cross entropy, which is essentially a measure of the difference between two codes [19]. Cross entropy is often the final loss function of machine learning or deep learning [20]. The closer the predicted distribution is to the real distribution, the smaller the value will be. With the wide application of cross entropy, in 2020, Cui et al. [21] applied a new loss function to optimize the end-to-end network for the first time, which is composed of binary cross entropy and dice coefficient; the best performance indexes can be achieved, thus verifying the validity of the model.

Deep learning methods have been applied to cross-domain sentiment mining tasks successfully with excellent representation learning and high efficiency classification abilities. Zhao et al. [22] presented a two-stage bidirectional LSTM (Bi-LSTM) and parameter transfer framework for short text cross-domain sentiment classification tasks. In 2019, Dey et al. [23] explored a three-step methodology, in which distinct balanced training, text preprocessing, and machine learning methods were tested, using two languages: English and Italian. In [24], cross-domain-labeled Web sources (Amazon and Tripadvisor) are used to train supervised learning models (including two deep learning algorithms) that are tested for typically unlabelled social media reviews (Facebook and Twitter), whose train model is tested on Facebook data for both English and Italian. In weight computing, Dey et al. [25] calculated the sentiment score of the n-grams by using the individual sentiment scores of the unigrams and precalculated values of intensifiers and negations attached with it. These scores are multiplied with the corresponding feature-importance value to generate the final score of SEND features of each review.

In the deep learning network model, the CNN has made great achievements in the field of image processing, whose convolution and pooling structure can extract image information very well. Therefore, RNN is widely used as a neural network for processing sequence data in the field of text analysis. Because of its memory function, it is better at processing sequence-changing data, among which, the LSTM recurrent neural network solves the problem of gradient disappearance and gradient explosion in the recurrent network, which makes the analysis and modelling of long sequence data successful. This study focuses on the optimization and improvement of RNN and CNN to achieve better text sentiment classification. According to the characteristics and shortcomings of each deep neural network, the following three text sentiment classification models are proposed.

Based on the CNN model and LSTM network, the ideas of the hinge loss and the triplet loss are used to improve the cross entropy loss used in the two classifications problems. The LSTM binary-optimize (LSTM-BO) model and CNN binary-optimize (CNN-BO) model are proposed which are more effective in fitting the predicted errors and preventing overfitting.

Considering the characteristics of the processing text of the recurrent neural network, the influence of input words of the final classification is analysed, which can obtain the importance of each word with the classification results. The proposed weight-recurrent neural network (W-RNN) model gives higher weight to words with stronger emotional tendency to reduce the loss of emotional information, which improves the accuracy of classification.

The rest of the paper is organized as follows. Section 2 is about the deep learning structure of cross-domain sentiment classification. Section 3 elaborates our numerical example. Results and discussion are presented in Section 4, and Section 5 is the summary of our research work.

2. Deep Learning Structure for Cross-Domain Sentiment Classification

2.1. The Improved LSTM-BO and CNN-BO Models

2.1.1. Hinge Loss Function and Triplet Loss Function. The hinge loss function is a loss function in the machine learning field and can be used for the "max-margin" classification,

often used to be the objective function of the SVM. Triplet loss is a loss function in the deep learning, which was originally proposed by Schroff et al. [26] to train less sensitive samples, such as face similarity measurement. The input of triplet loss is a triple a, p, n: a (anchor); p (positive) referring to a sample of the same category as a; n (negative) referring to a sample of a different category from a. The calculation of the sample similarity is achieved by optimizing the distance between a and p to be less than the distance between a and n. The formula as follows:

$$L = \max\left(d\left(a, p\right) - d\left(a, n\right) + \text{margin}, 0\right). \tag{1}$$

So, the ultimate optimization goal is to shorten the distance between a and p and extend the distance between a and n. The results are divided into three cases:

(i) Easy triplets: $L = 0$, that is, $d\left(a, p\right) + \text{margin} < d\left(a, n\right)$; this situation does not need to be optimized and meets the requirement that the distance of a and p is close, and the distance of a and n is far.

(ii) Hard triplets: $d\left(a, n\right) > d\left(a, p\right)$, that is, the distance between a and p is far.

(iii) Semihard triplets: $d\left(a, p\right) < d\left(a, n\right) < d\left(a, p\right) + \text{margin}$, that is, the distance between a and n is very close but has a boundary value margin.

2.1.2. The Improved Cross Entropy Loss Function. The text sentiment analysis task is fundamentally a classification problem. For the classification model, there will exist a problem that the optimization goal and the evaluation index are inconsistent. In the two-category task, the model uses cross entropy as the loss function, whose source is the maximum likelihood estimate. However, the final evaluation goal of the sentiment classification task is the accuracy of the model, instead of the size of the cross entropy. Usually, the cross entropy is small and the classification accuracy is high, but this relationship is not necessarily true.

In the two-category task, due to problems such as model fitting ability and data category imbalance, it is difficult for the model to achieve the positive sample output to be 1 and the negative sample output to be 0. In the actual prediction, the model considers that when the classification result is greater than 0.5, it is a positive sample, and when it is less than 0.5, it is a negative sample. This means that the model can be selectively updated. Therefore, an improved model is proposed in this paper: we set a threshold M, where M belongs to (0, 1). When the model's output of a positive sample is higher than M, or the output of a negative sample is lower than $1 - M$, the model will not be updated; the model will be updated only when the output of a sample is between M and $1 - M$, which can ensure that the model focuses on those samples that are not predictive. This can prevent the model from reducing the loss function and selecting those easy-to-fit samples to overtrain, making the model more effectively fit the samples whose prediction is wrong, thus improving the classification effect.

Based on the above model ideas, this passage references the thought of hinge loss and triplet loss to improve the loss function in the two-category model. The commonly used cross entropy loss function is formulated as follows:

$$L_{\text{old}} = -\sum_{y} y_{\text{true}} \log y_{\text{pred}}, \tag{2}$$

where y_{true} is the actual output result and y_{pred} is the expected value.

Select a threshold M and introduce the unit step function $\theta(x)$:

$$\theta(x) = \begin{cases} 1, & x > 0, \\ \dfrac{1}{2}, & x = 0, \\ 0, & x < 0. \end{cases} \tag{3}$$

Then, the new loss function is considered:

$$L_{\text{new}} = -\sum_{y} \lambda\left(y_{\text{true}}, y_{\text{pred}}\right) y_{\text{true}} \log y_{\text{pred}}, \tag{4}$$

where

$$\begin{aligned} \lambda\left(y_{\text{true}}, y_{\text{pred}}\right) = 1 &- \theta y_{\text{true}} - m\theta\left(y_{\text{pred}} - m\right) \\ &- \theta\left(1 - m - y_{\text{true}}\right)\theta\left(1 - m - y_{\text{pred}}\right), \end{aligned} \tag{5}$$

where L_{new} adds corrections $\lambda\left(y_{\text{true}}, y_{\text{pred}}\right)$ to the cross entropy, which means that while entering a positive sample, $y_{\text{true}} = 1$. Apparently, $\lambda\left(1, y_{\text{pred}}\right) = 1 - \theta\left(y_{\text{pred}} - m\right)$ at this time; if $y_{\text{pred}} > m$, $\lambda\left(1, y_{\text{pred}}\right) = 0$ will be established, and the cross entropy will automatically be 0 (reaching the minimum). On the contrary, if $y_{\text{pred}} < m$, then $\lambda\left(1, y_{\text{pred}}\right) = 1$; at this time, the cross entropy is maintained. That is to say, if the positive sample is higher than m, then it will not be updated. If it is less than m, it will continue to update; similarly, the negative sample can be analysed. As a result, the conclusion is that if the output is already lower than $1 - m$, then it will not be updated, and if it is higher than $1 - m$, it will continue to update.

2.1.3. The LSTM-BO and CNN-BO Models. The LSTM-BO and CNN-BO models are based on the LSTM and CNN in keras which are combined with improved cross entropy loss function described in Section 2.1.2.

2.2. Weight-Recurrent Neural Network (W-RNN) for Cross-Domain Sentiment Classification

2.2.1. Basic Structure of Recurrent Neural Network. Recurrent neural network (RNN) refers to the network structure that processes input data sequences in the same structure over time [27]. The proposed RNN effectively solves the problem of processing sequence information. In traditional neural networks, nodes inside the hidden layer are connectionless and each output is independent of one another. However, in RNN, the hidden layer nodes are

connected to each other in the time dimension, and the input of each node includes not only the input currently input by the input layer but also the output information of the previous state of the hidden layer, that is, the network can remember the previous information, used to calculate the current output, as shown in Figure 1.

The characteristic that the RNN can recall the previous information is based on the hidden layer, which is constantly repeated as a memory unit and saves the information from the previous state. As a logical structure, the internal structure of the memory unit is shown in Figure 2. At the time t, weight W of the input x_t and old information $h_t - 1$ from the time $t - 1$ are processed by the self-joining matrix U through the hidden layer, whose sum adds the offset b together to obtain the output h_t of the current state hidden layer via an activation function (such as, Tan$h$).

The time t's and previous information continues to propagate through h_t until the end which is the true output of the hidden layer. The above calculation process is formulated as

$$h_t = \mathrm{Tan}h\left(Wx_t + Uh_{t-1} + b\right). \tag{6}$$

2.2.2. W-RNN Model for Cross-Domain Sentiment Classification. It can be acknowledged that in the standard recurrent neural network, the calculation result of the recurrent unit at each moment is not utilized, but only passed to the recurrent unit at the next moment for another round of calculation until the last moment, whose output is used as the output of the recurrent layer. The calculation of the state of each moment depends on the previous moment, and the temporal order information is continuously retained by such dependence. In this section, combined with the characteristics of the recurrent neural network, the importance of each word will be obtained by analysing the influence of the input words on the final classification. Based on this thought, the W-RNN model is constructed, which provides higher weight for words with stronger emotional tendency and reduces the loss of text emotional information, thus improving the accuracy of text sentiment classification.

Recurrent neural networks are one of the most important models of many sequence tasks. The common methods for text classification tasks are shown in Figure 3.

How to measure the importance of the impact of input $w_1, \ldots, w_n$ on the final classification results? Assuming that it is an emotional classification task, firstly, the words which have a more important impact on the final classification need to be found.

Because the state vector of the last step of the RNN (the vector represented by the orange shade in Figure 3) is passed to the subsequent classifier for classification, the state vector h_n of the last step is a target vector. RNN is a recursive process, and $h_0, h_1, \ldots, h_{n-1}$ is gradually approaching h_n.

So, the distance $\|h_n - h_0\|, \|h_n - h_1\|, \ldots, \|h_n - h_{n-1}\|, 0$ of the intermediate vector to the target vector can be considered in turn. From h_i to h_{i+1}, because of the excessive consideration of the word w_{i+1}, it can be deduced that the distance between h_i and the target vector should be $\|h_n - h_i\|$,

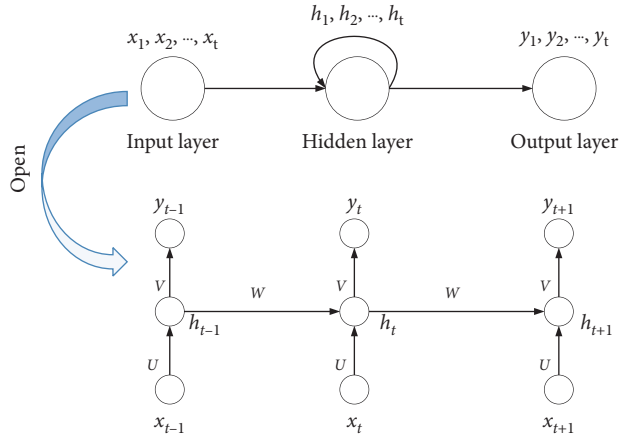


Figure 1: The structure diagram of recurrent neural network model.

but now the distance becomes $\|h_n - h_{i+1}\|$, so we can use the difference $\|h_n - h_i\| - \|h_n - h_{i+1}\|$ to measure the impact of the word w_{i+1} on the final classification. If the difference is positive, it means that the introduction of w_{i+1} narrows the distance from the target, so it promotes the correct classification; otherwise, if the difference is negative, it means that it has a reaction to the classification; the larger the value, the greater the degree of effect. So, this indicator can be used to sort in descending order and get the importance of each word. This article excludes the effects of dimension by dividing by the norm of the target vector:

$$\frac{\left\|\left\|h_n - h\right\|_i}{\left\|\left\|h_n\right\|\right\|} - \frac{\left\|\left\|h_n - h_i + 1\right\|\right\|}{\left\|h_n\right\|}. \tag{7}$$

2.3. Algorithm Flow. According to the above model structure, we can get the algorithm flow of emotion classification by using W-RNN model (Algorithm 1).

Batch_size is the scale, used to group C_{TR} for training, and the left samples with less than Batch_size are grouped together; epochs represent the number of training iterations.

In this algorithm, firstly, distances are ranked according to h'. Secondly, every secondary feature c is calculated from the word vector. Then, the output z is obtained by c, and the z value is used to update the weight. The W-RNN gives high weight to words with strong emotional tendency and reduces the function of words with weak emotion in sentences, which decreases the loss of text emotional information.

3. Experiments and Setting

3.1. Experimental Environment. The specific experimental environment configuration of this model is shown in Table 1.

3.2. Dataset. Since the classification model may possess different adaptabilities of different languages and texts of different lengths, in order to verify the performance of the model, the experiment was tested with different types of datasets under several famous corpora. This

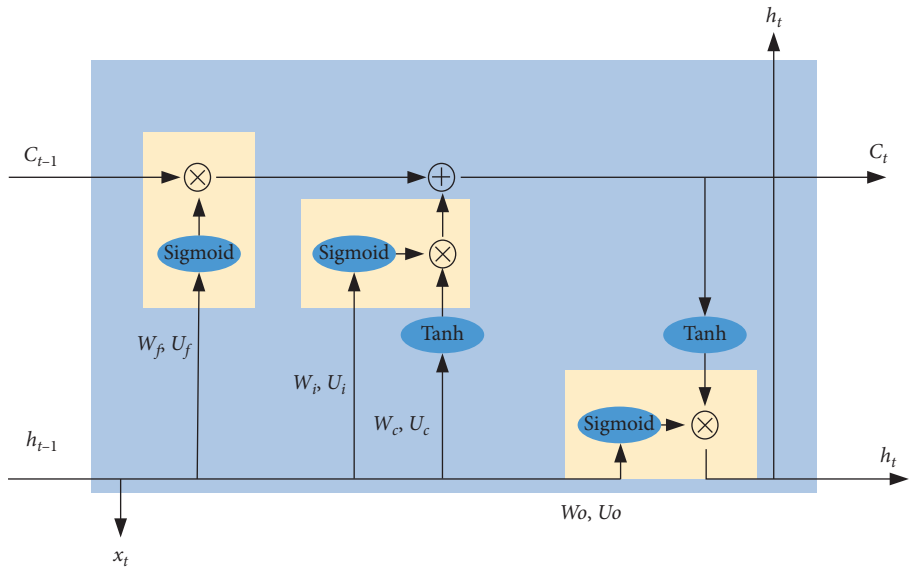


Figure 2: Structure of a standard RNN unit.

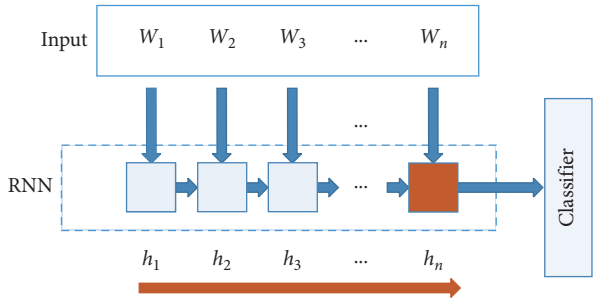


Figure 3: Text classification model based on recurrent neural network.

experiment uses IMDB [28] English film review data and Netease news classified text data, covering different languages, different lengths, and different types of text classification tasks.

The following two datasets are specifically described.

The IMDB English film review dataset is data from Amazon's Internet Movie Database (IMDB), which includes a lot of information about the film, such as actors, film length, content introductions, ratings, and reviews. For the text classification task, the film review data used in this experiment distinguish positive review from negative review, that is, they contain two categories, which belong to the two-category sentiment analysis classification problem. The dataset contains a total of 50,000 comment text data, and its label distribution is balanced, that is, there are 25,000 positive reviews and 25,000 negative reviews. In addition, the dataset provides 50,000 unlabelled data for unsupervised learning.

The Chinese dataset is a collection of commodity reviews provided by Data Hall, which contains six aspects of comment data (books, hotels, computers, milk, mobile phones, and water heaters), a total of 21,107 text data, in which there are 10,428 negative data and 10,679 positive data. Figure 4 describes the above Chinese dataset.

In Table 2, the number of positive emotion samples and negative emotion samples and sum of samples for two datasets are given.

3.3. Data Preprocessing. Text datasets used in the experiment are both in Chinese and in English. The Chinese dataset is a collection of commodity review corpus provided by Data Hall, which contains six aspects of comment data, a total of 21,107 text data. The English dataset is the IMDB film review dataset, with a total of 50,000 comment text data. In the sentiment classification task, the training data and test data of the two datasets are randomly generated in a ratio of 80 : 20.

The preprocessing work on the data mainly includes the cleaning of invalid special characters and punctuation, the cleaning of common pause words in the language, and the segmentation of Chinese language using the jieba word segmentation tool based on python. This experiment introduces Word2Vec as pretraining, aiming to construct the word vector. The appropriate word vector can improve the performance and calculation speed of the model. Every word vector dimension pretrained by Word2Vec is set to 50, with the window size set to 10 and trained by the skip-gram model. The training parameter settings of the Word2vec model are described in Table 3.

3.4. Superparameter Setting. In the neural network model training process, the superparameter is a parameter that sets the value before the model training. Generally, the superparameter needs to be optimized, and a set of optimal superparameter is selected for the model to improve the performance and effect of the learning. The superparameter configuration of the model is shown in Table 4.

Table 5 shows the superparameter settings of each model, among which the selection of superparameter is obtained by the optimization experiment of the bold font parameters of Table 6.

Input:
CWE-word vector
CTR-training corpus
CTE-test corpus
Output: Prediction results of test samples.
(1) pro_processing (CWE)
(2) Dict = word2vec (CWE)//create the word vector dictionary Dict
(3) batches [] ⟵ Divide (CTR)//divide CTR into several batches
(4) for i ⟵ 0 to epochs do
(5) for j ⟵ 0 to length (batches) do
(6) for k ⟵ 0 to length (batches[j]) do
(7) v ⟵ FindWord (batches [j][k])//find the words vector v in batches[j][k] from Dict
(8) h ⟵ v//the feature vector h is extracted from v
(9) h' ⟵ Measure (h)//measure the impact of h
(10) v ⟵ Sort (v, h')//sort words vector in descending order according to h'
(11) c ⟵ ExtractFeature (v)//extract secondary feature c from the word vector
(12) z ⟵ Softmax (c)//Get the prediction results of samples by Softmax classifier
(13) end for
(14) Update (z, w, (b)//update parameters w and b of the model by backpropagation
(15) end for
(16) end for
(17) for i ⟵ 0 to length (CTE) do
(18) v ⟵ FindWord (CTE [i])
(19) h ⟵ v
(20) h' ⟵ Measure (h)
(21) v ⟵ Sort (v, h')
(22) c ⟵ ExtractFeature (v)
(23) output ⟵ Softmax (c)
(24) end for

Algorithm 1: Text sentiment analysis algorithm based on W-RNN.

Table 1: Experimental environment and configuration.

Experimental environment	Specific configuration
Operating system	Windows 10 × 64
CPU	Intel (R) Core (TM) i7-4790, 3.6 GHz
Memory	8G
Hardware	1T
Programming language	Python 3.5
Deep learning framework	keras2.0
Word vector training tool	Word2vec
Word segmentation tool	Jieba

4. Result Analysis

4.1. Analysis of Experimental Results Based on LSTM-BO and CNN-BO Models. The loss function of the traditional LSTM model and the CNN model improved, and the LSTM-BO and CNN-BO models constructed to perform the text sentiment analysis task, the parameter optimization experiment, and the benchmark model comparison experiment are used to verify the effectiveness of the new network model in the emotion classification task.

4.1.1. Parameter Optimization Experiment. In order to investigate the influence of each parameter on the model effect, in this paper, four sets of parameter optimization experiments are designed and compared on the IMDB public dataset.

(i) Discussion of experimental results of threshold M selection

In this experiment, based on the LSTM-BO and CNN-BO models, the threshold M is selected from 0.5 to 1.0, increasing by 0.1 in turn. The experimental results are shown in Table 6.

It can be seen from Table 6 that the LSTM-BO and CNN-BO models have the highest accuracy at the threshold M = 0.6, which are 82.14% and 88.74%, respectively; the loss rates are the same when M = 0.6 and M = 0.7, which are, respectively, 0.2267 and 0.1637. It can be seen from Figures 5 and 6 that with the increase of the M value, the accuracy of the two models generally shows a trend of increasing first and then decreasing, and the overall loss rate tends to decrease first and then rise. When the threshold M is 0.6, the accuracy of the LSTM-BO model reaches the peak value, and the loss rate reaches the minimum value; when the M value changes from 0.5 to 0.6, the accuracy rate is greatly improved, which is 7.96% higher; when the M value is changed from 0.9 to 1.0, the loss rate changes greatly, increasing by 0.4435. The accuracy and loss rate of the CNN-BO model are similar to those of the

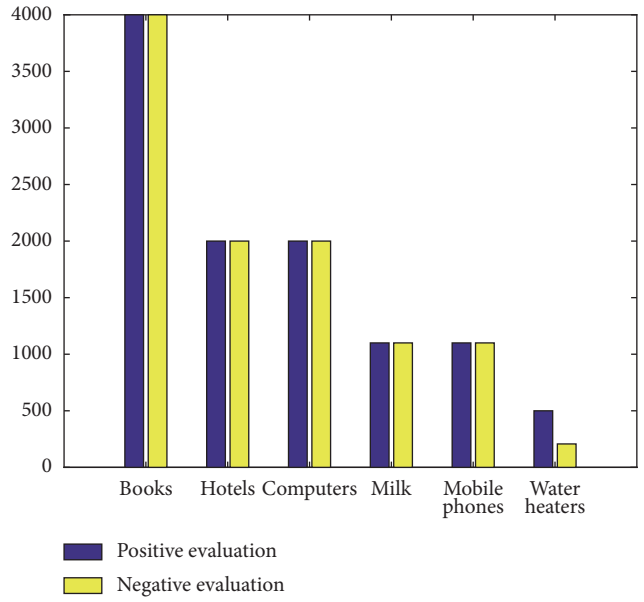


Figure 4: Chinese dataset category.

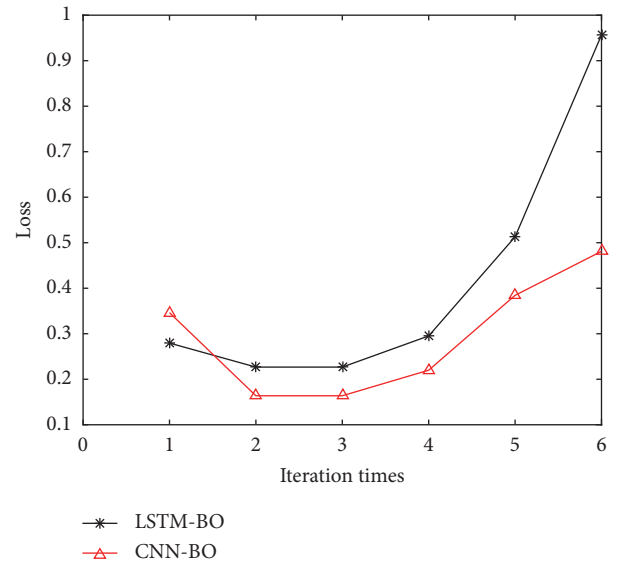


Figure 6: Change rate of each model under different thresholds.

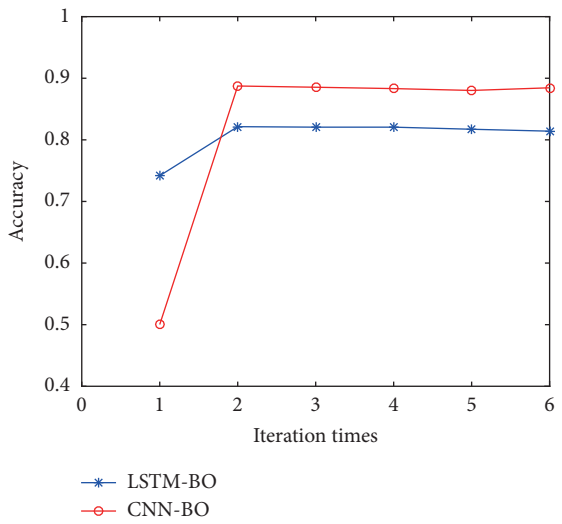


Figure 5: Variation rate of each model classification under different thresholds.

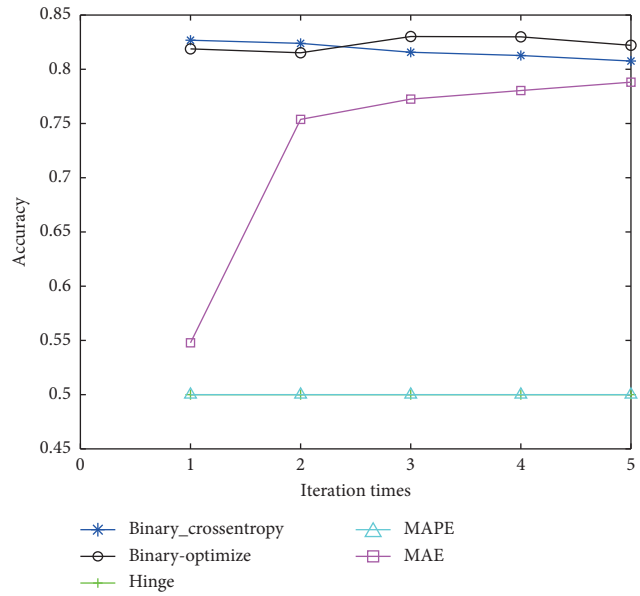


Figure 7: The variation diagram of accuracy rate with the number of iterations under different loss functions.

LSTM-BO model, but the overall effect is better. When $M = 0.6$, the accuracy reaches the peak and the loss rate is the smallest. When the M value is changed from 0.5 to 0.6, the accuracy rate is increased by 38.71%, and the loss rate is reduced by 0.1827. Based on the above analysis, the threshold M of this paper is 0.6.

(ii) Influence of different loss functions on the model

In Table 7, binary_crossentropy is a standard cross entropy loss function; binary-optimize is the loss function proposed in this paper; hinge is a hinge loss function, commonly used in SVM classifiers; mean_absolute_percentage (MAPE) is the average absolute percentage error loss function; mean_absolute_error (MAE) is the absolute value variance loss function. Figure 7 shows the variation of the accuracy of the LSTM model with the number of iterations under different loss functions of the English dataset. It can be seen from the figure that the LSTM-BO model with improved loss function has the highest accuracy in the sentiment classification task, which is 82.21% and has been leading after the second iteration; the LSTM model using the standard cross entropy loss function with the hinge, MAPE, and MAE loss functions has high

Table 2: Dataset table.

Data	Positive emotion sample	Negative emotion sample	Sum of samples
English	25000	25000	50000
Chinese	10679	10428	21107

Table 3: Word2vec model training parameter settings.

Parameter	Value
Vector dimension	50
Context window	10
Sampling value	$1e-3$
Iteration times	4
Minimum number of occurrences	5
Algorithm	Skip-gram
Worker number	4

accuracy. The accuracy of the LSTM model using the hinge and MAPE loss functions is kept at 50% and the number of iterations is more than 5 times. Based on the above experimental results, the effectiveness of the improved loss function is proved.

(iii) The selection experiment of word vector dimension

In this experiment, the vector dimension of the selected words is 50, 100, 150, 200, 250, and 300, respectively. It can be seen from Table 8 that the LSTM-BO model has the highest accuracy of 82.48%, when the word vector dimension is 100 dimensions, and the loss rate is at least 0.2234. When the word vector dimension is 50 dimensions, the CNN-BO model has a maximum accuracy of 88.74% and a loss rate of 0.1637.

The dropout technique weakens the coadapting property of adjacent elements in the same layer by randomly discarding certain elements in the previous layer during the training process. By using dropout, the overfitting phenomenon is significantly reduced, and thus it is widely used in the training process of deep learning. In order to study the impact of dropout on the training process, this experiment sets the value of dropout to a series of different values during each training process with other parameters fixed. The results are shown in Table 9.

It can be seen from Table 9 that when dropout was set to 0.2, the LSTM-BO model achieved the highest accuracy of 82.14%, the minimum loss rate of 0.2247, and the shortest time consumption. When dropout was set to 0.2, the LSTM-BO model has a maximum accuracy of 88.74% and the shortest time consumption. When dropout is 0.3, the lowest loss rate is 0.1590.

4.1.2. Comparison of Experimental Results. In order to verify the validity of the LSTM-BO and CNN-BO models, based on the Chinese and English datasets, the results were compared with the emotional classification results of the benchmark models LSTM and CNN. The experimental results are shown in Table 10.

(1) Accuracy Analysis. Figures 8 and 9 show the accuracy of each model in the training set and test set sentiment classification tasks under different datasets. The horizontal axis represents the number of iterations and the vertical axis represents the accuracy. The blue curve represents the change in the accuracy of the training set, and the orange curve represents the change in the accuracy of the test set. Figure 10 shows a bar graph of the accuracy of the final test set for each model on different datasets. Figure 11 shows a plot of the change in the accuracy of each model in the 10 iterations of the English dataset. From the above experimental results, the following conclusions can be drawn:

(i) Table 10 shows that there exist some differences in the experimental results of different datasets. The accuracy of LSTM and LSTM-BO models in Chinese datasets is 5.94% and 5.37% higher than that of English datasets, respectively. The accuracy of CNN and BO models on the Chinese dataset is 0.5% and 0.17% higher than the English datasets, respectively.

(ii) It can be seen from Table 10 and Figure 10 that the CNN-BO model has the highest accuracy of emotional classification in the Chinese and English datasets compared with the other three models, which are 88.91% and 88.74%, respectively. The accuracy of the LSTM-BO model is 1.16% and 0.59% higher than that of the benchmark LSTM model in the English dataset and the Chinese dataset, respectively. The accuracy of the CNN-BO model is higher than that of the benchmark CNN model in the English dataset and the Chinese dataset which is increased by 0.5% and 0.07%.

(iii) As can be seen from Figures 8 and 9, the accuracy of the four models on the training set increases slowly with the number of iterations, and there is a significant change between the first and second iterations, and eventually it stabilizes. However, the increase in the number of iterations cannot increase the accuracy in the test set signficantly. The fluctuation exists in the test, especially the LSTM and CNN models fluctuate greatly in the test. It can be seen from Figure 11 that in the English dataset, the LSTM-BO model has high accuracy after the second iteration, and the CNN-BO model has high accuracy after the fourth iteration. The analysis results in this paper are that LSTM-BO and CNN-BO models can more effectively fit the samples that predict errors, help

Table 4: Model parameter configuration.

Name	Description
Batch_size	Number of batch training samples
Hidden_dam	Hidden layer nodes
Embedding_dam	Word vector dimension
Filters	Filter size
Kernel_size	Number of convolution kernels
Max_features	The maximum characteristic number of word bag model
Epoch	Times of model iterations
Min_count	Minimum word frequency

Table 5: Model superparameter settings.

	LSTM	LSTM-BO	CNN	CNN-BO	W-RNN
Hidden_dam	256	256	256	256	256
Batch_size	32	32	32	32	32
Embedding_dam	50	50	50	50	50
Filters	—	—	250	250	—
Kernel_size	—	—	3	3	—
Max_features	5000	5000	5000	5000	5000
Epoch	10	10	10	10	10
Min_count	10	10	10	10	10
Dropout	0.2	0.2	0.2	0.2	0.5
Activation function	Sigmoid	Sigmoid	Relu	Relu	Sigmoid

Table 6: Accuracy and loss function of each model under different thresholds (M).

	LSTM-BO		CNN-BO	
	Accuracy (%)	Loss (%)	Accuracy (%)	Loss (%)
0.5	74.18	0.2795	50.03	0.3464
0.6	**82.14**	**0.2267**	**88.74**	**0.1637**
0.7	82.06	0.2267	88.54	0.1637
0.8	82.06	0.2952	88.32	0.2199
0.9	81.72	0.5129	88.02	0.3844
1.0	81.41	0.9564	88.45	0.4811

Table 7: The variation of accuracy rate with the number of iterations under different loss functions.

Times loss function	Iteration				
	1	2	3	4	5
Binary_crossentropy	82.68	82.39	81.57	81.27	80.76
Binary-optimize	81.88	81.52	83.02	82.99	82.21
Hinge	50.00	50.00	50.00	50.00	50.00
MAPE	50.00	50.00	50.00	50.00	50.00
MAE	54.78	75.38	77.25	78.04	78.81

prevent overfitting, and improve the accuracy of sentiment classification tasks.

(2) Loss Rate Analysis. Figures 12 and 13 show the variation curves of the loss rates of the models in the training set and test set sentiment classification tasks in different datasets, respectively. The horizontal axis represents the number of iterations and the vertical axis represents the accuracy. The black curve represents the change in the training set loss rate, and the yellow curve represents the test set loss rate change. Figure 13 shows a plot of the change in loss rate for each model over 10 iterations in the English dataset. From the above experimental results, the following conclusions can be drawn:

(i) It can be seen from Figures 12 and 13 that the loss rate of the four models in the Chinese and English training sets decreases with the increase of the number of iterations, but the change of the model on the test set is more obvious. The loss rate of the LSTM model and the CNN model in the test set fluctuated greatly and showed an upward trend. The loss rate of the LSTM-BO and CNN-BO models on the test set slowly decreased and finally tended to be flat.

(ii) It can be seen from Figure 14 that after the second iteration of the English dataset, the loss rate of each model has a large trend, and the loss rate of LSTM and CNN models increases with the number of iterations and finally ends after the10th iteration; they were 0.8170 and 0.5588, respectively. The LSTM-BO and CNN-BO models slowly decrease as the number of iterations increases, eventually reaching a steady state. The loss ratios of the LSTM-BO and CNN-BO models were reduced by 0.5903 and 0.3951, respectively, over the benchmark LSTM and CNN models. The analysis results in this paper are that the improved model has better generalization ability, which can converge after multiple iterations and achieve lower loss rate.

(3) Time Performance Analysis. Figure 15 shows the time consumption of each model in the Chinese and English datasets. As can be seen from the figure, the LSTM-BO is reduced by 162 seconds and 51 seconds, respectively, in the Chinese and English datasets compared to the LSTM model; CNN-BO is reduced by 128 seconds and 202 seconds, respectively, compared to the CNN model. The analysis results in this paper are that for the LSTM-BO and CNN-BO models, the samples will not be updated when the predicted value of the positive sample is higher than M or the predicted value of the negative sample is lower than M, focusing on those whose predictions are not accurate, thus reducing the

Table 8: Experimental results under different word vector dimensions.

	Word vector dimension	50	100	150	200	250
LSTM-BO	Accuracy (%)	82.14	**82.48**	81.69	81.76	82.04
	Loss (%)	0.2267	**0.2234**	0.2372	0.2577	0.2302
CNN-BO	Accuracy (%)	**88.74**	87.66	88.56	88.34	87.46
	Loss (%)	**0.1637**	0.1845	0.1652	0.1732	0.1832

Table 9: Results of different dropout values.

	Dropout	Accuracy (%)	Loss (%)	Time (s)
LSTM-BO	0.1	81.62	0.2348	1265
	0.2	**82.14**	**0.2267**	**1136**
	0.3	82.06	0.2285	1148
	0.4	82.05	0.2274	1143
	0.5	81.98	0.2354	1254
CNN-BO	0.1	87.72	0.1833	1249
	0.2	**88.74**	**0.1637**	**1172**
	0.3	88.72	0.1590	1184
	0.4	88.39	0.1684	1227
	0.5	87.43	0.1842	1265

Table 10: Comparison of classification results on different network models.

Dataset	Network model	Accuracy (%)	Loss (%)	Time (s)
English dataset	LSTM	80.98	0.8170	1187
	LSTM-BO	**82.14**	**0.2267**	**1136**
	CNN	88.34	0.5588	1374
	CNN-BO	**88.74**	**0.1637**	**1172**
Chinese dataset	LSTM	86.92	0.6820	1425
	LSTM-BO	**87.51**	**0.1705**	**1263**
	CNN	88.84	0.5874	1523
	CNN-BO	**88.91**	**0.2231**	**1395**

consumption of time. The calculation amount is decreased with the preprocessing of Chinese data by removing the useless words and punctuations, but the same operation has not done on the English data.

4.2. Analysis of Results Based on W-RNN Model. To qualitatively and quantitatively evaluate the W-RNN model proposed in this section, this experiment compares the effects of different models in the emotional analysis task under the Chinese and English datasets. The specific method is as follows: for the quantitative evaluation experiment, some data are selected from the Chinese and English datasets as the training set, the classification model is trained, and finally the emotional classification task is completed in the test set to measure the accuracy; for the qualitative evaluation experiment, the emotional weight calculated by the analysis model is to verify the validity of the model.

4.2.1. Results and Discussion for Qualitative Experiments. In the qualitative analysis experiment, firstly, the classification model is trained in the training set. Then, we randomly select 3 comment corpora under the English and Chinese datasets and process them by the trained W-RNN model and generate the results in the table. In the result, the first component in the brackets indicates the position of the word after the word segmentation, the second component indicates the word segmentation, and the third component indicates the weight of the word. For example, the first component in the first row of the table: (7, "poor", 0.36663848) is the result of word "poor", in which 7 means "poor" is the seventh position after sentence participle, "poor" means the word itself, 0.36663848 indicates the weighting factor of "poor".

The following results can be obtained from the above experimental results:

(i) It can be seen from Table 11 that the W-RNN sentiment classification model proposed in this paper ranks the words with strong emotional tendencies in the front and gives higher weights. For example, in the second sentence, the word "good" is given a weight of 0.2046071, and the word "very" in the fourth sentence is given a weight of 0.17955697.

(ii) The evaluation scheme of such importance will automatically consider the influence of the position of the words. If an emotional word is repeated in the sentence, the words appearing later are generally weighted lower.

4.2.2. Results and Discussion for Quantitative Experiments. In order to verify the effectiveness of the W-RNN model, based on the Chinese and English datasets, the results are compared with the emotional classification results of the benchmark model RNN. The experimental results are shown in Table 12.

(i) Accuracy analysis

Figure 16 shows the performance of each model in the training set and test set sentiment classification tasks in different datasets. The horizontal axis represents the number of iterations and the vertical axis represents the accuracy. The blue curve represents the change in the accuracy of the training set, and the orange curve represents the change in the accuracy of the test set. Figure 17 shows a bar graph of the accuracy of the final test set for each model on different datasets. The following results can be obtained from the above experimental results:

(i) Table 12 shows that the experimental results of different datasets have certain differences. The accuracy of RNN and W-RNN models in Chinese datasets is 6.53% and 8.12% higher than that of English datasets, respectively.

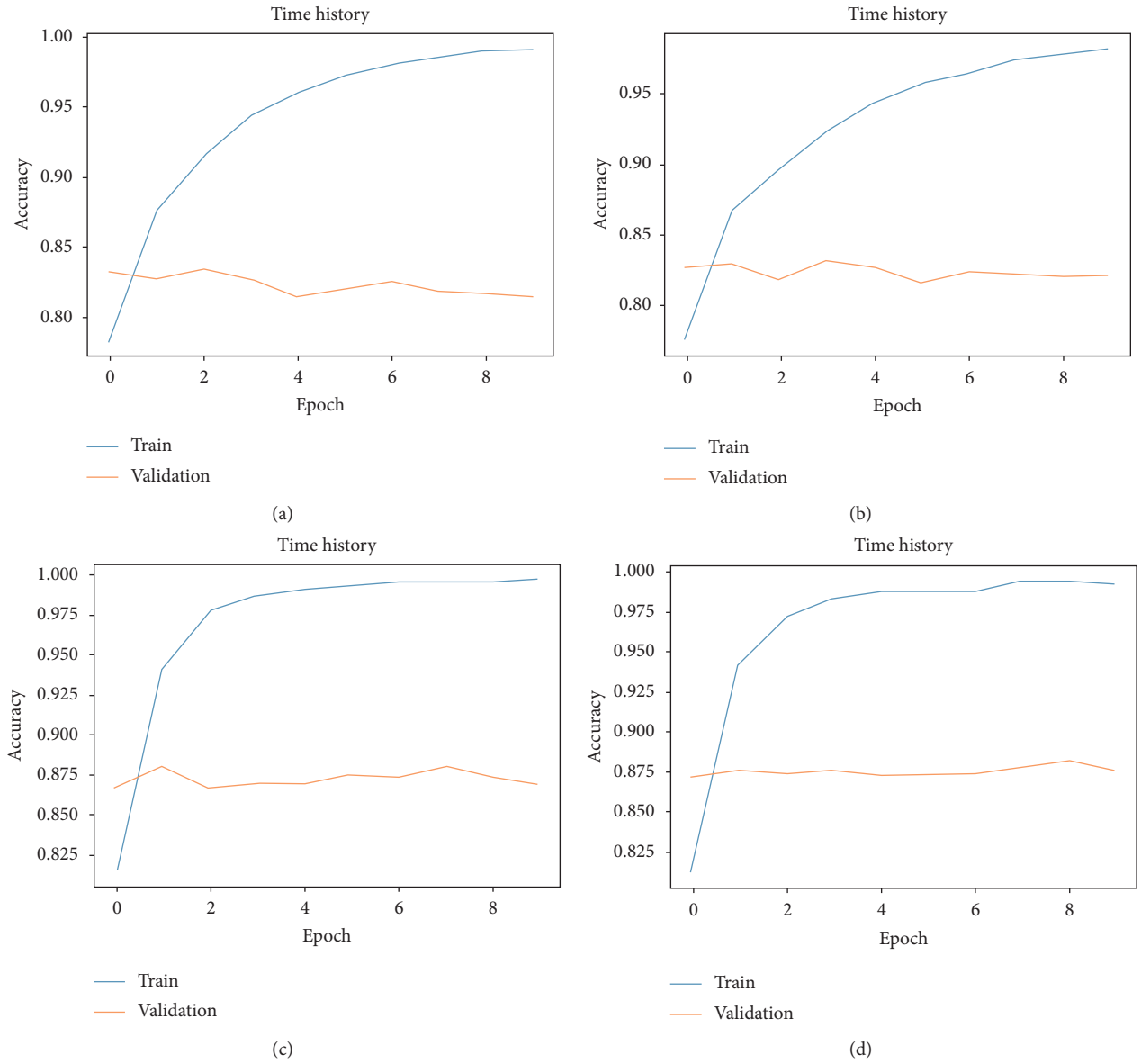


Figure 8: Accuracy change of LSTM model and LSTM-BO model in Chinese and English datasets. (a) LSTM (English dataset). (b) LSTM-BO (English dataset). (c) LSTM (Chinese dataset). (d) LSTM-BO (Chinese dataset).

(ii) Figure 16 shows that the accuracy of the two models in the Chinese and English training sets increases slowly with the number of iterations and eventually stabilizes. The accuracy rate is flat on the test set. After 10 iterations, the accuracy of the W-RNN model over the benchmark RNN model in the English dataset and the Chinese dataset increased by 1.56% and 3.19%, respectively. The analysis result of this paper is that W-RNN model can analyse the influence of input words on the final classification, assign higher weight to words with stronger emotional tendency, and reduce the loss of emotional information, thus improving the accuracy of text sentiment classification.

(ii) Loss analysis

Figure 18 shows the variation of the loss of the RNN model and the W-RNN model in different iterations of the Chinese and English datasets. The black curve represents the training set and the yellow curve represents the test set. It can be seen from the figure that the loss of the two models in the Chinese and English training sets shows a decreasing trend with the increase of the number of iterations and finally reaches a lower value and becomes stable. However, the difference between the two models in the test set is that the loss of the W-RNN model after 10 iterations in the English dataset is 0.1500, which is 0.3670 lower than the RNN model; the loss of the W-RNN model after 10

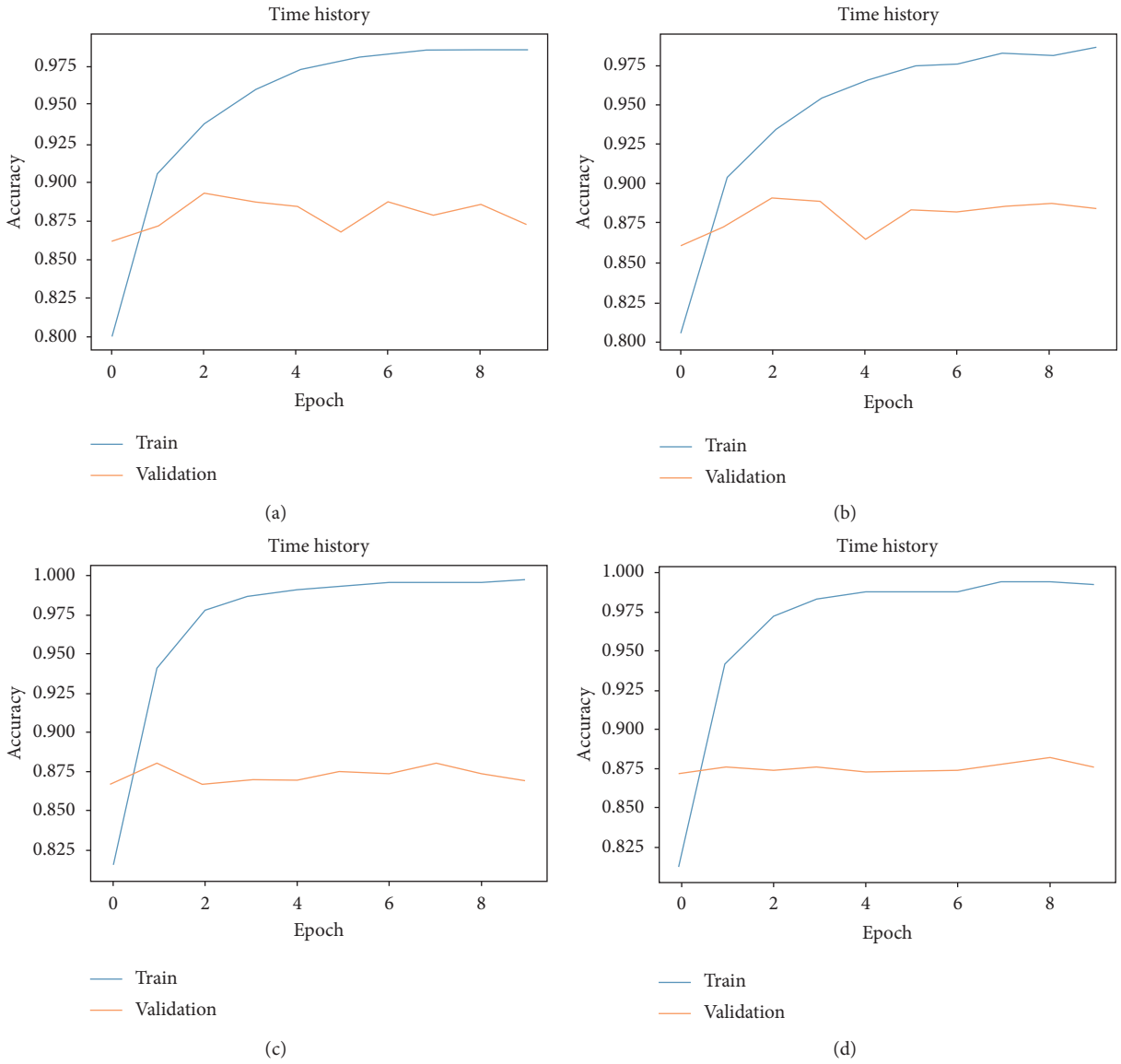


Figure 9: Emotional classification accuracy of CNN model and CNN-BO model in Chinese and English datasets. (a) CNN (English dataset). (b) CNN-BO (English dataset). (c) CNN (Chinese dataset). (d) CNN-BO (Chinese dataset).

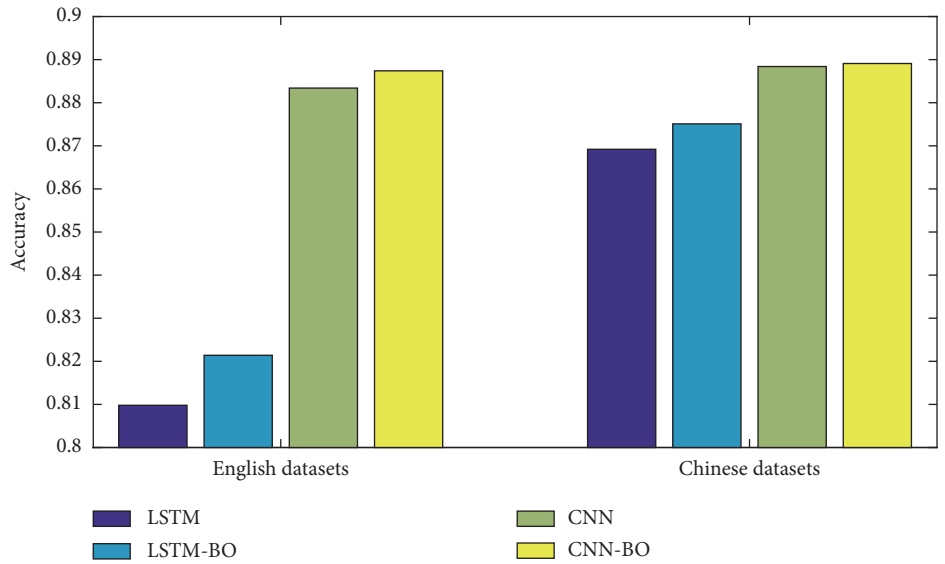


Figure 10: Analysis of the accuracy of the model in Chinese and English datasets.

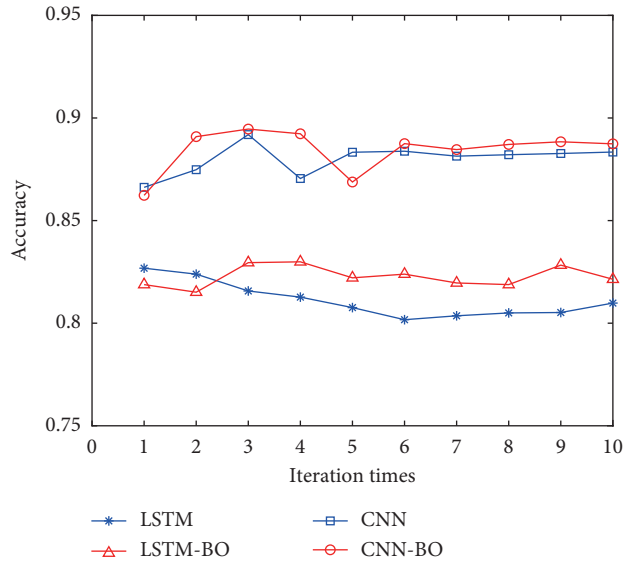


Figure 11: The accuracy of each model with the number of iterations under the English dataset.

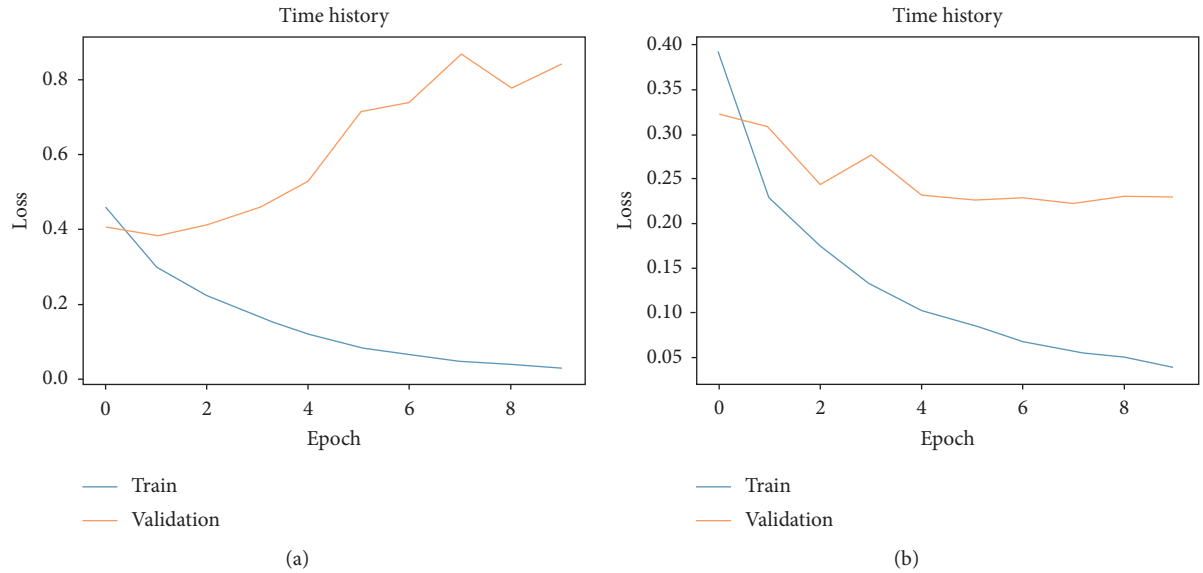


Figure 12: Continued.

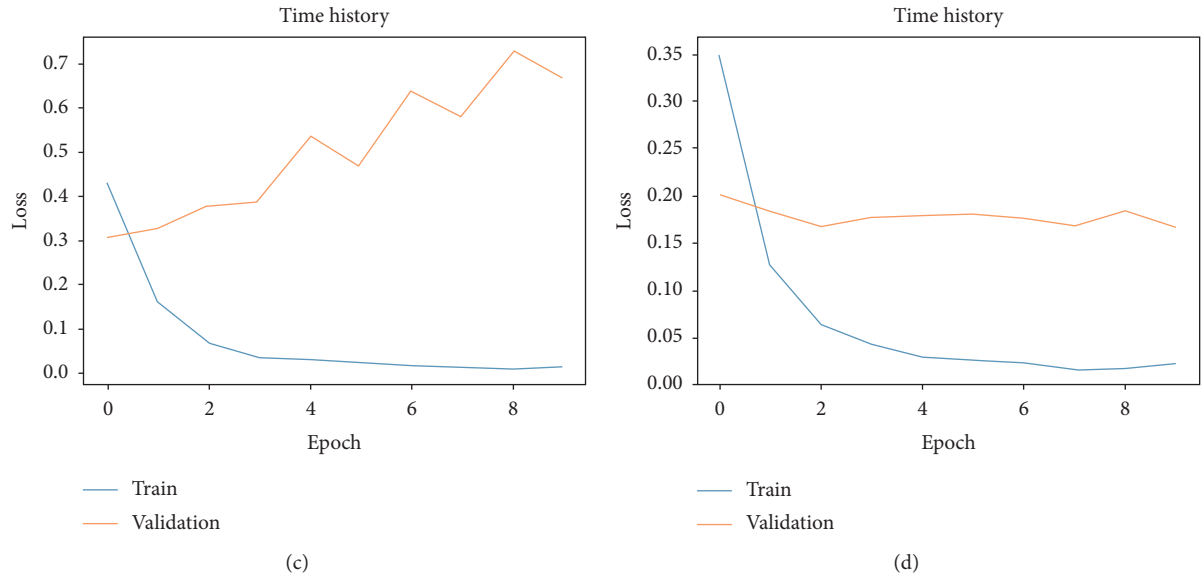


Figure 12: Loss rate change of LSTM model and LSTM-BO model in Chinese and English datasets. (a) Loss rate change of LSTM in English dataset. (b) Loss rate change of LSTM-BO in English dataset. (c) Loss rate change of LSTM in Chinese dataset. (d) Loss rate change of LSTM-BO in Chinese dataset.

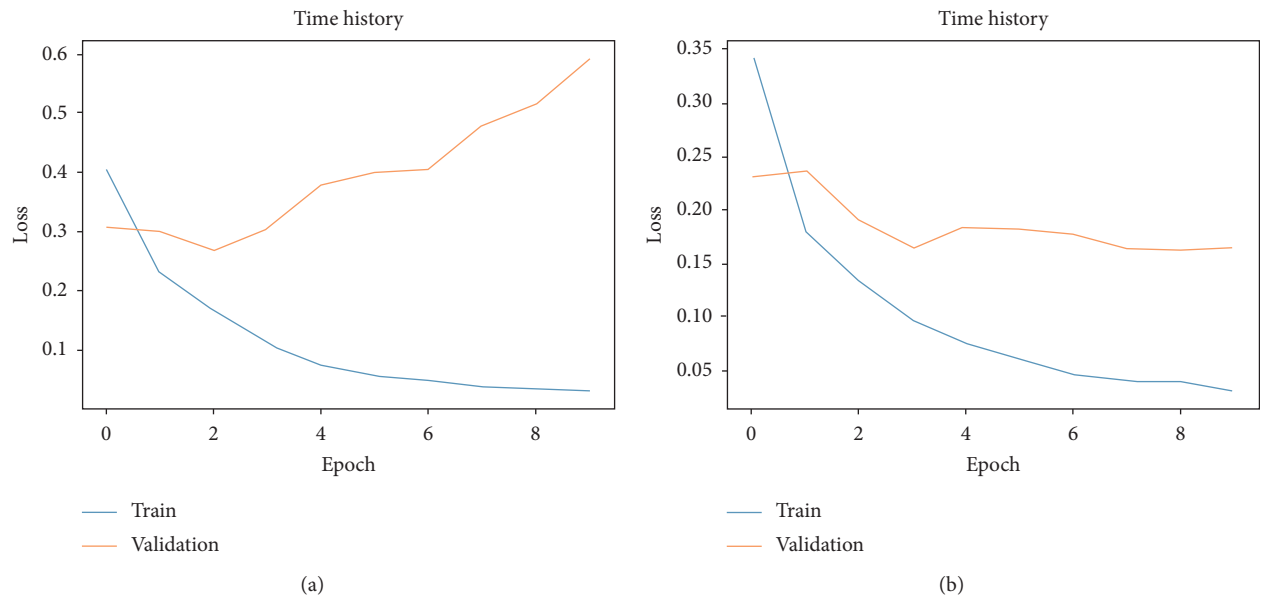


Figure 13: Continued.

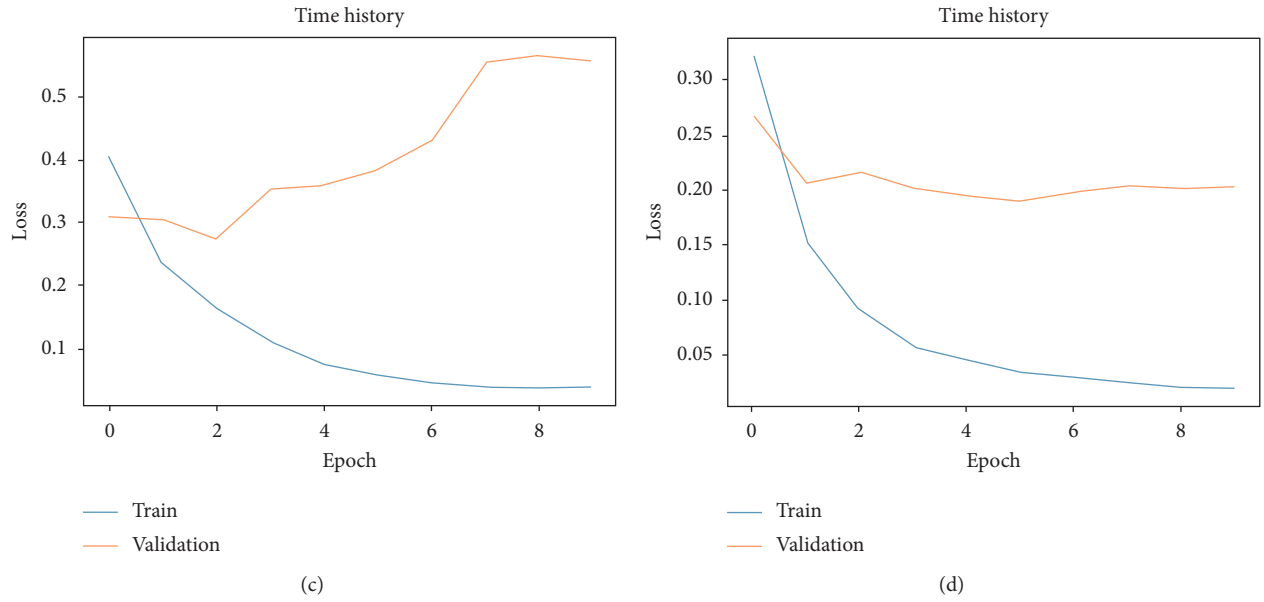


Figure 13: Changes in loss rate of CNN model and CNN-BO model in Chinese and English datasets. (a) Change in loss rate of CNN in English dataset. (b) Change in loss rate of CNN-BO in English dataset. (c) Change in loss rate of CNN in Chinese dataset. (d) Change in loss rate of CNN-BO in Chinese dataset.

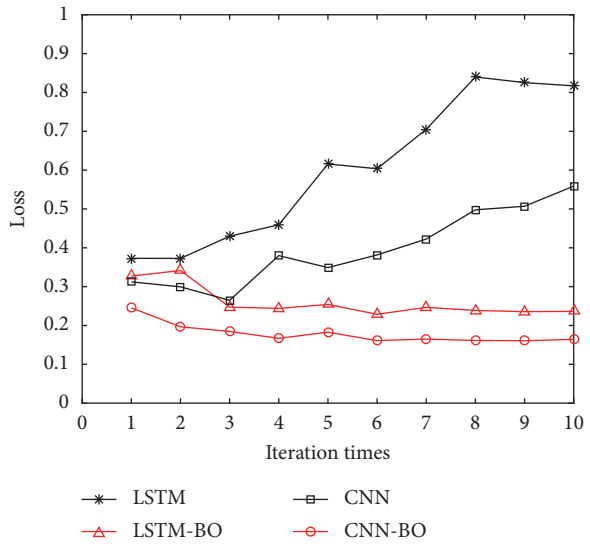


Figure 14: Changes in the loss rate of each model with the number of iterations under the English dataset.

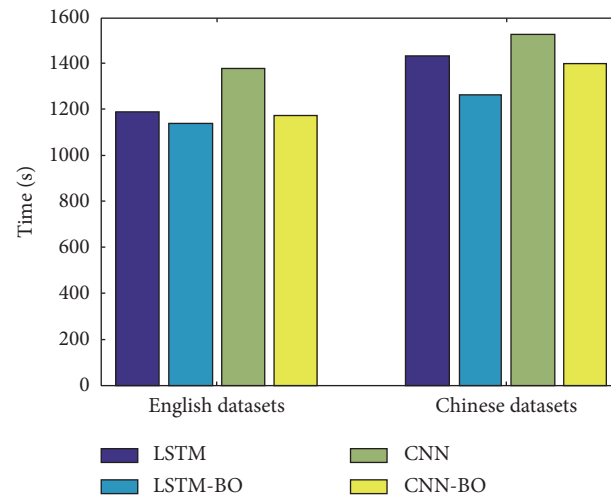


Figure 15: Time performance graph of Chinese and English datasets.

iterations in the Chinese dataset is 0.4616, which is 0.2332 lower than the RNN model. The analysis in this paper can prevent the overfitting of the model to a certain extent, and it can effectively extract the text features and reduce the loss.

(iii) Time performance analysis

Figure 19 shows the time consumption of the two models in the Chinese and English datasets. It can be seen from Figure 19 that the running time of W-RNN model is over 159 seconds than the RNN model in the English dataset and over 184 seconds than the RNN model in the Chinese dataset. The reason why the W-RNN is slower than RNN is that the W-RNN model has time consumption on calculating word weights and sorting according to the weight of words.

4.2.3. Discussion on W-RNN Model for Optimizing Loss Function. Based on the improved scheme of Section 2.1, loss function, the W-RNN model loss function is optimized, and the weight-recurrent neural network-binary-optimize (W-RNN-BO) model is constructed, and experiments are carried out on the Chinese and English datasets, respectively. The

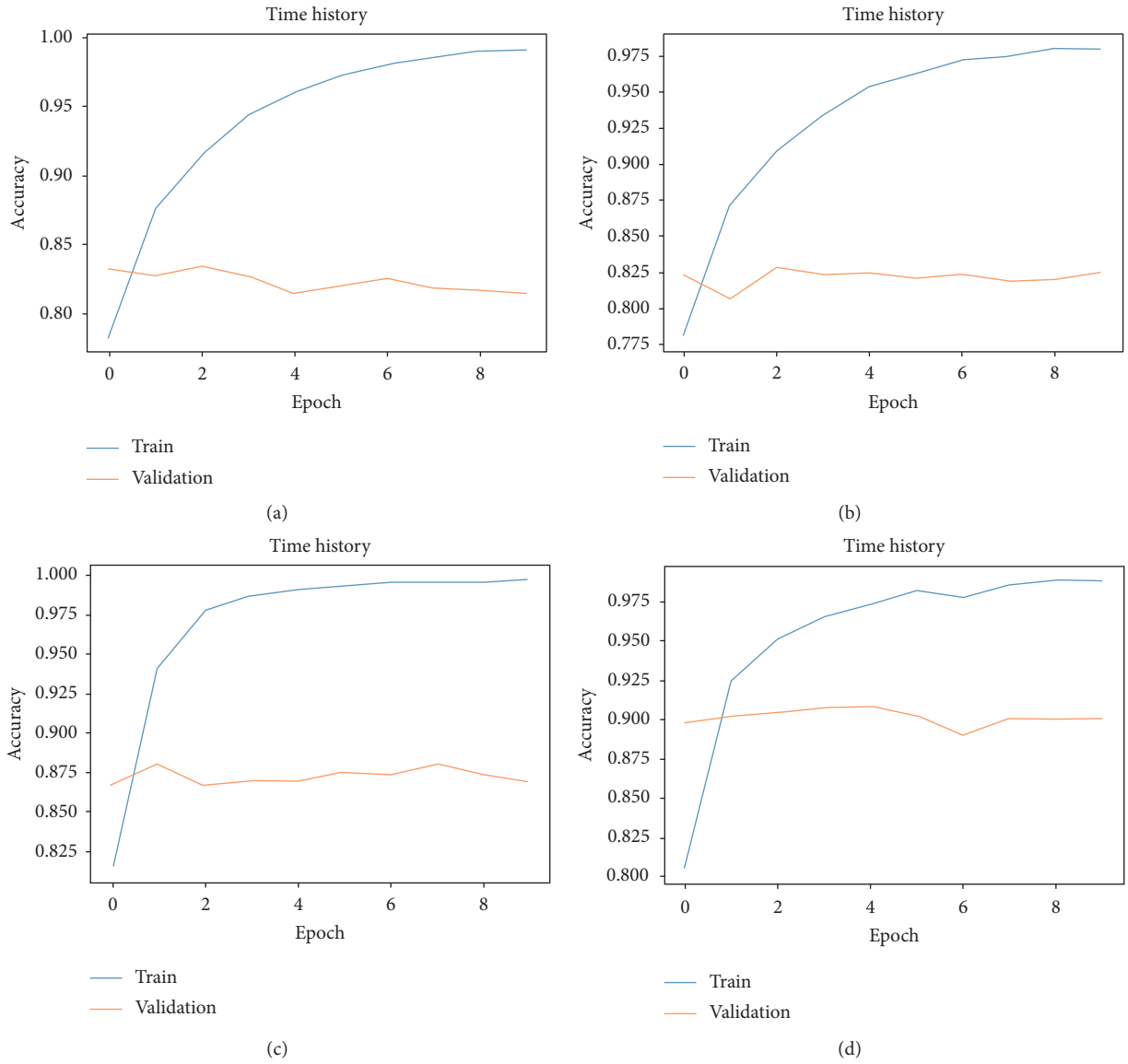


Figure 16: Accuracy of each model in Chinese and English datasets. (a) Changes in the accuracy of RNN in the English dataset. (b) Changes in the accuracy of W-RNN in the English dataset. (c) Changes in the accuracy of RNN in the Chinese dataset. (d) Changes in the accuracy of W-RNN in the Chinese dataset.

number of iterations is 10 times. The experimental results are shown in Table 13.

Table 13 shows the accuracy, loss rate, and time performance of the W-RNN-BO and W-RNN models in the Chinese and English datasets; Figure 20 shows a line graph showing the change in the accuracy of each model as the number of iterations increases in the Chinese and English datasets. The following results can be obtained from the above experimental results:

(i) In terms of accuracy, it can be seen from Figure 20 that the accuracy of the W-RNN-BO model is higher than that of the W-RNN model after the second iteration in the Chinese and English datasets. And after the 10th iteration, the W-RNN-BO model is 0.59% and 0.78% higher than the W-RNN model in the Chinese and English datasets, respectively.

(ii) In terms of loss rate, it can be seen from Table 13 that the loss convergence effect of the W-RNN-BO model is better. After the 10th iteration in the Chinese and English datasets, the loss rate is reduced to 0.1405 and 0.1430, respectively, which is lower than W-RNN of 0.3211 and 0.3070.

(iii) In terms of time performance, the W-RNN-BO model was reduced by 79 seconds in the English dataset compared to the W-RNN model and 105 seconds in the Chinese dataset compared to the W-RNN model.

This experiment also fully proves the effectiveness of the improved loss function described in Section 2.1, so the model possesses better generalization ability and improves

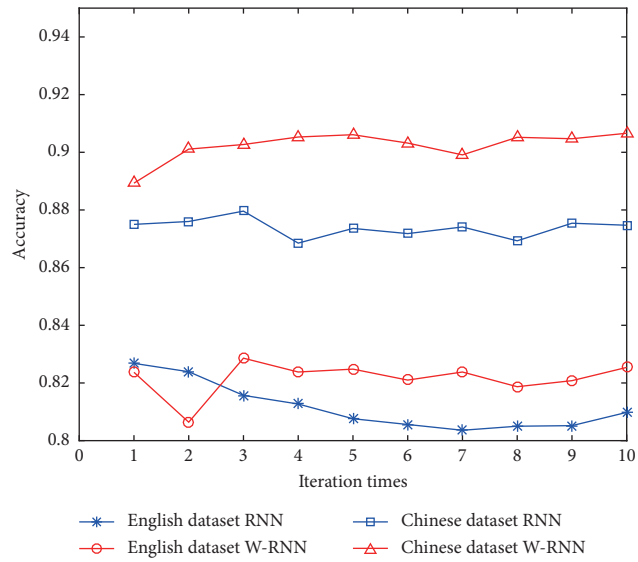


Figure 17: Accuracy of each model in Chinese and English datasets.

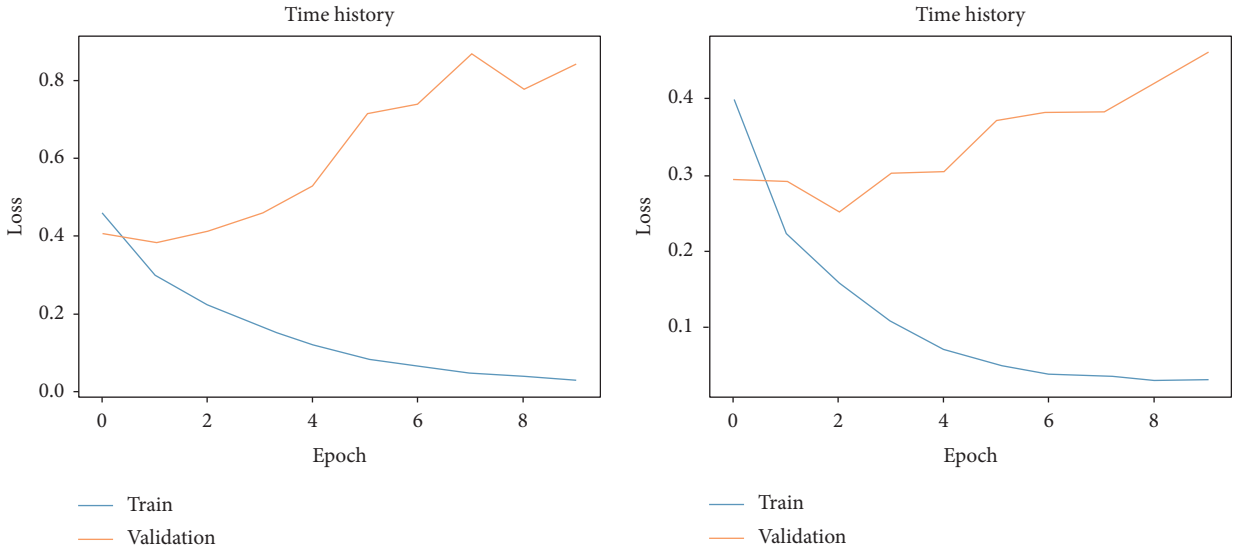


Figure 18: Continued.

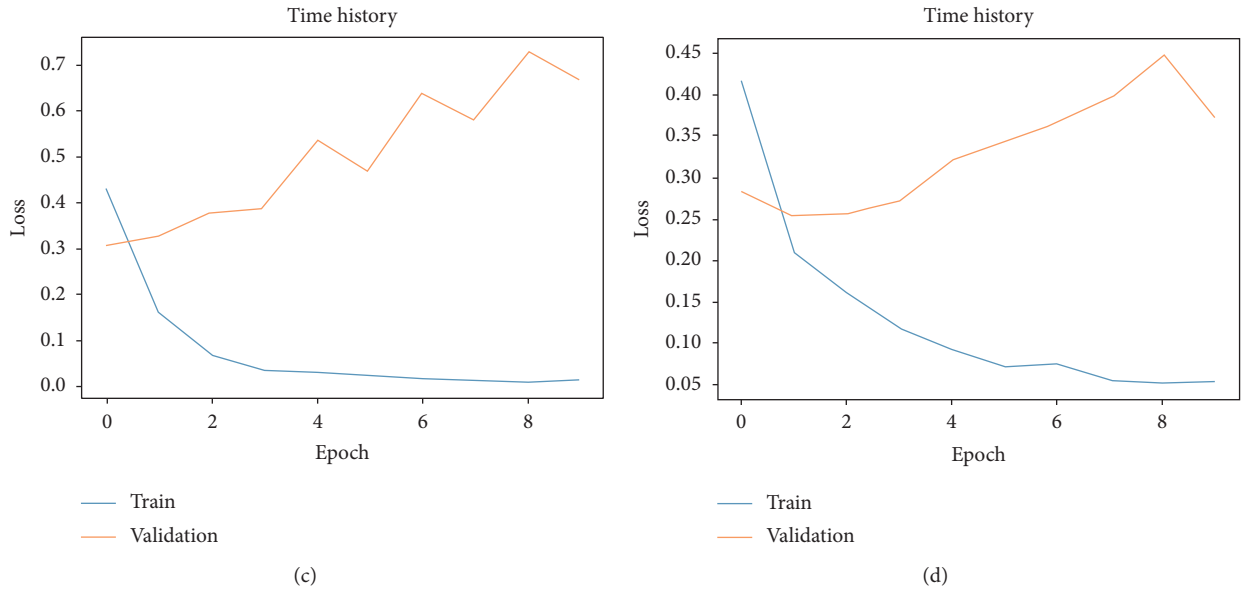


Figure 18: Loss of each model in Chinese and English datasets. (a) Change in loss rate of RNN in English dataset. (b) Change in loss rate of W-RNN in English dataset. (c) Change in loss rate of RNN in Chinese dataset. (d) Change in loss rate of W-RNN in Chinese dataset.

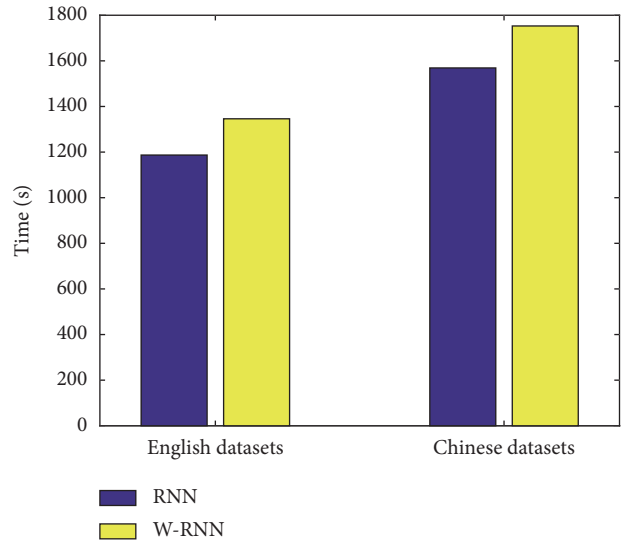


Figure 19: Time performance of each model in Chinese and English datasets.

Table 11: Analysis of output results of W-RNN model.

(1) The layout of this book is very poor, and four or five pages in front are blank. [(7, “poor”, 0.36663848), (17, “blank”, 0.16964875), (13, ”pages”, 0.16570723), (6, “very”, 0.11767213), (4, “book”, 0.07908833), (5, “is”, 0.0750145), (4, “this”, 0.06969571), (0, “the”, 0.04458785), (16,”front”, 0.023623824), (16, “are”, 0.006633401), (10, “four”, −0.0), (12, “five”, −0.015168488), (2, “of”, −0.022220552), (1, “layout”, −0.08092123)]
(2) This picture book is very good, my son likes it and has been reluctant to return it. [(1, “picture”, 0.30500817), (5, “good”, 0.2046071), (4, “very”, 0.17955697), (2, “book”, 0.17825627), (4, ’ 9′, 0.16365236), (14, “reluctant”, 0.15772212), (7, “my”, 0.11790274), (16, “return”, 0.10876195), (6, ’.’, 0.0819352), (3, “is”, 0.064508274), (0, “this”, 0.013084531), (12, “has”, 0.0077801645), (8, “son”, 0.0015705228), (15, “to”, −0.023336783), (11, “and”, −0.16156882), (13, “been”, −0.39944094)]

Table 12: Comparison of classification results on different network models.

Dataset	Model	Accuracy (%)	Loss (%)	Time (s)
English dataset	RNN	80.98	0.8170	1187
	W-RNN	**82.54**	**0.4500**	**1346**
Chinese dataset	RNN	87.47	0.6948	1569
	W-RNN	**90.66**	**0.4616**	**1753**

Table 13: Comparison of classification results on different network models.

Dataset	Model	Accuracy (%)	Loss (%)	Time (s)
English dataset	W-RNN	82.54	0.4500	1346
	W-RNN-BO	**83.32**	**0.1430**	**1267**
Chinese dataset	W-RNN	90.66	0.4616	1753
	W-RNN-BO	**91.25**	**0.1405**	**1648**

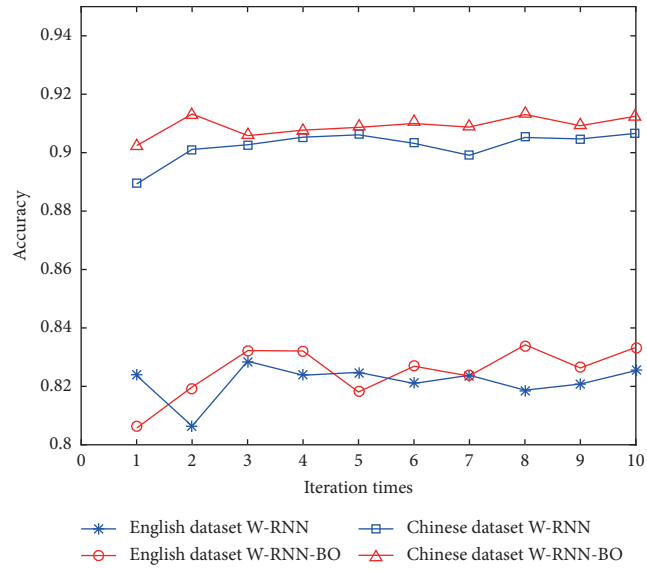


Figure 20: Variation rate of each model under different iterations.

the accuracy, loss rate, time, and other performances in the emotional classification task.

5. Conclusion

In order to solve the shortcomings of traditional deep neural networks in sentiment analysis tasks, three emotion classification models are proposed in this paper, based on deep neural networks. Firstly, based on the LSTM and CNN models, the traditional cross entropy loss function is improved. The LSTM-BO and CNN-BO models are designed so that the improved model can more effectively fit the prediction error samples and prevent the overfitting phenomenon. In addition, combined with the characteristics of the circulating neural network, by analysing the influence of the input words on the final classification, the importance of each word to the classification results is obtained, and the W-RNN model is constructed. The model gives higher weight to words with stronger emotional tendency and reduces the loss of emotional information. In order to verify the effectiveness of the three sentiment classification models, qualitative and quantitative sentiment analysis experiments of two kinds of datasets in Chinese and English were designed. The experimental results show that the three models proposed in this paper improve the accuracy of text sentiment classification to a certain extent and also perform better in loss rate and time performance.

In the next work, we will consider the characteristics of CNN with its extraction of text features and RNN with its capacity on series tasks, which can be combined with self-attention to construct a better model for text feature extraction and classification.

Data Availability

The data used to support the findings of this study not available in [28] can be obtained from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors are grateful to Chi-Hua Chen and Fangying Song for providing necessary advice. This research was partially supported by the National Key Research and Development Program of China (2018YFB1201500), National Natural Science Foundation of China (grant no. 61773313), Natural Science Basic Research Program of Shaanxi (program no. 2020JM-709), and Scientific Research Foundation of National University of Defense Technology (grant no. ZK18-03-43).

References

[1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification usingmachine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10*, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 79–86, July 2002.

[2] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 417–424, July 2002.

[3] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 142–150, July 2011.

[4] C. Du and L. Huang, "Sentiment analysis method based on piecewise convolutional neural network and generative adversarial network," *International Journal of Computers, Communications & Control*, vol. 14, 2019.

[5] Y. Zhang, Y. Tong, and Y. Jiang, "Study of sentiment classification for Chinese microblog based on recurrent neural network," *Chinese Journal of Electronics*, vol. 25, no. 4, pp. 601–607, 2016.

[6] A. McCallum, D. Freitag, and F. C. Pereira, "Maximum entropy markov models for information extraction and segmentation," *ICML*, vol. 17, pp. 591–598, 2000.

[7] Y. F. Pan, X. Hou, and C. L. Liu, "Text localization in natural scene images based on conditional random field," in *Proceedings of the 10th International Conference on Document Analysis and Recognition*, IEEE, Barcelona, Spain, pp. 6–10, July 2009.

[8] T. Joachims, *Learning to Classify Text Using Support Vector Machines*, Springer, Berlin, Germany, 2002.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[10] C.-H. Chen, F. Song, F.-J. Hwang, and L. Wu, "A probability density function generator based on neural networks," *Physica A: Statistical Mechanics and its Applications*, vol. 541, 2019.

[11] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[12] Y. Zhao, B. Qin, T. Liu, and D. Tang, "Social sentiment sensor: a visualization system for topic detection and topic sentiment analysis on microblog," *Multimedia Tools and Applications*, vol. 75, no. 15, pp. 8843–8860, 2016.

[13] Y. Kim, "Convolutional neural networks for sentence classification," 2014, http://arxiv.org/abs/1408.5882_2014.

[14] A. Kennedy and D. Inkpen, "Sentiment classification of movie reviews using contextual valence shifters," *Computational Intelligence*, vol. 22, pp. 110–112, 2011.

[15] Y. Kim, "Convolutional neural networks for sentence classification," 2014, http://arxiv.org/abs/1408.5882.

[16] T. Zagibalov and J. Carroll, "Automatic seed word selection for unsupervised sentiment classification of Chinese text," in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1073–1080, July 2008.

[17] D. Tritchler, *Loss Function, Encyclopedia of Biostatistics*, John Wiley & Sons, 2005.

[18] K. P. Kording and D. M. Wolpert, "The loss function of sensorimotor learning," *Proceedings of the National Academy of Sciences*, vol. 101, no. 26, pp. 9839–9842, 2004.

[19] X. Li, Y. Q. Lu, and D. Tao, "Robust subspace clustering by Cauchy loss function," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 7, pp. 2067–2078, 2019.

[20] A. Jati, N. Kumar, R. Chen et al., "Hierarchy-aware loss function on a tree structured label space for audio event detection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, May 2019.

[21] S. Cui, M. Chen, and C. Liu, "DsUnet: a new network structure for detection and segmentation of ultrasound breast lesions," *Journal of Medical Imaging and Health Informatics*, vol. 10, no. 3, pp. 661–666, 2020.

[22] C. Zhao, S. Wang, and D. Li, *Deep Transfer Learning for Social Media Cross-Domain Sentiment Classification, Chinese National Conference on Social Media Processing*, pp. 232–243, Springer, Berlin, Germany, 2017.

[23] A. Dey, M. Jenamani, and J. J. Thakkar, "Cross-D-vectorizers: a set of feature-spaces for cross-domain sentiment analysis from consumer review," *Multimedia Tools and Applications*, vol. 78, no. 16, pp. 23141–23159, 2019.

[24] P. Zola, P. Cortez, C. Ragno, and E. Brentari, "Social media cross-source and cross-domain sentiment classification," *International Journal of Information Technology and Decision Making*, vol. 18, no. 1, pp. 1469–1499, 2019.

[25] A. Dey, M. Jenamani, and J. J. Thakkar, "Sentiment weight of N-grams in dataset (SEND): a feature-set for cross-domain sentiment classification, 2017 ninth international conference on advances in pattern recognition (ICAPR)," pp. 1–6, IEEE, 2017.

[26] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: a unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, San Juan, PR, USA, June 2015.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] C. Potts, "On the negativity of negation," *Semantics and Linguistic Theory*, vol. 20, pp. 636–659, 2010.

Hindawi
Scientific Programming
Volume 2020, Article ID 1631869, 12 pages
https://doi.org/10.1155/2020/1631869

Research Article

A Novel Linguistic *Z*-Number QUALIFLEX Method and Its Application to Large Group Emergency Decision Making

Xue-Feng Ding, Li-Xia Zhu, Mei-Shun Lu, Qi Wang, and Yi-Qi Feng

School of Management, Shanghai University, Shanghai 200444, China

Correspondence should be addressed to Xue-Feng Ding; athena_tju@sina.com

Received 24 November 2019; Revised 17 March 2020; Accepted 10 June 2020; Published 26 June 2020

Academic Editor: Rahman Ali

After an unconventional emergency event occurs, a reasonable and effective emergency decision should be made within a short time period. In the emergency decision making process, decision makers' opinions are often uncertain and imprecise, and determining the optimal solution to respond to an emergency event is a complex group decision making problem. In this study, a novel large group emergency decision making method, called the linguistic *Z*-QUALIFLEX method, is developed by extending the QUALIFLEX method using linguistic *Z*-numbers. The evaluations of decision makers on the alternative solutions are first expressed as linguistic *Z*-numbers, and the group decision matrix is then constructed by aggregating the evaluations of all subgroups. The QUALIFLEX method is used to rank the alternative solutions for the unconventional emergency event. Besides, a real-life example of emergency decision making is presented, and a comparison with existing methods is performed to validate the effectiveness and practicability of the proposed method. The results show that the proposed linguistic *Z*-QUALIFLEX can accurately express the evaluations of the decision makers and obtain a more reasonable ranking result of solutions for emergency decision making.

1. Introduction

Emergency decision making is a human activity based on cognitive information, especially in the large group context. How to effectively express human cognitive information and perform cognitive computation in the large group emergency decision making is a highly challenging task and issue. Recently, a lot of studies have been carried out for cognition information representation and computation in solving various decision making problems, which include location evaluation [1, 2], enterprise resource planning system selection [3, 4], doctor selection [5], medical inquiry application evaluation [6], and others [7–9]. These research studies have proposed many effective methods and algorithms for using cognitive information in practical decision making processes. In this paper, we will pay attention to the large group emergency decision making problem based on human cognitive information.

Over the past several years, different unconventional emergency events occurred frequently all over the world. Examples include the Indian Ocean tsunami in 2004, Hurricane Katrina in the US in 2005, the 9.0 magnitude earthquake in Japan in 2011, and the Tianjin Port explosion in China in 2015, among others. These emergency events are difficult to predict and often impossible to control. They not only bring painful disasters and huge losses of life and properties but also cause serious social panic and potential secondary derivative hazards. When an emergency event happens, to reduce casualties and property losses and to eliminate various potential derivative hazards, decisions of reasonable and effective emergency relief solutions need to be made within a short time period [10]. However, because of their complexity, inadequate precursors, and devastating and potential secondary derivatives, it is difficult to deal with unconventional emergency events by conventional emergency management methods [11]. Therefore, the emergency decision making problem has gained increasing attention from both scholars and practitioners over recent years [12–21].

Previous works in solving emergency decision making problems involve only a small number of decision makers. However, such problems are usually large group emergency decision making (LGEDM) problems. Emergency management often involves many different organizations and departments, and the decision making team often includes a large number of experts from various professional backgrounds [12]. The large team, or large group, decision making problems have the following characteristics [21, 22]: (a) the team may have 20 or more members, and the members cooperate but may also have conflicts; (b) the various decision criteria may be incommensurate although related; and (c) the opinions of the team members may change over time. The large group decision making problem aims to support decision makers in making the best choice from all the alternatives or solutions while considering multiple criteria effectively and efficiently. Therefore, it is preferred to address emergency decision making problems in the environment of the large group.

Owing to the uncertainty and complexity of the emergency situations, it is hard for decision makers to express their judgments with accuracy, especially under time pressure [23]. Various theories, such as fuzzy sets, intuitionistic fuzzy sets, and hesitant fuzzy sets, have been proposed to deal with imprecise decision information. However, these theories are inefficient in expressing the reliability of decision makers' assessments [24]. This study uses linguistic Z-numbers [6] to represent inaccurate judgments of the decision makers. Compared with other methods such as fuzzy sets, intuitionistic fuzzy sets, and hesitant fuzzy sets, the linguistic Z-numbers can not only describe the cognition of the decision makers better but also consider the reliability of the cognitive information so as to represent the evaluations of decision makers more accurately.

The QUALIFLEX (Qualitative Flexible Multiple Criteria) method was developed by Paelinck [25] for group decision making using cardinal and ordinal information under uncertainty and fuzzy circumstance. Compared with other outranking methods such as SAW (simple additive weighting method), WP (weighted product method), LA (liner allocation method), TOPSIS, and ELECTRE, the QUALIFLEX method is a more flexible sorting method. In the QUALIFLEX method, a concordance/discordance index is first determined for each pair of alternatives through pairwise comparisons of the alternatives with respect to each criterion for all possible permutations of the alternatives. A weighted concordance/discordance index is then calculated for each pair of alternatives in each permutation. Finally, a comprehensive concordance/discordance index is calculated for each possible permutation of the alternatives. The permutation with the maximal value of the comprehensive concordance/discordance index is determined to be the best permutation, and the alternative ranked on the top of the best permutation is identified as the optimal alternative [26].

In order to describe the decision maker's assessment more accurately under the emergency circumstance and make an effective emergency decision as quickly as possible, this study develops an extended QUALIFLEX method using linguistic Z-numbers, called the linguistic Z-QUALIFLEX method, for solving LGEDM problems. Linguistic Z-numbers are first used to represent the decision makers' evaluations of the alternative solutions over various criteria, and the decision makers are divided into several subgroups according to the similarities of their evaluations. A group linguistic decision matrix is then constructed by aggregating the evaluations of all subgroups. The QUALIFLEX method is finally used to rank the alternative emergency solutions. Besides, an empirical example is provided to verify the applicability and the effectiveness of the proposed method. The proposed linguistic Z-QUALIFLEX method can accurately express the evaluations of the decision makers, characterize the similarities of their evaluations, and can also help decision makers get the best solution quickly and effectively.

The remainder of this paper is organized as follows. Previous related works are briefly reviewed in Section 2. Section 3 briefly introduces the basic concepts of Z-numbers and the QUALIFLEX method. Section 4 presents the linguistic Z-QUALIFLEX method for LGEDM problems. In Section 5, a numerical example is presented, and a comparative analysis is provided to illustrate the feasibility and validity of the proposed method. Conclusions and suggestions for future research are presented in Section 6.

2. Literature Review

Nowadays, a variety of methods have been proposed for EDM in the literature. For example, Ding et al. [27] proposed a new EDM approach by using picture fuzzy sets and an axiomatic design technique for determining the optimal rescue plan to reduce the damages of emergencies. Ding and Liu [28] presented an integrated approach based on prospect theory and the VIKOR (VIseKriterijumska Optimizacija I Kompromisno Resenje) method for EDM with 2-dimension uncertain linguistic information, and Ding and Liu [29] solved the EDM problem by the use a combined approach of Pythagorean fuzzy uncertain linguistic variables and zero-sum game theory. Li and Cao [30] provided a risk decision analysis method by extending the TODIM method with interval numbers to solve multiattribute risk decision making problems in emergency response. Peng et al. [31] introduced a deviation-based method with q-rung orthopair fuzzy number (q-ROFN) to deal with EDM problems. In [32], a consistency-based EDM method was designed by extending the incomplete probabilistic linguistic term set in an incomplete probabilistic linguistic preference relation (InPLPR). In [33], a method combining probabilistic linguistic term sets (PLTSs) and the D-S evidence theory is applied to the EDM problem.

In addition, there are a few research studies that focused on EDM on the situation of the large group in recent years. Song and Li [34] proposed a consensus process of the large group EDM by using multigranular probabilistic fuzzy linguistic preference relations (MGPFLPRs) to represent subgroup's preferences information. Xu et al. [35] presented a framework for the LGEDM problem in a linguistic environment by considering decision makers' risk appetites.

Xu et al. [36] found a LGEDM model for determining the severity of emergencies based on a decision paradigm obtained from the analysis of similar cases. Xu et al. [37] established a consensus model for the effective management of opinion differences and noncooperative behaviors in LGEDM environment. Cai et al. [16] suggested a multistage LGEDM model based on the preference information expressed as interval numbers and the similarity measures of decision makers. Aiming at the lower consensus and the urgency of LGEDM, Xu et al. [10] described a preference consensus model by taking into account the noncooperative behaviors and minority opinions of decision makers, and Xu et al. [38] proposed a dynamical consensus method based on exit-delegation mechanism considering consistency and consensus measures simultaneously.

The above literature review shows that various theories, such as interval numbers, 2-dimension uncertain linguistic variables, and probabilistic linguistic term sets, have been employed to deal with imprecise cognitive information in the EDM process. However, these theories are inefficient in expressing the reliability of decision makers' cognitive assessments. Moreover, no or little attention has been paid to the large group EDM problems under the context of linguistic Z-numbers. In addition, various MCDM methods have been proposed to deal with EDM problems. To the best of our knowledge, no previous studies have investigated EDM problems with the QUALIFLEX method yet. Therefore, in this paper, we fill the above gaps by developing a novel, integrated cognitive approach based on linguistic Z-numbers and an extended QUALIFLEX method to solve the EDM problem within the large group environment. The developed method in this paper can not only express the subjective cognitive evaluation information of decision makers more precisely but also support the rescue team in finding the optimal response to an emergency event effectively.

3. Preliminaries

3.1. Linguistic Z-Numbers. Combining Z-numbers [39] and linguistic variables [40], Wang et al. [6] introduced the concept of linguistic Z-numbers to address the reliability of information as a significant dimension in a decision making process.

Definition 1 (see [41]). An uncertain variable is a measurable function from an uncertainty space to the set of real numbers.

Definition 2 (see [6]). Suppose that X is a finite universe of discourses. $S_1 = (s_0, s_1, \ldots, s_{g-1})$ and $S_2 = (s_0', s_1', \ldots, s_{g'-1}')$ are two finite and totally ordered discrete linguistic term sets, where g and g' are the positive odd integers. Let $A_{\phi(x)} \in S_1$ and $B_{\varphi(x)} \in S_2$. A linguistic Z-number set Z in X is expressed as

$$Z = \left\{ \left(x, A_{\phi(x)}, B_{\varphi(x)} \right) \middle| x \in X \right\}, \tag{1}$$

where $A_{\phi(x)}$ represents the fuzzy restriction on the domain of the uncertain variables, and $B_{\varphi(x)}$ is a measure of reliability of $A_{\phi(x)}$.

Normally, S_1 and S_2 represent different preference information and, hence, are not always consistent. For simplicity, $z = (A_{\phi(x)}, B_{\varphi(x)})$ is called a linguistic Z-number, where $A_{\phi(x)} \in S_1$ and $B_{\varphi(x)} \in S_2$ are the two linguistic terms.

Definition 3 (see [6]). Let $z_u = (A_{\phi(u)}, B_{\varphi(u)})$ and $z_v = (A_{\phi(v)}, B_{\varphi(v)})$ be two linguistic Z-numbers and f^* and g^* be the possible functions of $F_1(\theta_u)$, $F_2(\theta_u)$, $F_3(\theta_u)$, and $F_4(\theta_u)$ where $F_1(\theta_u)$, $F_2(\theta_u)$, $F_3(\theta_u)$, and $F_4(\theta_u)$ are strictly monotonically increasing functions. Then, the operational laws of Z-numbers are defined as follows:

(1) $\mathrm{neg}(z_u) = (f^{*-1}(f^*(A_{l-1}) - f^*(A_{\phi(u)})), g^{*-1}(g^*(B_{k-1}) - g^*(B_{\varphi(u)})))$

(2) $z_u \oplus z_v = (f^{*-1}(f^*(A_{\phi(u)}) + f^*(A_{\phi(v)}))g^{*-1}((f^*(A_{\phi(u)}) \times g^*(B_{\varphi(u)}) + f^*(A_{\phi(v)}) \times g^*(B_{\varphi(v)}))/(f^*(A_{\phi(u)}) + f^*(A_{\phi(v)}))))$

(3) $z_u \otimes z_v = (f^{*-1}(f^*(A_{\phi(u)}) f^*(A_{\phi(v)})), g^{*-1}(g^*(B_{\varphi(u)}) g^*(B_{\varphi(v)})))$

(4) $\lambda z_u = (f^{*-1}(\lambda f^*(A_{\phi(u)})), B_{\varphi(u)})$, for $\lambda \geq 0$

(5) $(z_u)^\lambda = (f^{*-1}(f^*(A_{\phi(u)})^\lambda), g^{*-1}(g^*(B_{\varphi(u)})^\lambda))$, for $\lambda \geq 0$

Definition 4 (see [6]). Suppose that $S = \{s_0, s_1, \ldots, s_{g-1}\}$ is a linguistic term set and $z_u = (A_{\phi(u)}, B_{\varphi(u)})$ is a linguistic Z-number; then, the score function of z_u is defined as

$$S(z_u) = f^*\left(A_{\phi(u)}\right) \times g^*\left(B_{\varphi(u)}\right), \tag{2}$$

and the accuracy function of z_u is defined as

$$A(z_u) = f^*\left(A_{\phi(u)}\right) \times \left(1 - g^*\left(B_{\varphi(u)}\right)\right). \tag{3}$$

Definition 5 (see [6]). Let $z_u = (A_{\phi(u)}, B_{\varphi(u)})$ and $z_v = (A_{\phi(v)}, B_{\varphi(v)})$ be two linguistic Z-numbers. Then, the comparison laws of z_u and z_v are defined as follows:

(1) When $A_{\phi(u)} > A_{\phi(v)}$ and $B_{\varphi(u)} > B_{\varphi(v)}$, then z_u is strictly greater than z_v, expressed as $z_u > z_v$

(2) When $S(z_u) \geq S(z_v)$ and $A(z_u) > A(z_v)$, then z_u is greater than z_v, expressed as $z_u \succ z_v$

(3) When $S(z_u) = S(z_v)$ and $A(z_u) = A(z_v)$, then z_u equals z_v, expressed as $z_u \sim z_v$

(4) When $S(z_u) \leq S(z_v)$ and $A(z_u) < A(z_v)$, then z_u is less than z_v, expressed as $z_u \prec z_v$

Definition 6 (see [6]). Let $z_u = (A_{\phi(u)}, B_{\varphi(u)})$ and $z_v = (A_{\phi(v)}, B_{\varphi(v)})$ be two linguistic Z-numbers and f^* and g^* be two linguistic scale functions. Then, the distance between z_u and z_v is defined as

$$d(z_u, z_v) = \frac{1}{2}\left(\left|f^*\left(A_{\phi(u)}\right) \times g^*\left(B_{\varphi(u)}\right) - f^*\left(A_{\phi(v)}\right) \times g^*\left(B_{\varphi(v)}\right)\right| + \max\left\{\left|f^*\left(A_{\phi(u)}\right) - f^*\left(A_{\phi(v)}\right)\right|, \left|g^*\left(B_{\varphi(u)}\right) - g^*\left(B_{\varphi(v)}\right)\right|\right\}\right). \tag{4}$$

To aggregate the evaluations expressed by linguistic Z-numbers, the linguistic Z-number weighted averaging (LZNWA) operator is proposed.

Definition 7. Suppose $Z = \{z_j \mid j = 1, 2, \ldots, n\}$ is a collection of linguistic Z-numbers $z_j = (A_{\phi(j)}, B_{\varphi(j)})$, and LZNWA: $Z^n \longrightarrow Z$, then the LZNWA operator is given by

$$\mathrm{LZNWA}(z_1, z_2, \ldots, z_n) = \overset{n}{\underset{j=1}{\oplus}}\left(w_j z_j\right), \tag{5}$$

where $\oplus$ is the same operator defined in Definition 3, w_j is the weight of z_j, satisfying $0 \le w_j \le 1$, for $j = 1, 2, \ldots, n$, and $\sum_{j=1}^{n} w_j = 1$. If $w = (1/n, 1/n, \ldots, 1/n)$, the LZNWA operator becomes the linguistic Z-number arithmetic mean (LZAM) operator.

3.2. QUALIFLEX Method. The QUALIFLEX method, proposed by Paelinck [25], is a useful outranking technique for solving multiple attribute decision making problems with exact values for the evaluations given by the decision makers [42, 43]. The QUALIFLEX method treats cardinal and ordinal information simultaneously in the process of decision making. In the QUALIFLEX method, pairwise comparisons of the alternatives with respect to each criterion under all possible alternative permutations are made, and the optimal alternative(s) can be determined by maximizing the value of the comprehensive concordance/discordance index [44] among all possible alternative permutations. Let $E = \{E_1, E_2, \ldots, E_m\}$ be a set of alternatives, $C = \{C_1, C_2, \ldots, C_n\}$ be a set of criteria, and $W = \{w_1, w_2, \ldots, w_n\}$ be the weights of the criteria satisfying $0 \le w_j \le 1$, for $j = 1, 2, \ldots, n$, and $\sum_{j=1}^{n} w_j = 1$. There are $m!$ permutations of the rankings of the alternatives. The decision process of the QUALIFLEX method is described step by step as follows [25, 45]:

Step 1: list all $m!$ permutations of the rankings of the alternatives. Suppose P_η, as shown in equation (6), is the η th permutation:

$$P_\eta = \left(\ldots, E_\alpha, \ldots, E_\beta, \ldots\right), \quad \text{for } \eta = 1, 2, \ldots, m!, \tag{6}$$

where $E_\alpha, E_\beta \in E$, and the ranking of E_α is higher or equal to that of E_β.

Step 2: determine the concordance/discordance index $I_j^\eta(E_\alpha, E_\beta)$ by using equation (7) for each pair of alternatives (E_α, E_β), and measure the similarity between E_α and E_β, in each permutation P_η with respect to each criterion j:

$$I_j^\eta\left(E_\alpha, E_\beta\right) = \begin{cases} E_\alpha \succ E_\beta \Leftrightarrow \text{concordance exists}, \\ E_\alpha \sim E_\beta \Leftrightarrow \text{ex aequo exists}, \\ E_\alpha \prec E_\beta \Leftrightarrow \text{discordance exists}. \end{cases} \tag{7}$$

Step 3: calculate the weighted concordance/discordance index $I^\eta(E_\alpha, E_\beta)$ by using equation (8) for each pair of alternatives E_α and E_β in each permutation P_η:

$$I^\eta\left(E_\alpha, E_\beta\right) = \sum_{j=1}^{n} I_j^\eta\left(E_\alpha, E_\beta\right) w_j. \tag{8}$$

Step 4: determine the comprehensive concordance/discordance index I^η by using equation (9) for each permutation P_η and determine the optimal alternative:

$$I^\eta = \sum_{\alpha,\beta=1,2,\ldots,m} I^\eta\left(E_\alpha, E_\beta\right). \tag{9}$$

The best ranking of the alternatives is the one with the maximum value of the comprehensive concordance/discordance index $I^* = \max_{\eta=1,2,\ldots,m!}\{I^\eta\}$. The alternative ranked on the top of the corresponding ranking P^* is the optimal alternative.

4. The Proposed LGEDM Approach

In this section, the linguistic Z-QUALIFLEX method, an extended QUALIFLEX approach with linguistic Z-numbers, is proposed to solve emergency decision making problems when a large group of decision makers is involved. The evaluations of the alternatives with respect to the criteria, represented by Z-numbers, are given by the large group of decision makers. The major steps of the proposed approach are shown in Figure 1. The decision makers are first divided into several subgroups according to the similarities of their evaluations. The evaluations represented by linguistic Z-numbers given by the decision makers in each subgroup are then aggregated into a group linguistic Z-number decision matrix using the LZAM operator. The best ranking of the alternatives is finally identified by using the QUALIFLEX method.

In the QUALIFLEX method, let $E = \{E_1, E_2, \ldots, E_m\}$ be a set of emergency solutions, i.e., alternatives, $C = \{C_1, C_2, \ldots, C_n\}$ be a set of criteria, $W = \{w_1, w_2, \ldots, w_n\}$ be the weights of the criteria satisfying $0 \le w_j \le 1$, for $j = 1, 2, \ldots, n$, and $\sum_{j=1}^{n} w_j = 1$, and $\mathrm{DM} = \{D_1, D_2, \ldots, D_l\}$ with $l \ge 20$ be a set of decision makers in an LGEDM problem. Each decision maker D_k gives his/her judgment of E_i with respect to C_j using linguistic Z-numbers, represented by z_{ij}^k. The linguistic Z-number evaluation matrices $Z_k = [z_{ij}^k]_{m\times n}$, with $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$ for $k = 1, 2, \ldots, l$, are then obtained. Each element z_{ij}^k of Z_k is given by $z_{ij}^k = (A_{\phi(ij)}^k, B_{\varphi(ij)}^k)$, where $A_{\phi(ij)}^k$ is the linguistic evaluation of alternative E_i with respect to criterion C_j given by the decision maker

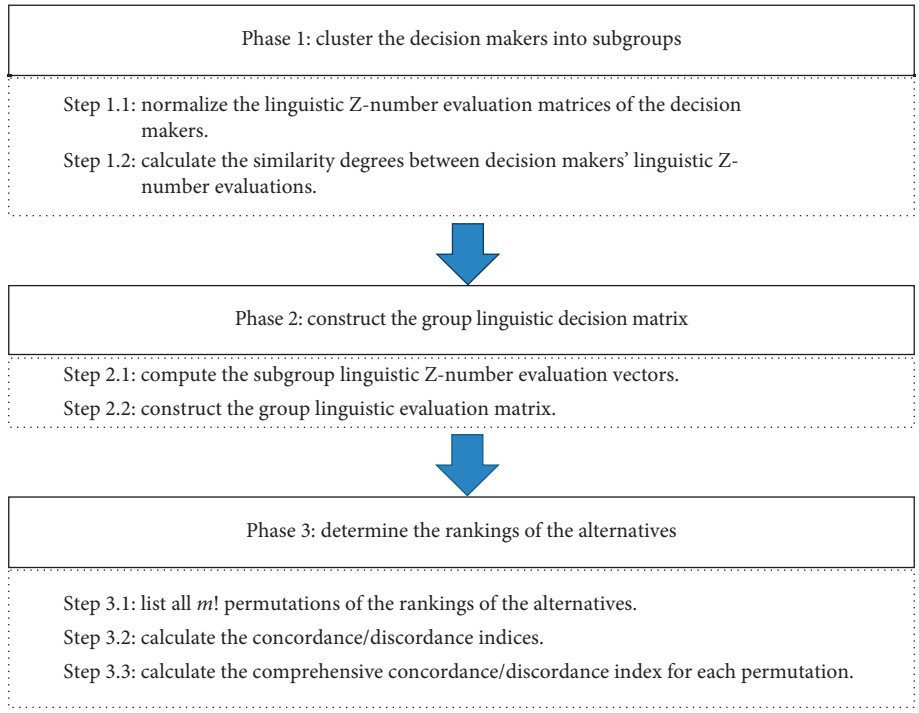


Figure 1: Flowchart of the proposed linguistic Z-QUALIFLEX method.

D_k using the linguistic term set $S_1 = (s_0, s_1, \ldots, s_{g-1})$, and $B^k_{\varphi(ij)}$ is a measure of reliability of $A^k_{\phi(ij)}$ using the linguistic term set $S_2 = (s'_0, s'_1, \ldots, s_{g'-1})$. The evaluations of decision makers are assumed to be independent of each other. The linguistic Z-QUALIFLEX method used to solve LGEDM problems is presented step by step in three phases in the following.

Phase 1: cluster the decision makers into subgroups

Step 1.1: normalize the linguistic Z-number evaluation matrix $Z_k = [z^k_{ij}]_{m\times n}$ by using

$$\overline{z}^k_{ij} = \begin{cases} z^k_{ij}, & \text{for benefit criteria,} \\ \text{neg}\left(z^k_{ij}\right), & \text{for cost criteria.} \end{cases} \tag{10}$$

The normalized linguistic Z-number evaluation matrix is represented by $\overline{Z}_k = [\overline{z}^k_{ij}]_{m\times n}$.

Step 1.2: calculate the similarity degrees between decision makers' linguistic Z-number evaluations. Let $V^{(i)}_k = (\overline{z}^k_{i1}, \overline{z}^k_{i2}, \ldots, \overline{z}^k_{in})$ denote the normalized linguistic evaluation vector for alternative E_i, for $i = 1, 2, \ldots, m$, by decision maker k, for $k = 1, 2, \ldots, l$. The similarity degree $S(V^{(i)}_\mu, V^{(i)}_\nu)$ between $V^{(i)}_\mu$ and $V^{(i)}_\nu$ is given by

$$S\left(V^{(i)}_\mu, V^{(i)}_\nu\right) = \frac{1}{1 + \sum_{j=1}^{n} d\left(\overline{z}^\mu_{ij}, \overline{z}^\nu_{ij}\right)}. \tag{11}$$

Step 1.3: divide decision makers into subgroups for each alternative. Suppose $\lambda^{(i)}$ is the clustering threshold of alternative E_i obtained by using

$$\lambda^{(i)} = S\text{Min} + \frac{2}{3}(S\text{Max} - S\text{Min}), \tag{12}$$

where

$$S\text{Max} = \max\left\{S\left(V^{(i)}_\mu, V^{(i)}_\nu\right) \middle| V^{(i)}_\mu, V^{(i)}_\nu \in \Omega^{(i)}, V^{(i)}_\mu \neq V^{(i)}_\nu\right\},$$
$$S\text{Min} = \min\left\{S\left(V^{(i)}_\mu, V^{(i)}_\nu\right) \middle| V^{(i)}_\mu, V^{(i)}_\nu \in \Omega^{(i)}, V^{(i)}_\mu \neq V^{(i)}_\nu\right\}, \tag{13}$$

and $\Omega^{(i)}$ are the evaluation vectors of all decision makers for alternative E_i.

For alternative E_i, if $S(V^{(i)}_\mu, V^{(i)}_\nu) \geq \lambda^{(i)}$, then $V^{(i)}_\mu$ and $V^{(i)}_\nu$ are put into the same subgroup. Let $H^{(i)}$ denote the number of subgroups after clustering for alternative E_i, $\Lambda^{(i)}_h$ be the decision makers in subgroup h, $\Omega^{(i)}_h$ be the corresponding assessment vectors of $\Lambda^{(i)}_h$, and $k^{(i)}_h$ be the number of decision makers in $\Lambda^{(i)}_h$. The requirement $k^{(i)}_h \geq 2$ is needed. If there is a $\Lambda^{(i)}_\xi$, for any $\xi = 1, 2, \ldots, H^{(i)}$, such that $k^{(i)}_\xi = 1$, $\Lambda^{(i)}_\xi$ needs to be modified. The method proposed by Cai et al. [16] is used for modification.

Phase 2: construct the group linguistic decision matrix

Step 2.1: aggregate the linguistic Z-number evaluation vectors of the decision makers in each subgroup into a subgroup linguistic Z-number evaluation vector by the LZAM operator. For each alternative E_i, the linguistic Z-number evaluation vectors $\Omega^{(i)}_{h_i}$, for $1 \leq h \leq H^{(i)}$, of $k^{(i)}_h$ decision makers in subgroup $\Lambda^{(i)}_h$ are aggregated into the subgroup linguistic evaluation vector $\widetilde{V}^{(i)}_h$ by using the LZAM operator:

$$\widetilde{V}_h^{(i)} = \mathrm{LZAM}\left(\overline{V}_1^{(i)}, \overline{V}_2^{(i)}, \ldots, \overline{V}_{k_h^{(i)}}^{(i)}\right) = \overset{k_h^{(i)}}{\underset{r=1}{\oplus}}\left(\frac{1}{k_h^{(i)}}\overline{z}_{ijr}^{(i)}\right), \tag{14}$$

where $(1/k_h^{(i)})$ is the weight of $\overline{z}_{ijr}^{(i)}$.

Step 2.2: construct the group linguistic evaluation matrix $\widehat{Z} = [\widehat{z}_{ij}]_{m\times n}$. Using the subgroup linguistic evaluation vectors $\widetilde{V}_h^{(i)}$, for $i = 1, 2, \ldots, m$, the group linguistic evaluation vector can be obtained by using the LZAM operator:

$$\widetilde{V}^{(i)} = \mathrm{LZAM}\left(\widetilde{V}_1^{(i)}, \widetilde{V}_2^{(i)}, \ldots, \widetilde{V}_{H^{(i)}}^{(i)}\right) = \overset{k_h^{(i)}}{\underset{r=1}{\oplus}}\left(w_h^{(i)}\overline{z}_{ijr}^{(i)}\right), \tag{15}$$

where $w_h^{(i)} = (k_h^{(i)}/l)$. The m group linguistic evaluation vectors $\widetilde{V}^{(i)}$, for $i = 1, 2, \ldots, m$, form the group linguistic evaluation matrix $\widehat{Z} = [\widehat{z}_{ij}]_{m\times n}$.

Phase 3: rank the alternatives by using the QUALIFLEX method

Step 3.1: list all $m!$ permutations of the rankings of the alternatives:

$$P_\eta = \left(\ldots, E_\alpha, \ldots, E_\beta, \ldots\right), \quad \text{for } \eta = 1, 2, \ldots, m!, \tag{16}$$

where P_η is the ηth permutation, $E_\alpha, E_\beta \in E$, and E_α is ranked higher or equal to E_β in the permutation.

Step 3.2: calculate the concordance/discordance index $I_j^\eta(E_\alpha, E_\beta)$ of each pair of alternatives (E_α, E_β) in each permutation P_η by considering every criterion using

$$I_j^\eta\left(E_\alpha, E_\beta\right) = \begin{cases} d\left(z_{\alpha j}, z_0\right) - d\left(z_{\beta j}, z_0\right) > 0 \Leftrightarrow \text{concordance exists}, \\ d\left(z_{\alpha j}, z_0\right) - d\left(z_{\beta j}, z_0\right) = 0 \Leftrightarrow \text{ex aequo exists}, \\ d\left(z_{\alpha j}, z_0\right) - d\left(z_{\beta j}, z_0\right) < 0 \Leftrightarrow \text{discordance exists}. \end{cases} \tag{17}$$

Step 3.3: calculate the weighted concordance/discordance index $I^\eta(E_\alpha, E_\beta)$ of each permutation P_η by using

$$I^\eta\left(E_\alpha, E_\beta\right) = \sum_{j=1}^{n} I_j^\eta\left(E_\alpha, E_\beta\right) w_j. \tag{18}$$

Step 3.4: determine the comprehensive concordance/discordance index I^η by using

$$I^\eta = \sum_{\alpha,\beta=1,2,\ldots,m} \sum_{j=1}^{n} I_j^\eta\left(E_\alpha, E_\beta\right) w_j. \tag{19}$$

The best ranking of the alternatives is P^* with the maximum value of the comprehensive concordance/discordance index I^*, i.e., $I^* = \max_{\eta=1,2,\ldots,m!}\{I^\eta\}$. The alternative ranked on the top of P^* is the optimal solution for the LGEDM problem.

5. A Numerical Example

In China, from June to July, there are often persistent rainstorms in the middle and lower reaches of the Yangtze River, which are of long duration, wide area, and heavy rainfall. July to August is the main rainstorm season in northern provinces, and the rainstorm intensity is very high. These torrential rains often lead to floods, which may cause river levee breach incidents, resulting in great harm to people and public property.

In order to reduce the huge damage to people and public property caused by these accidents and provide scientific support for the emergency management of such emergencies, in this section, an LGEDM problem of a river levee breach incident is used as the illustrative example to demonstrate the effectiveness and practicability of the proposed linguistic Z-QUALIFLEX method.

5.1. Decision Process and Results. In the morning of June 21, 2017, a severe thunderstorm watch with red color warning of possible heavy rainfall in the next three days was issued by the meteorological department in area C in Hubei province of China. According to weather report, the probabilities of moderate, heavy, and violent rainfalls are 0.30, 0.50, and 0.20, respectively. A riverbank breach will cause a wide range flooding disaster and a large number of casualties, seriously affecting the lives of people in the surrounding area. 20 experts from the emergency management department, the meteorological department, the water conservancy department, and the transportation department were invited to choose the best emergency solution within a short period of time. There are three emergency solutions $E = \{E_1, E_2, E_3\}$ for the decision makers to choose from. E_1: strengthen the riverbank, keep close monitoring, and expand the emergency force of inspection and rescue. E_2: strengthen the riverbank, keep close monitoring, send the villager rescue team to the site, and transfer the villagers in low-lying areas to safe areas. E_3: strengthen the riverbank, keep close monitoring, and send the villager rescue team and medical team to the site, allocate a large quantity of relief supplies, and transfer the nearby villagers to safe areas. Four criteria $C = \{C_1, C_2, C_3, C_4\}$ are considered for each emergency solution: C_1—number of victims, C_2—public satisfaction, C_3—property loss, and C_4—rescue cost. The weights of these criteria are supposed as W =(0.30, 0.25, 0.30, and 0.15). Suppose that all decision makers DM = $\{D_1, D_2, \ldots, D_{20}\}$ have the same weights and are required to give their

linguistic Z-number evaluations for the solutions over each criterion. Note that C_2 is a benefit, i.e., maximization, criterion, while C_1, C_3, and C_4 are cost, i.e., minimization, criteria.

The evaluations given by the 20 decision makers on three solutions by considering the four criteria are represented by linguistic Z-numbers. The linguistic term set S_1 for $A^k_{\phi(ij)}$ used by the decision makers is

$$\begin{aligned} S_1 = \{&s_0 = \text{very low influence},\ s_1 = \text{low influence},\ s_2 = \text{slightly low influence},\\ &s_3 = \text{moderate influence},\ s_4 = \text{slightly high influence},\ s_5 = \text{high influence},\\ &s_6 = \text{very high influence}\}, \end{aligned} \tag{20}$$

and the linguistic term set S_2 for $B^k_{\varphi(ij)}$ is

$$\begin{aligned} S_2 = \Big\{&s_{0_l} = \text{very unreliable},\ s_{1_l} = \text{unreliable},\ s_{2_l} = \text{somewhat reliable},\\ &s_{3_l} = \text{reliable},\ s_{4_l} = \text{very reliable}\Big\}. \end{aligned} \tag{21}$$

The decision process and the results of the proposed linguistic Z-QUALIFLEX method of selecting the best solution for this example LGEDM problem are summarized as follows:

Phase 1: divide the decision makers into subgroups

The linguistic Z-number evaluation matrices are obtained first from all the decision makers. By using equation (10), the linguistic Z-number evaluation matrices of the decision makers are normalized. As a result, a normalized linguistic evaluation matrix composed of the evaluation vectors of 20 decision makers is constructed for each alternative. Due to space limitation, only $V^{(1)}_k = (\overline{z}^{(1)}_{k1}, \overline{z}^{(1)}_{k2}, \ldots, \overline{z}^{(1)}_{k4})$, for $k = 1, 2, \ldots, 20$, i.e., the normalized linguistic evaluation vectors by the 20 decision makers for alternative E_1, are listed, which are shown in Table 1.

The similarity degrees between the decision makers' linguistic evaluation vectors of each alternative are then calculated by using equation (11). The result for alternative E_1 is shown in Table 2. By using equation (12) and the steps of modification, the decision makers for each alternative are divided into several subgroups, and the clustering results of E_1 are shown in Table 3.

Phase 2: construct the group linguistic decision matrix

For each alternative E_i, for $i = 1, 2, 3$, the linguistic Z-number evaluation vectors in $\Omega^{(i)}_h$, for $1 \le h \le {}^{(i)}_H$, of the $k^{(i)}_h$ decision makers in subgroup $\Lambda^{(i)}_h$ are aggregated into the subgroup linguistic evaluation vector $\widetilde{V}^{(i)}_h$ by using equation (14). By aggregating the subgroup linguistic evaluation vectors of all alternatives based on equation (15), the group linguistic evaluation matrix $\widehat{Z} = [\widehat{z}_{ij}]_{3\times4}$ is obtained as shown in Table 4.

Phase 3: rank the alternatives using the QUALIFLEX method

By equation (16), there are 6 (=3!) permutations of the rankings for the three candidate emergency solutions, i.e.,

$$\begin{aligned} P_1 &= (E_1, E_2, E_3),\\ P_2 &= (E_1, E_3, E_2),\\ P_3 &= (E_2, E_1, E_3),\\ P_4 &= (E_2, E_3, E_1),\\ P_5 &= (E_3, E_1, E_2),\\ P_6 &= (E_3, E_2, E_1). \end{aligned} \tag{22}$$

By using equation (17), the concordance/discordance index $I^\eta_j(E_\alpha, E_\beta)$ of each pair of solutions (E_α, E_β), for $\alpha, \beta = 1, 2, 3$, in each permutation P_η, for $\eta = 1, 2, \ldots, 6$, is calculated by considering each criterion. Because of space limitation, only the concordance/discordance indices of permutation P_1 are presented in Table 5. The weighted concordance/discordance indices $I^\eta(E_\alpha, E_\beta)$ of each permutation P_η are obtained by using equation (18), and the results of permutation P_1 are listed in Table 6. Finally, the comprehensive concordance/discordance indices I^η are calculated by using equation (19), and the results are $I^1 = 6.268$, $I^2 = 6.63$, $I^3 = -0.362$, $I^4 = 6.624$, $I^5 = 0.87$, and $I^6 = -5.754$, respectively. Hence, the best ranking of the alternatives is $E_1 \succ E_3 \succ E_2$. Therefore, E_1 is the best emergency solution.

5.2. Comparative Analysis. To demonstrate the effectiveness and practicability of the proposed linguistic Z-QUALIFLEX method, a comparative analysis is conducted. Four typical large group decision making methods at present including the interval type-2 fuzzy TOPSIS method [12], the preference conflict method [16], the large group EDM method considering experts' hesitation preference [46], and the large group decision making method based on expert's consensus [47] are used for comparison. The ranking results of the

Table 1: The normalized linguistic evaluation matrix of the 20 decision makers for alternative E_1.

DM	C_1	C_2	C_3	C_4
D_1	(A_4, B_3)	(A_3, B_2)	(A_4, B_2)	(A_1, B_4)
D_2	(A_5, B_2)	(A_4, B_2)	(A_4, B_2)	(A_4, B_2)
D_3	(A_3, B_2)	(A_4, B_3)	(A_3, B_1)	(A_3, B_2)
D_4	(A_6, B_2)	(A_2, B_3)	(A_4, B_4)	(A_1, B_4)
D_5	(A_6, B_2)	(A_1, B_3)	(A_5, B_3)	(A_3, B_2)
D_6	(A_5, B_4)	(A_2, B_3)	(A_2, B_2)	(A_2, B_4)
D_7	(A_1, B_2)	(A_1, B_1)	(A_3, B_1)	(A_2, B_1)
D_8	(A_3, B_1)	(A_3, B_1)	(A_3, B_1)	(A_3, B_1)
D_9	(A_4, B_1)	(A_3, B_1)	(A_6, B_4)	(A_1, B_1)
D_{10}	(A_6, B_4)	(A_2, B_2)	(A_3, B_0)	(A_2, B_4)
D_{11}	(A_3, B_0)	(A_3, B_0)	(A_4, B_3)	(A_3, B_1)
D_{12}	(A_6, B_4)	(A_1, B_3)	(A_2, B_1)	(A_2, B_2)
D_{13}	(A_3, B_1)	(A_3, B_1)	(A_6, B_3)	(A_3, B_1)
D_{14}	(A_6, B_4)	(A_6, B_3)	(A_3, B_2)	(A_0, B_4)
D_{15}	(A_1, B_2)	(A_1, B_2)	(A_3, B_0)	(A_1, B_2)
D_{16}	(A_5, B_2)	(A_3, B_0)	(A_4, B_1)	(A_3, B_0)
D_{17}	(A_5, B_1)	(A_4, B_1)	(A_6, B_3)	(A_3, B_1)
D_{18}	(A_6, B_3)	(A_1, B_3)	(A_5, B_3)	(A_3, B_3)
D_{19}	(A_6, B_4)	(A_1, B_1)	(A_4, B_2)	(A_2, B_1)
D_{20}	(A_3, B_2)	(A_3, B_2)	(A_4, B_2)	(A_4, B_3)

three emergency solutions obtained by these three methods are shown in Figure 2.

From Figure 2, it can be seen that the rankings of the emergency solutions obtained by the proposed method are the same as those of the interval type-2 fuzzy TOPSIS method and the large group EDM method considering experts' hesitation preference, but the rankings are different from those of the preference conflict method and the large group decision making method based on experts' consensus. Compared with the interval type-2 fuzzy TOPSIS method and the large group EDM method considering experts' hesitation preference, the calculation process is simpler by the proposed method.

The ranking result obtained with the proposed method is more practical. The reasons are as follows: in this example, first, the best solution E_1 selected by the proposed method not only ensures public safety but also saves cost. The probability of a violent rainfall causing collapse of the riverbank in area C is 0.20, which means that the probability of collapse of the riverbank is small. The solution E_3 involves sending the villager rescue team, the medical and the health team to the site, allocating a large quantity of relief supplies, and transferring the nearby villagers, which requires large quantities of manpower, material, and financial resources. If E_3 is taken as the emergency solution under the situation of a relatively small probability of collapse of the riverbank, it may not only cause wastes of the manpower, material, and financial resources but also lead to the discontent of the transferred villagers, which can minimize casualties but costs the most. E_2 cannot minimize casualties or improve personnel satisfaction, and it needs more material consumption compared with E_1. Therefore, it is more reasonable for the emergency management office to inform all kinds of emergency organizations and villagers in advance to prepare for the emergency and the transfer, and to strengthen the riverbank, keep close monitoring, and expand the emergency force of inspection and rescue at the same time just as what the alternative E_1 will do. Once a breach of the riverbank happens, the villagers will be transferred immediately, and less manpower, material, and financial resources are needed. Besides, in reality, a river levee breach occurred in Matou village in the afternoon of June 23, 2017. The whole village was in danger of flooding, and the lives and property of 2,600 people in the village were threatened. The emergency solution E_1 is also in agreement with the emergency measure taken in reality by the emergency management office to the river levee breach incident occurred in area C.

Comparing with the existing LGEDM methods, the linguistic Z-QUALIFLEX method has the following advantages: (1) the linguistic Z-numbers can describe the inherent thoughts of the decision makers more precisely and practically, ensuring the accuracy of the final results. (2) The similarity degrees and the LZAM operator are used for dividing the large group of decision makers into subgroups and for aggregating subgroups, which not only takes the conflicting nature of large group decision preferences into account but also considers the subgroup opinions under different preferences. (3) The QUALIFLEX method based on preference relation similarities is used to rank the emergency solutions. This method can handle cardinal and ordinal information in the process of decision making at the same time and can well express the complexity of the LGEDM problem. (4) Many existing large group decision making methods need to constantly recalculate expert weights or other parameters to classify expert groups, and the initial subgroups of experts or the value of some thresholds are mostly set artificially, which not only increases the calculation complexity but also may cause the decision results change due to different parameter values. The proposed method adopts an automatic threshold determination method to classify decision makers. The calculation is

Table 2: The similarity degrees between decision makers for alternative E_1.

DM	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}	D_{20}
D_1	1.000	0.572	0.239	0.646	0.288	0.517	0.186	0.227	0.233	0.413	0.161	0.296	0.206	0.342	0.237	0.402	0.277	0.289	0.252	0.643
D_2	0.572	1.000	0.296	0.301	0.333	0.313	0.182	0.233	0.228	0.271	0.182	0.296	0.212	0.232	0.210	0.484	0.301	0.338	0.250	0.513
D_3	0.239	0.296	1.000	0.181	0.373	0.192	0.274	0.455	0.494	0.173	0.246	0.227	0.288	0.177	0.242	0.271	0.280	0.200	0.174	0.439
D_4	0.646	0.301	0.181	1.000	0.508	0.667	0.156	0.179	0.182	0.517	0.138	0.350	0.172	0.444	0.187	0.265	0.232	0.446	0.335	0.346
D_5	0.288	0.333	0.373	0.508	1.000	0.508	0.210	0.286	0.255	0.425	0.206	0.600	0.275	0.318	0.272	0.322	0.281	0.625	0.514	0.272
D_6	0.517	0.313	0.192	0.667	0.508	1.000	0.158	0.185	0.184	0.642	0.140	0.417	0.174	0.444	0.194	0.258	0.213	0.458	0.502	0.383
D_7	0.186	0.182	0.274	0.156	0.210	0.158	1.000	0.392	0.347	0.178	0.329	0.234	0.338	0.115	0.575	0.265	0.256	0.187	0.571	0.236
D_8	0.227	0.233	0.455	0.179	0.286	0.185	0.392	1.000	0.633	0.232	0.500	0.331	0.833	0.132	0.383	0.504	0.508	0.246	0.244	0.275
D_9	0.233	0.228	0.494	0.182	0.255	0.184	0.347	0.633	1.000	0.231	0.272	0.266	0.467	0.131	0.301	0.330	0.342	0.242	0.254	0.340
D_{10}	0.413	0.271	0.173	0.517	0.425	0.642	0.178	0.232	0.231	1.000	0.154	0.438	0.224	0.567	0.211	0.290	0.251	0.542	0.488	0.291
D_{11}	0.161	0.182	0.246	0.138	0.206	0.140	0.329	0.500	0.272	0.154	1.000	0.249	0.583	0.115	0.317	0.403	0.527	0.171	0.201	0.187
D_{12}	0.296	0.296	0.227	0.350	0.600	0.417	0.234	0.331	0.266	0.438	0.249	1.000	0.321	0.398	0.326	0.354	0.307	0.563	0.530	0.267
D_{13}	0.206	0.212	0.288	0.172	0.275	0.174	0.338	0.833	0.467	0.224	0.583	0.321	1.000	0.124	0.356	0.470	0.675	0.238	0.234	0.254
D_{14}	0.342	0.232	0.177	0.444	0.318	0.444	0.115	0.132	0.131	0.567	0.115	0.398	0.124	1.000	0.132	0.197	0.154	0.502	0.425	0.200
D_{15}	0.237	0.210	0.242	0.187	0.272	0.194	0.575	0.383	0.301	0.211	0.317	0.326	0.356	0.132	1.000	0.374	0.265	0.245	0.310	0.263
D_{16}	0.402	0.484	0.271	0.265	0.322	0.258	0.265	0.504	0.330	0.290	0.403	0.354	0.470	0.197	0.374	1.000	0.675	0.283	0.285	0.308
D_{17}	0.277	0.301	0.280	0.232	0.281	0.213	0.256	0.508	0.342	0.251	0.527	0.307	0.675	0.154	0.265	0.675	1.000	0.233	0.240	0.259
D_{18}	0.289	0.338	0.200	0.446	0.625	0.458	0.187	0.246	0.242	0.542	0.171	0.563	0.238	0.502	0.245	0.283	0.233	1.000	0.413	0.304
D_{19}	0.252	0.250	0.174	0.335	0.514	0.502	0.571	0.244	0.254	0.488	0.201	0.530	0.234	0.425	0.310	0.285	0.240	0.413	1.000	0.220
D_{20}	0.643	0.513	0.439	0.346	0.272	0.383	0.236	0.275	0.340	0.291	0.187	0.267	0.254	0.200	0.263	0.308	0.259	0.304	0.220	1.000

Table 3: The subgroups for each of the alternatives.

Alternatives	Subgroups
E_1	$\{D_8, D_{13}\}$; $\{D_1, D_4, D_5, D_6, D_{10}, D_{18}, D_{20}\}$; $\{D_{16}, D_{17}\}$; $\{D_3, D_7, D_9, D_{12}, D_{14}, D_{19}\}$; $\{D_2, D_{11}, D_{15}\}$.
E_2	$\{D_1, D_2, D_4, D_6, D_7, D_{11}, D_{13}, D_{14}, D_{16}, D_{17}, D_{18}, D_{19}\}$; $\{D_3, D_8, D_{15}, D_{20}\}$; $\{D_5, D_9, D_{10}, D_{12}\}$.
E_3	$\{D_1, D_2, D_4, D_5, D_6, D_{10}, D_{12}, D_{13}, D_{14}, D_{15}, D_{16}, D_{17}, D_{18}, D_{19}, D_{20}\}$; $\{D_3, D_8\}$; $\{D_7, D_9, D_{11}\}$.

Table 4: The group linguistic evaluation matrix.

Alternatives	C_1	C_2	C_3	C_4
E_1	$(A_{3.400}, B_{3.997})$	$(A_{5.119}, B_{2.145})$	$(A_{4.633}, B_{4.337})$	$(A_{5.346}, B_{4.221})$
E_2	$(A_{1.118}, B_{0.889})$	$(A_{2.264}, B_{1.521})$	$(A_{1.056}, B_{0.708})$	$(A_{1.014}, B_{0.826})$
E_3	$(A_{0.991}, B_{0.880})$	$(A_{4.168}, B_{2.636})$	$(A_{1.093}, B_{1.084})$	$(A_{0.920}, B_{0.622})$

Table 5: The concordance/discordance indices for P_1.

P_1	$I_j^1(E_1, E_2)$	$I_j^1(E_1, E_3)$	$I_j^1(E_2, E_1)$	$I_j^1(E_2, E_3)$	$I_j^1(E_3, E_1)$	$I_j^1(E_3, E_2)$
C_1	2.898	3.005	−2.898	0.127	−3.005	0.127
C_2	−2.855	−3.005	2.855	−1.904	3.005	1.904
C_3	3.608	3.570	−3.608	−0.038	−3.570	0.038
C_4	4.350	4.444	−4.350	0.094	−4.444	−0.094

Table 6: The weighted concordance/discordance indices for P_1.

	$I_j^1(E_1, E_2)$	$I_j^1(E_1, E_3)$	$I_j^1(E_2, E_1)$	$I_j^1(E_2, E_3)$	$I_j^1(E_3, E_1)$	$I_j^1(E_3, E_2)$
P_1	3.312	3.391	−3.312	−0.435	−3.391	0.435
P_2	3.318	2.877	−3.318	−0.435	−2.877	0.435
P_3	3.312	2.877	−3.312	0.073	−2.877	−0.073
P_4	3.312	2.877	−3.312	−0.435	−2.877	0.435
P_5	3.312	2.877	−3.312	−0.435	−2.877	0.435
P_6	3.312	2.877	−3.312	−0.435	−2.877	0.435

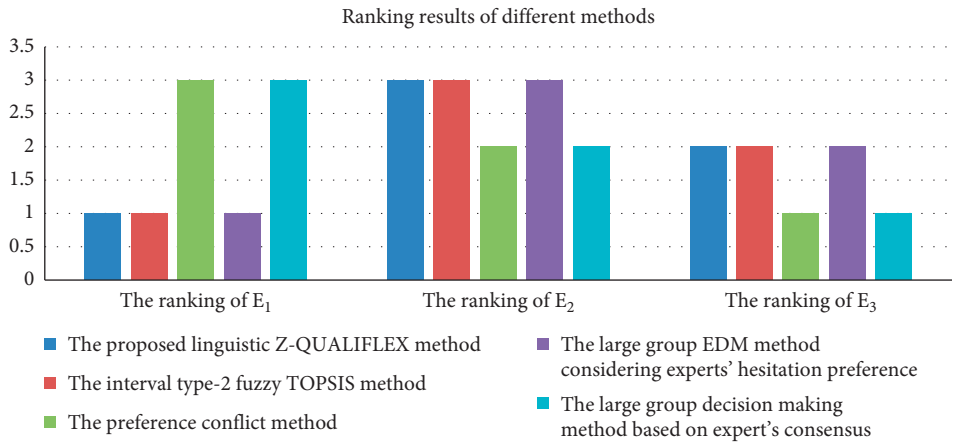


Figure 2: Ranking results of different methods.

relatively simple and helps to reduce and eliminate the decision making risk caused by subjectivity.

6. Conclusions

This study proposed an extended QUALIFLEX method using linguistic Z-numbers, called the linguistic Z-QUALIFLEX method, to deal with the LGEDM problem. Linguistic Z-numbers are first used to express the evaluations of the decision makers, which can more precisely express the inherent opinions of the decision makers. The similarity degrees between the decision makers' linguistic evaluation vectors of each alternative are then calculated and used to divide the large group of decision makers into several subgroups. The

evaluations of the decision makers in each subgroup are aggregated, and the group linguistic decision matrix is constructed by the LZAM operator. The QUALIFLEX method is finally used to rank the emergency solutions. To demonstrate the effectiveness and practicability of the proposed linguistic Z-QUALIFLEX method, a real-life example of a river levee breach incident is presented, and the obtained ranking result is compared with those of the existing LGEDM methods.

There are some improvements to be made in future studies. (1) The proposed method should be extended to support LGEDM problems by considering more complex influencing factors. (2) The complex psychological behaviors of the decision makers in the decision making process under uncertain or emergency circumstances should be considered. (3) More linguistic expression techniques should be investigated for reducing the subjectivity in the decision making process. (4) More reasonable and effective approaches for dividing a large group of decision makers into subgroups need to be developed.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (No. 71502098), the Soft Science Research Support Project of Shanghai Science and Technology Development Fund (No. 19692109000), and the Training program of School of management of Shanghai University (2020-SDGY-KZ-003).

References

[1] L. Wang, Y.-M. Wang, and L. Martínez, "A group decision method based on prospect theory for emergency situations," *Information Sciences*, vol. 418-419, pp. 119–135, 2017.

[2] P. Ji, H.-Y. Zhang, and J.-Q. Wang, "A projection-based outranking method with multihesitant fuzzy linguistic term sets for hotel location selection," *Cognitive Computation*, vol. 10, no. 5, pp. 737–751, 2018.

[3] Z.-P. Tian, J. Wang, J.-Q. Wang, and H.-Y. Zhang, "A likelihood-based qualitative flexible approach with hesitant fuzzy linguistic information," *Cognitive Computation*, vol. 8, no. 4, pp. 670–683, 2016.

[4] P. Liu and X. Zhang, "A novel picture fuzzy linguistic aggregation operator and its application to group decision making," *Cognitive Computation*, vol. 10, no. 2, pp. 242–259, 2018.

[5] J. Hu, L. Pan, and X. Chen, "An interval neutrosophic projection-based VIKOR method for selecting doctors," *Cognitive Computation*, vol. 9, no. 6, pp. 801–816, 2017.

[6] J.-Q. Wang, Y.-X. Cao, and H.-Y. Zhang, "Multicriteria decision making method based on distance measure and Choquet integral for linguistic Z-numbers," *Cognitive Computation*, vol. 9, no. 6, pp. 827–842, 2017.

[7] B. Farhadinia and Z. Xu, "Distance and aggregation-based methodologies for hesitant fuzzy decision making," *Cognitive Computation*, vol. 9, no. 1, pp. 81–94, 2017.

[8] F. Meng, C. Wang, and X. Chen, "Linguistic interval hesitant fuzzy sets and their application in decision making," *Cognitive Computation*, vol. 8, no. 1, pp. 52–68, 2016.

[9] H.-Y. Zhang, P. Ji, J.-Q. Wang, and X.-H. Chen, "A neutrosophic normal cloud and its application in decision making," *Cognitive Computation*, vol. 8, no. 4, pp. 649–669, 2016.

[10] X.-H. Xu, Z.-J. Du, and X.-H. Chen, "Consensus model for multicriteria large group emergency decision making considering non-cooperative behaviors and minority opinions," *Decision Support Systems*, vol. 79, pp. 150–160, 2015.

[11] Y. Liu, Z.-P. Fan, and Y. Zhang, "Risk decision analysis in emergency response: a method based on cumulative prospect theory," *Computers & Operations Research*, vol. 42, pp. 75–82, 2014.

[12] Z. Wu and J. Xu, "A consensus model for large scale group decision making with hesitant fuzzy information and changeable clusters," *Information Fusion*, vol. 41, pp. 217–231, 2018.

[13] A. Pagano, I. Pluchinotta, R. Giordano, A. B. Petrangeli, U. Fratino, and M. Vurro, "Dealing with uncertainty in decision making for drinking water supply systems exposed to extreme events," *Water Resources Management*, vol. 32, no. 6, pp. 2131–2145, 2018.

[14] B. Sun, W. Ma, B. Li, and X. Li, "Three-way decisions approach to multiple attribute group decision making with linguistic information-based decision-theoretic rough fuzzy set," *International Journal of Approximate Reasoning*, vol. 93, pp. 424–442, 2018.

[15] S. Bingzhen and M. Weimin, "An approach to evaluation of emergency plans for unconventional emergency events based on soft fuzzy rough set," *Kybernetes*, vol. 45, no. 3, pp. 461–473, 2016.

[16] C.-G. Cai, X.-H. Xu, P. Wang, and X.-H. Chen, "A multistage conflict style large group emergency decision making method," *Soft Computing*, vol. 21, no. 19, pp. 5765–5778, 2017.

[17] J. He, C. Feng, D. Hu, and L. Liang, "A decision model for emergency warehouse location based on a novel stochastic mcda method: evidence from China," *Mathematical Problems in Engineering*, vol. 2017, Article ID 7804781, 10 pages, 2017.

[18] F. Jin, Z. Ni, H. Chen, and Y. Li, "Approaches to decision making with linguistic preference relations based on additive consistency," *Applied Soft Computing*, vol. 49, pp. 71–80, 2016.

[19] Y. Xu, W. Zhang, and H. Wang, "A conflict-eliminating approach for emergency group decision of unconventional incidents," *Knowledge-Based Systems*, vol. 83, pp. 92–104, 2015.

[20] Y. Xu, F. Ma, W. Xu, and H. Wang, "An incomplete multigranular linguistic model and its application in emergency decision of unconventional outburst incidents," *Journal of Intelligent & Fuzzy Systems*, vol. 29, no. 2, pp. 619–633, 2015.

[21] B. Liu, Y. Shen, X. Chen, H. Sun, and Y. Chen, "A complex multiattribute large group PLS decision making method in the interval-valued intuitionistic fuzzy environment," *Applied Mathematical Modelling*, vol. 38, no. 17-18, pp. 4512–4527, 2014.

[22] B. Liu, Y. Shen, W. Zhang, X. Chen, and X. Wang, "An interval-valued intuitionistic fuzzy principal component analysis model-based method for complex multiattribute large

group decision making," *European Journal of Operational Research*, vol. 245, no. 1, pp. 209–225, 2015.

[23] L. Zhang, Y. Wang, and X. Zhao, "A new emergency decision support methodology based on multisource knowledge in 2-tuple linguistic model," *Knowledge-Based Systems*, vol. 144, pp. 77–87, 2018.

[24] S. Xian, J. Chai, and H. Guo, "Linguistic-induced ordered weighted averaging operator for multiple attribute group decision making," *International Journal of Intelligent Systems*, vol. 34, no. 2, pp. 271–296, 2019.

[25] J. H. P. Paelinck, "Qualiflex: a flexible multiple criteria method," *Economics Letters*, vol. 1, no. 3, pp. 193–197, 1978.

[26] A. Isik and E. Adali, "The qualiflex method for the insurance company selection problem," *European Scientific Journal*, vol. 12, no. 10, pp. 348–358, 2016.

[27] X.-F. Ding, L. Zhang, and H.-C. Liu, "Emergency decision making with extended axiomatic design approach under picture fuzzy environment," *Expert Systems*, vol. 37, no. 2, 2020.

[28] X.-F. Ding and H.-C. Liu, "An extended prospect theory-VIKOR approach for emergency decision making with 2-dimension uncertain linguistic information," *Soft Computing*, vol. 23, no. 22, pp. 12139–12150, 2019.

[29] X.-F. Ding and H.-C. Liu, "A new approach for emergency decision making based on zero-sum game with Pythagorean fuzzy uncertain linguistic variables," *International Journal of Intelligent Systems*, vol. 34, no. 7, pp. 1667–1684, 2019.

[30] M.-Y. Li and P.-P. Cao, "Extended TODIM method for multiattribute risk decision making problems in emergency response," *Computers & Industrial Engineering*, vol. 135, pp. 1286–1293, 2019.

[31] X. Peng, R. Krishankumar, and K. S. Ravichandran, "Generalized orthopair fuzzy weighted distance-based approximation (WDBA) algorithm in emergency decision making," *International Journal of Intelligent Systems*, vol. 34, no. 10, pp. 2364–2402, 2019.

[32] J. Gao, Z. Xu, Z. Liang, and H. Liao, "Expected consistency-based emergency decision making with incomplete probabilistic linguistic preference relations," *Knowledge-Based Systems*, vol. 176, pp. 15–28, 2019.

[33] P. Li and C. Wei, "An emergency decision making method based on D-S evidence theory for probabilistic linguistic term sets," *International Journal of Disaster Risk Reduction*, vol. 37, p. 101178, 2019.

[34] Y. Song and G. Li, "Consensus constructing in large scale group decision making with multigranular probabilistic 2-tuple fuzzy linguistic preference relations," *IEEE Access*, vol. 7, pp. 56947–56959, 2019.

[35] X. Xu, L. Wang, X. Chen, and B. Liu, "Large group emergency decision making method with linguistic risk appetites based on criteria mining," *Knowledge-Based Systems*, vol. 182, pp. 1–13, 2019.

[36] X. Xu, Y. Huang, and K. Chen, "Method for large group emergency decision making with complex preferences based on emergency similarity and interval consistency," *Natural Hazards*, vol. 97, no. 1, pp. 45–64, 2019.

[37] X.-H. Xu, Z.-J. Du, X.-H. Chen, and C.-G. Cai, "Confidence consensus-based model for large scale group decision making: a novel approach to managing noncooperative behaviors," *Information Sciences*, vol. 477, pp. 410–427, 2019.

[38] X.-H. Xu, X.-Y. Zhong, X.-H. Chen, and Y.-J. Zhou, "A dynamical consensus method based on exit-delegation mechanism for large group emergency decision making," *Knowledge-Based Systems*, vol. 86, pp. 237–249, 2015.

[39] L. A. Zadeh, "A note on *Z*-numbers," *Information Sciences*, vol. 181, no. 14, pp. 2923–2932, 2011.

[40] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning-I," *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[41] D. B. Liu, *Uncertainty Theory*, Springer Berlin Heidelberg, Heidelberg, Germany, 2007.

[42] X. Zhang, "Multicriteria Pythagorean fuzzy decision analysis: a hierarchical QUALIFLEX approach with the closeness index-based ranking methods," *Information Sciences*, vol. 330, pp. 104–124, 2016.

[43] T.-Y. Chen, "Interval-valued intuitionistic fuzzy QUALIFLEX method with a likelihood-based comparison approach for multiple criteria decision analysis," *Information Sciences*, vol. 261, pp. 149–169, 2014.

[44] J.-J. Peng, J.-Q. Wang, and W.-E. Yang, "A multivalued neutrosophic qualitative flexible approach based on likelihood for multicriteria decision making problems," *International Journal of Systems Science*, vol. 48, no. 2, pp. 425–435, 2017.

[45] J.-Q. Wang, J.-J. Kuang, J. Wang, and H.-Y. Zhang, "An extended outranking approach to rough stochastic multi-criteria decision making problems," *Cognitive Computation*, vol. 8, no. 6, pp. 1144–1160, 2016.

[46] X. Xu and Y. Yu, "Incomplete information complement method based on the hesitation of experts in large group emergency decision making," *Information and Control*, vol. 48, no. 6, pp. 678–686, 2019.

[47] P. Wang, X. Xu, and S. Huang, "An improved consensus-based model for large group decision making problems considering experts with linguistic weighted information," *Group Decision and Negotiation*, vol. 28, no. 3, pp. 619–640, 2019.

Hindawi
Scientific Programming
Volume 2020, Article ID 8403262, 10 pages
https://doi.org/10.1155/2020/8403262

Research Article

Object Detection through Modified YOLO Neural Network

Tanvir Ahmad,[1] **Yinglong Ma**,[1] **Muhammad Yahya**,[2] **Belal Ahmad**,[3] **Shah Nazir**,[4] **and Amin ul Haq**[5]

[1]*School of Control and Computer Engineering, North China Electric Power University, Beijing, China*
[2]*Department of Electrical and Electronics, Universiti Kuala Lumpur, Kuala Lumpur, Malaysia*
[3]*Embedded and Pervasive Computing (EPIC) Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China*
[4]*Department of Computer Science, University of Swabi, Ambar, Khyber Pakhtunkhwa, Pakistan*
[5]*School of Computer Science and Engineering, University of Electronic Science and Technology, Chengdu, China*

Correspondence should be addressed to Tanvir Ahmad; tanvirahmad@ncepu.edu.cn

Received 7 December 2019; Revised 3 February 2020; Accepted 22 February 2020; Published 6 June 2020

Guest Editor: Rahman Ali

In the field of object detection, recently, tremendous success is achieved, but still it is a very challenging task to detect and identify objects accurately with fast speed. Human beings can detect and recognize multiple objects in images or videos with ease regardless of the object's appearance, but for computers it is challenging to identify and distinguish between things. In this paper, a modified YOLOv1 based neural network is proposed for object detection. The new neural network model has been improved in the following ways. Firstly, modification is made to the loss function of the YOLOv1 network. The improved model replaces the margin style with proportion style. Compared to the old loss function, the new is more flexible and more reasonable in optimizing the network error. Secondly, a spatial pyramid pooling layer is added; thirdly, an inception model with a convolution kernel of $1 * 1$ is added, which reduced the number of weight parameters of the layers. Extensive experiments on Pascal VOC datasets 2007/2012 showed that the proposed method achieved better performance.

1. Introduction

Human beings can easily detect and identify objects in their surroundings, without consideration of their circumstances, no matter what position they are in and whether they are upside down, different in color or texture, partly occluded, etc. Therefore, humans make object detection look trivial. The same object detection and recognition with a computer require a lot of processing to extract some information on the shapes and objects in a picture.

In computer vision, object detection refers to finding and identifying an object in an image or video. The main steps involved in object detection include feature extraction [1], feature processing [2–4], and object classification [5]. Object detection achieved excellent performance with many traditional methods that can be described from the following four aspects: bottom feature extraction, feature coding, feature aggregation, and classification. The feature extraction plays an essential role in the object detection and recognition process [6]. There will be more redundant information which can be modeled to achieve better performance than previous point-of-interest detection. Previously used scale-invariant feature transformations (SIFT) [7] and histogram of oriented gradients (HOG) [8] belong to this category.

The object detection is critical in different applications, such as surveillance, cancer detection, vehicle detection, and underwater object detection. Various techniques have been used to detect the object accurately and efficiently for different applications. However, these proposed methods still have problems with a lack of accuracy and efficiency. To tackle these problems of the object detection, machine learning and deep neural network methods are more effective in correcting object detection.

Thus, in this study, a modified new network is proposed based on the YOLOv1 [9] network model. The performance

of the modified YOLOv1 is improved through the following points:

(i) The loss function of the YOLOv1 network is optimized.

(ii) The inception model structure is added.

(iii) A spatial pyramid pooling layer is used.

(iv) The proposed model effectively extracts features from images, performing much better in object detection.

The remaining of this paper is organized as follows. Section 2 describes related work. Section 3 presents the methodology, which describes network architecture in detail. Section 4 presents the analysis of the improved network from various aspects. In Section 5, the experiment setup, results, and comparison with other networks are discussed. The paper conclusion and future work are given in Section 6.

2. Related Work

Detecting and identifying multiple objects in an image is hard for machines to recognize and classify. However, a noteworthy effort has been carried out in the past years in the detection of objects using convolutional neural networks (CNNs). In the object detection and recognition field, neural networks are in use for a decade but became prominent due to the improvement of hardware new techniques for training these networks on large datasets [10, 11]. In object detection and recognition, researchers have used deep learning for learning features directly from the image pixels, which are more effective than the manual features [4, 12]. Recently deep learning-based algorithms remove the manual features extraction methods and directly use features extracting methods [13] from the original images. This methodology has been successfully proven in feature pyramid network (FPN) [14], single shot detector (SSD) [15], and deconvolutional single shot detector (DSSD) [16]. Deep learning is a prevailing direction in the field of machine learning [17]. In [18, 19], researchers showed that the CNNs inherit the advantages of deep learning, which makes their results in the field of object detection and recognition greatly improved compared with the traditional methods. Researchers had made many efforts to use stochastic gradient descent and backpropagation to train deep networks for object detection [20]. Those networks were able to learn but were too slow in practice to be useful in real-time applications; the technique in [12] showed that stochastic gradient descent by backpropagation was effective in training CNNs. CNNs became in use but fell out of fashion due to the support vector machine as in [21] and other simpler methods like linear classifiers as in [22]. New techniques that have been developed recently [23, 24] show higher image classification accuracy in ImageNet large scale visual recognition [25]. These techniques have brought much more easiness to train large and deeper networks and shown enhanced performance. Newly, approaches have been established to identify vehicles and other objects from videos or static images using deep convolutional neural networks (DCNN) [26–30]. For example, faster R-CNN [19] proposes candidate regions and uses CNN to confirm candidates as valid objects. YOLO uses end-to-end unified, fully convolutional network structure that predicts the objectless assurance and the bounding boxes concurrently over the whole image. SSD [31] outperforms YOLO by discretizing the production space of bounding boxes into a set of avoidance boxes over different feature ratios and scales per feature map location. YOLO-2 [32] achieves state-of-the-art performance in object detection by improving various aspects of its earlier version. A fully convolutional network is utilized for object detection from three-dimensional (3D) range scan data with LIDAR. A 2D-DBN design is proposed, which uses second-order planes instead of first-order vectors as inputs and uses bilinear projection for retaining discriminative information to develop the recognition rate [33]. Although DCNN based approaches accomplish the state-of-the-art accuracy of detection or classification, these approaches often require intensive calculation and a considerable amount of labeled training data. Through the past few years, to use deep neural networks economically in real-time applications, a substantial amount of work has been done to report these two problems [34, 35]. In this study, a different modified architecture for object detection is addressed, which is capable of providing high accuracy and speed.

3. Methodology

In this section, the proposed model is described in detail. Firstly, the improvement based on loss function is presented. Secondly, the improvement based on inception structure model is described. And lastly, the improvement based on the spatial pyramid pooling layer is portrayed. The symbolic representations are described in Table 1.

3.1. Improvement in Network Design. The following improvements to the YOLO network model are made while maintaining the original model dominant idea.

3.1.1. Improvement Based on Loss Function. The loss function of the original YOLOv1 network takes the same error for the large and small objects, which makes the model's prediction for neighboring objects unsatisfactory. If two objects appear in the same grid, only one object can be detected, and there will be a problem in detecting small objects. Compared with the old loss function, the new loss function is more flexible and optimized. In the new loss function, the original difference is replaced by the proportionality. Equation (1) shows the original loss function of YOLOv1; YOLOv1 uses one single loss function for both bounding boxes and the classification of the object. Loss function can be described in five parts: the first and second are focusing on the loss of the bounding box coordinates, while the third and fourth are responsible for the difference in the confidence of having an object in the grid, and part five is responsible for the difference in class probability. The λ_{coord} and λ_{noobj} are scalars to weight each loss function,

Table 1: The mathematical symbols.

Symbol/ notation	Description
λ_{coord}	Hyperparameter is set to ensure a "fair" contribution of the bounding box location
λ_{noobj}	Hyperparameter is set for bounding box score prediction
C	Categories
$P(C)$	Probability of detected class categories
N	Training samples
X	Training input
Y	Output label
t	Input label

λ_{coord} is set to 5, and λ_{noobj} is set to 0.5 by the original author of YOLOv1.

$$\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\text{obj}} \left(x_i - x_i'\right)^2 + \left(y_i - y_i'\right)^2 \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\text{obj}} \left(\sqrt{w_i} - \sqrt{w_i'}\right)^2 + \left(\sqrt{h_i} - \sqrt{h_i'}\right)^2 \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij} \left(c_i - c_i'\right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\text{noobj}} \left(c_i - c_i'\right)^2 \\
&+ \sum_{i=0}^{S^2} I_{ij} \sum_{c \in \text{class}} \left(p_i(c) - p_i'(c)\right)^2.
\end{aligned} \tag{1}$$

In convolutional neural networks, variance function is often used as the loss function [36] of the network. For example, for a variety of problems, the total number of categories is C and training samples is N. The algorithm which is used for multiclassification first needs to find those weights and biases that make the output of the neural network close to $y(x)$ (which is labeled category) for all training inputs x; to quantify how close the output of all training inputs x is to $y(x)$, the loss function is defined as

$$E^N = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{C} \left(t_k^n - y_k^n\right)^2. \tag{2}$$

Here, t represents the label of the input object, and y represents the actual output value of the input object to the network. The function of choosing the variance form is the loss function to facilitate subsequent optimization. On the other hand, the current training level can be predicted by observing the severity of the fluctuation of the loss value in practice.

In the YOLOv1 network loss function design, the variance function is used as part of the entire loss function, the normalization idea of contrast is used to improve it, and the improved model replaces margin style with proportion style, so here the size of the object in the picture is considered. The specific modified loss function is shown in

$$\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\text{obj}} \left(x_i - x_i'\right)^2 + \left(y_i - y_i'\right)^2 \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\text{obj}} \left(\frac{w_i - w_i'}{w_i'}\right)^2 + \left(\frac{h_i - h_i'}{h_i'}\right)^2 + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \left(c_i - c_i'\right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\text{noobj}} \left(c_i - c_i'\right)^2 + \sum_{i=0}^{S^2} I_{ij}^{\text{obj}} \sum_{c \in \text{class}} \left(p_i(c) - p_i'(c)\right)^2.
\end{aligned} \tag{3}$$

Here, I_{ij}^{obj} indicates that the target object is assumed to be present in the i^{th} position of the area. x and y represent the current position of the image; w and h represent the width and height of the image. c is the total number of objects to be identified, and $p\ (c)$ is the probability that the object belongs to a specific class c. Here, it should be noted that the loss function guides the optimization of the class to which the object belongs and optimizes the position of the boundary box for detecting the object.

3.1.2. Improvement of Inception Structure Model. The third and fourth layers of the original network are replaced with new inception models. The inception model itself has the ability to deepen and widen the network and enhance the network; a $64 \times 1 \times 1$ layer is added between the first and second layers of the original network, which reduces the network parameters. Figure 1 shows the structure part of the YOLOv1 network after adding the inception model. Inception architecture is used to find out how an optimal local sparse structure in a convolutional neural network can be approximated and covered by readily available dense components.

The inception model can deepen and widen the network, and the convolutional kernel of different scales is connected in parallel. Thus, the multiscale feature can be more effective, and the hidden information in the image can be used more efficiently.

3.1.3. Improvement of SPP Structure Model. Figure 2 shows the addition of spatial pyramid pooling (SPP) layer, and below are the advantages of using it.

(i) It can output a fixed-size image for any size input or any ratio of the input image.

(ii) It can extract pool features at varying scales.

A classifier (SVM/Softmax), as well as fully connected layers, requires a fixed-length vector, which can be generated through Bag-of-Words (BoW) [35, 37, 38], the spatial pyramid downsampling boosts the BoW because it preserves spatial information by pooling the spatial bins. These spatial bins have sizes proportional to the image size, so the number of bins is fixed regardless of the image size, which makes the SPP [39, 40] not only improve network performance but also dramatically reduce the required calculation time by avoiding repeatedly computing the convolutional features.

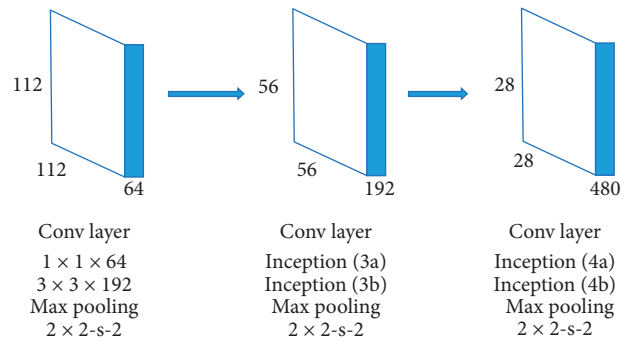


Figure 1: Partial structure of the new network after adding inception.

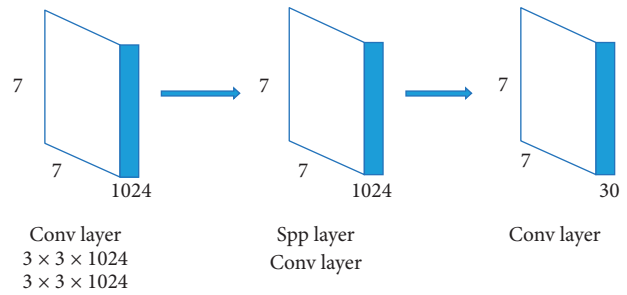


Figure 2: Partial structure of the new network after joining the SPP layer.

By using the SPP layer, more feature-rich image information is obtained, and also great improvements in the network's time efficiency are observed. Hence, this technique shows remarkable detection accuracy.

4. Analysis of the Network

Following is the comprehensive analysis of our proposed network and improved YOLO model based on the results of the experimental tests.

(i) By the analysis of the confusion matrix, we observed what kind of sample detection performance is better for the new network, what kind of sample detection performance is not good, and how to distinguish the easily confused categories and understand the advantages and disadvantages of the network.

(ii) We examined the network architecture of the new network model, such as the comparison of the number of network parameters, and assessed its performance.

4.1. Confusion Matrix. Through the confusion matrix, the test results are analyzed. A confusion matrix is a list of data classes; in each class, the actual data is classified so that we can observe which categories of samples are easily confused in the modified network. In the confusion matrix, the rows represent the true categories of the test images. The columns show the classes of the test images divided by the network in the actual test.

In the original Pascal VOC dataset, there are 20 categories of objects; here some representative categories, which easily cause misidentification, are selected.

Table 2 is the confusion matrix of the modified network model on the Pascal VOC 2007 dataset. It can be noticed from Table 2 that the airplane is mistakenly recognized as a bird, and the original samples belonging to birds are identified as airplanes. The reason is that the overall shape is too similar: the airplane has two wings, and so does the bird; the airplane's body shape is very similar to that of a bird; therefore, the results show that 22% of the airplanes are mistakenly identified as birds, and 36% of the birds were incorrectly identified as airplanes. In addition, the chair and sofa are also relatively easy to cause misidentification, because in real life it is very easy to differentiate between chairs and sofas, but in picture chairs and sofas are very easy to appear the same, which can cause miss identification very easily. And the same applies for sheep, horses, dogs, and cats.

From Table 2, it can be seen that the overall average misrecognition rate is not too high, indicating that the overall ability of the network to extract features and detect target objects in the image is relatively reliable.

4.2. Network Architecture. Here, the proposed network architecture is described. Before going into detail, please note that the first and second layers are the same: both are convolutional layers plus the downsampling layer structure; the third and fourth layers are the same: both are inception + pool structures; the fifth and sixth layers are the same: both are convolutional cascade structures; the seventh layer is spatial pyramid pooling layer; and the eighth and ninth layers are the fully connected layers.

For the first layer, it is assumed that the input is an image, r is the number of rows of the image, c is the number of columns of the image to a network of the first layer input, and the sliding step is s_1; the computational cost of obtaining a feature map is shown in the equation:

$$(x_1 y_1)\frac{r - x_1 + s_1}{s_1}\frac{c - y_1 + s_1}{s_1}. \tag{4}$$

Computing area is the size of the convolution kernel area, so the result of (4) is obtained, and then we assume that the first layer has n_1 feature maps, so the calculation of the first layer is

$$\frac{r - x_1 + s_1}{s_1}\frac{c - y_1 + s_1}{s_1}(x_1 y_1 n_1), \tag{5}$$

and the size of the feature map after convolution will become

$$\begin{aligned} r_1 &= \left\lceil \frac{r - x_1 + s_1}{s_1} \right\rceil, \\ c_1 &= \left\lceil \frac{c - y_1 + s_1}{s_1} \right\rceil. \end{aligned} \tag{6}$$

Next is the maximum downsampling layer; since the downsampling layer does not change the number of feature maps, the numbern_2 of the feature maps is equal to the

Table 2: Confusion matrix for the new network.

(%)	Bird	Chair	Sofa	Aero	Horse	Sheep	Dog	Cat
Bird	64			36				
Chair		43	57					
Sofa		41	59					
Aero	22			78				
Horse					78	22		
Sheep					28	72		
Dog							82	18
Cat							14	86

number n_1 of the previous feature maps. Assuming the size of the downsampling window, the size of the feature map obtained after downsampling is

$$x_2 = \frac{r_1}{r_2}, \quad y_2 = \frac{c_1}{c_2}. \tag{7}$$

The calculation of the total number of n_2 feature maps will become

$$n_2 r_2 c_2 x_2 y_2. \tag{8}$$

The following is the convolution second layer, assuming that the number of features

$$r_3 = \left\lceil \frac{r_2 - x_3 + s_3}{s_3} \right\rceil, \quad c_3 = \left\lceil \frac{c_2 - y_3 + s_3}{s_3} \right\rceil. \tag{9}$$

The calculation with the upper layer of the feature map for convolution operation will be as follows.

$$(x_3 y_3 n_3) \left\lceil \frac{r_2 - x_3 + s_3}{s_3} \right\rceil \left\lceil \frac{c_2 - y_3 + s_3}{s_3} \right\rceil. \tag{10}$$

Assuming that the output of the maximum downsampling layer in the second layer is characterized by the size of the downsampling window and with the step size s_4, calculation of the total amount of the layer can be obtained by the same way.

$$x_4 = \frac{r_3}{r_4}, \quad y_4 = \frac{c_3}{c_4}. \tag{11}$$

From the above, it can be seen that the output feature size of MaxPool2 is n_4. In the inception structure, the step size is 1, and the calculation is from left to right. The third layer's inception structure model is shown in Figure 3 and mathematically shown in

$$\begin{aligned} &n_4 \times 1 \times 1 \times 64, \\ &n_4 \times 1 \times 1 \times 96 + 96 \times 3 \times 3 \times 128, \\ &n_4 \times 1 \times 1 \times 16 + 16 \times 5 \times 5 \times 32, \\ &n_4 \times r_4 \times c_4 + n_4 \times 1 \times 1 \times 32. \end{aligned} \tag{12}$$

Thus, the whole calculation of inception four layers can be done in the above way. Next is the fifth layer of the convolution, and the total calculation is

$$\begin{aligned} &1 \times 1 \times r_5 \times c_5 \times 512 + 2 \times 3 \times 3 \times (r_5 - 3 + 1) \\ &\times (c_5 - 3 + 1) \times 1024 \\ &+ 3 \times 3 \times \left(\frac{r_5 - 3 + 2}{2}\right)\left(\frac{c_5 - 3 + 2}{2}\right) \times 1024. \end{aligned} \tag{13}$$

Since the sixth layer and the fifth layer have the same structure, the calculation is the same as (13).

The seventh layer is the pyramid layer, denoted by L, where $n = 1, 2, \ldots, L$. The calculation amount of the pyramid layer is

$$\sum_{n=1}^{L} \left(n_5 r_{6,n} c_{6,n} \left\lceil \frac{r_5}{r_{6,n}} \right\rceil \left\lceil \frac{c_5}{c_{6,n}} \right\rceil \right). \tag{14}$$

The eighth layer is fully connected. Assume that the number of input features is n_6, and the number of output features isn_7. Because the input of the layer is the former layer, it will be processed after all the features of the map are gathered as a vector, so $n_6$is

$$n_6 = \sum_{n=1}^{L} n_5 r_{6,n} c_{6,n}. \tag{15}$$

Because the full-connection layer is derived from the original neural network, the calculation method is the same as that of the neural network, so the computational cost of the layer is

$$n_7 \left(2 \sum_{n=1}^{L} n_5 r_{6,n} c_{6,n} + 1 \right). \tag{16}$$

From the above description of network architecture analysis, it is observed that the network's overall calculation, input layer image size, convolution kernel size, and the number of convolutional layers, shows that network depth and width are having big impact.

5. Experiment

Pascal VOC is divided into two datasets: Pascal VOC 2007 and Pascal VOC 2012 dataset. The newly designed network was tested on both datasets [41]. The Pascal VOC dataset consist of 20 categories: person, bird, cat, cow, horse, sheep, airplane, bike, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, and TV monitor. Figures 4 and 5 show the sample images.

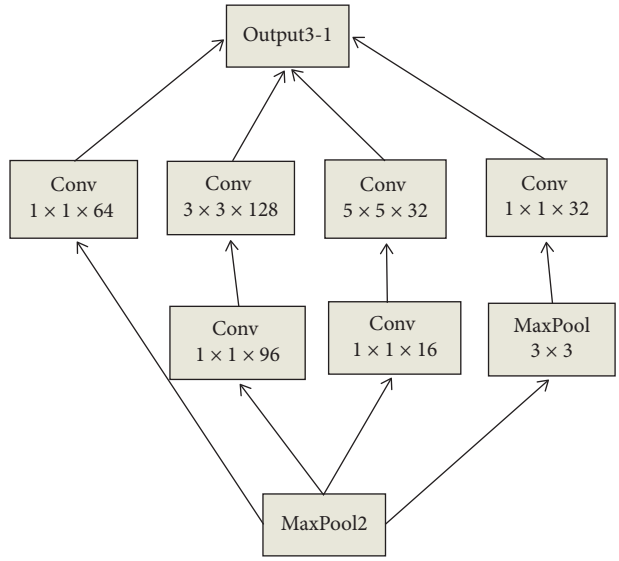


Figure 3: Inception model architecture.

(a) (b) (c)

(d) (e) (f)

Figure 4: Pascal VOC 2007 dataset images. (a) 000006. (b) 000008. (c) 000014. (d) 000015. (e) 000017. (f) 000018.

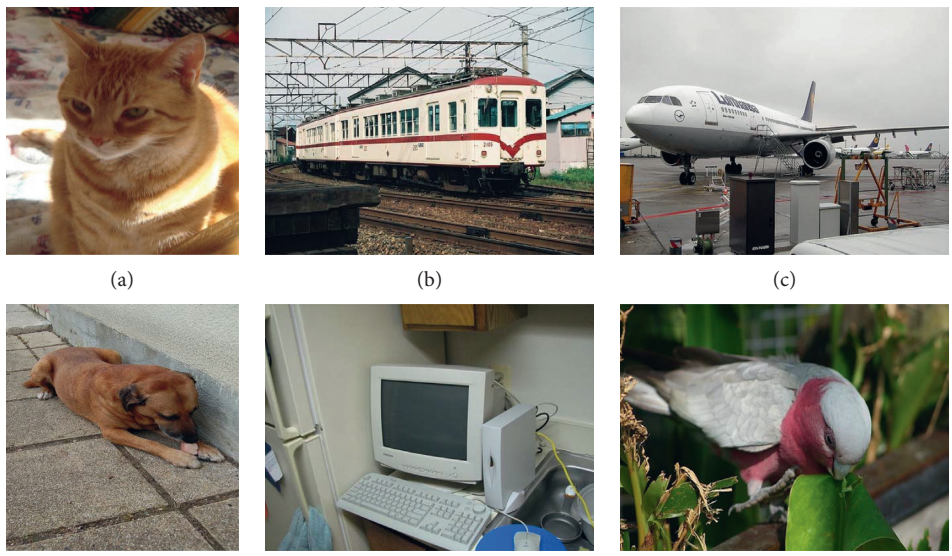
The whole experiment process is conducted on NVIDIA GeForce GTX 1060 GPU using the Ubuntu operating system. The number of iterations was 40000.

5.1. Results and Discussion. The results are discussed and the network performance is checked using t-SNE visualization tool, showing the extent to which the new network is able to extract rich features from images.

Next, the visualization of a large number of sample features in 2D is observed by using the t-SNE visualization tool, which maps high-dimensional to low-dimensional data [42].

Figure 6 shows ten selected categories from the Pascal VOC dataset (bird, chair, sofa, bike, airplane, horse, sheep, dog, cat, cow) using the t-SNE visualization tool; in the figure, different colors represent different types; if the two types are fused, this means that these types are easily getting confused with one another.

There are about seven categories which are not compatible with each other, indicating that the characteristics of these seven types of differences are relatively large and relatively easy to identify; in addition to several types of partial integration, the characteristics of several types have a certain degree of similarity, which is easy to cause misidentification. However, overall, the use of the new network to extract the characteristics is very effective and robust, but it is also inadequate and needs to be further improved. The improved network was tested on Pascal VOC 2007 and



(a) (b) (c)

(d) (e) (f)

Figure 5: Pascal VOC 2012 dataset images. (a) 000028. (b) 000031. (c) 000033. (d) 000036. (e) 000039. (f) 000040.

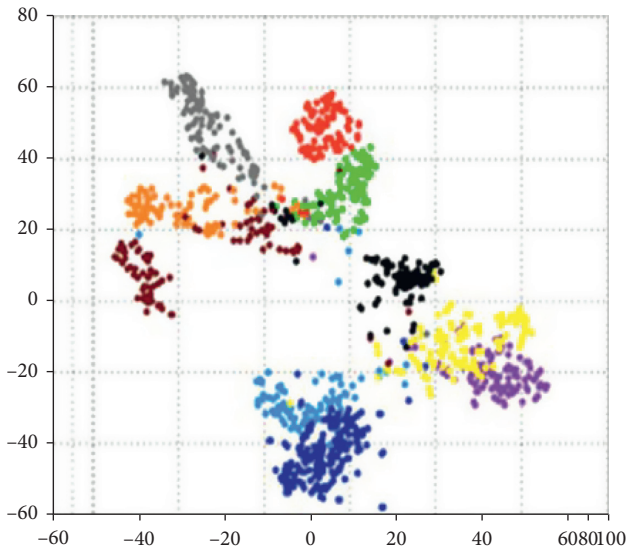


Figure 6: Two-dimensional visualization of ten samples.

Pascal VOC 2012, respectively. The results are shown in Tables 3 and 4.

The data in Tables 3 and 4 is expressed in percentage. In the above results, to make the comparison results more consistent, the training dataset used in the above algorithm is the train/val dataset of Pascal VOC 2007 and Pascal VOC 2012. The data presented in Tables 3 and 4 are test results for each class of 20 objects. Our modified network average detection rate is 65.6% and 58.7% on the Pascal VOC 2007 and 2012 dataset. To check the performance, we compared the results of our modified network with those of R-CNN and YOLOv1, as depicted in Tables 5 and 6 for Pascal VOC 2007 and 2012, respectively. Table 5 shows the Pascal VOC 2007 comparison test results, and in Table 6 Pascal VOC 2012 comparative test results are presented.

Table 3: Pascal VOC 2007 test results.

VOC 2007	The modified YOLOv1	VOC 2007	The modified YOLOv1
Aero	77.9	Table	51.2
Bike	77.6	Dog	81.9
Bird	63.7	Horse	77.5
Boat	47.6	M-bike	78.7
Bottle	44.8	Person	68.6
Bus	70.7	Plant	37.1
Car	68.9	Sheep	71.8
Cat	85.3	Sofa	58.4
Chair	42.2	Train	71.0
Cow	71.9	Tv	64.6
Average recognition rate			65.6

Table 4: Pascal VOC 2012 test results.

VOC 2012	The modified YOLOv1	VOC 2012	The modified YOLOv1
Aero	76.1	Table	49.1
Bike	67.8	Dog	80.3
Bird	58.0	Horse	72.7
Boat	39.9	M-bike	71.9
Bottle	24.2	Person	64.2
Bus	68.9	Plant	29.0
Car	57.6	Sheep	54.5
Cat	82.5	Sofa	55.2
Chair	36.3	Train	73.9
Cow	61.1	Tv	51.7
Average recognition rate			58.7

It can be seen from the tables that our modified model has improved recognition over the YOLOv1 and R-CNN model in almost every type. Table 7 depicts the processing time of an image of three different networks, R-CNN, YOLOv1, and our improved YOLO, for testing the same image. The time taken by the R-CNN network is 6.9 seconds,

Table 5: Pascal VOC 2007 comparison test results.

VOC 2007	R-CNN	YOLOv1	The modified YOLOv1
Aero	63.5	78	77.9
Bike	66	74.2	77.6
Bird	47.9	61.3	63.7
Boat	37.7	45.7	47.6
Bottle	29.9	42.7	44.8
Bus	62.5	68.2	70.7
Car	70.2	66.8	68.9
Cat	60.2	80.2	85.3
Chair	32	40.6	42.2
Cow	57.9	70	71.9
Table	47	49.8	51.2
Dog	53.5	79	81.9
Horse	60.1	74.5	77.5
M-bike	64.2	77.9	78.7
Person	52.2	64	68.6
Plant	31.3	35.3	37.1
Sheep	55	67.9	71.8
Sofa	50	55.7	58.4
Train	57.7	68.7	71
TV	63	62.6	64.6
Average recognition rate	53.1	63.4	65.6

Table 6: Pascal VOC 2012 comparative test results.

VOC 2012	R-CNN	YOLOv1	The modified YOLOv1
Aero	68.1	77	76.1
Bike	63.8	64.2	67.8
Bird	46.1	57.7	58
Boat	29.4	38.3	39.9
Bottle	27.9	22.7	24.2
Bus	56.6	68.3	68.9
Car	57	55.9	57.6
Cat	65.9	81.4	82.5
Chair	26.5	36.2	36.3
Cow	48.7	60.8	61.1
Table	39.5	48.5	49.1
Dog	66.2	77.2	80.3
Horse	57.3	72.3	72.7
M-bike	65.4	71.3	71.9
Person	53.2	63.5	64.2
Plant	26.2	28.9	29
Sheep	54.5	52.2	54.5
Sofa	38.1	54.8	55.2
Train	50.6	73.9	73.9
TV	51.6	50.8	51.7
Average recognition rate	49.6	57.9	58.7

Table 7: Comparison of test results for time performance.

Device	R-CNN (s)	YOLOv1 (s)	The modified YOLOv1 (s)
GPU time/image	6.9	0.14	0.11

the YOLO network takes 0.14 seconds, and our model takes 0.11 seconds. Figures 7 and 8 show the testing results on Pascal VOC 2007 and Pascal VOC 2012 dataset images [41].

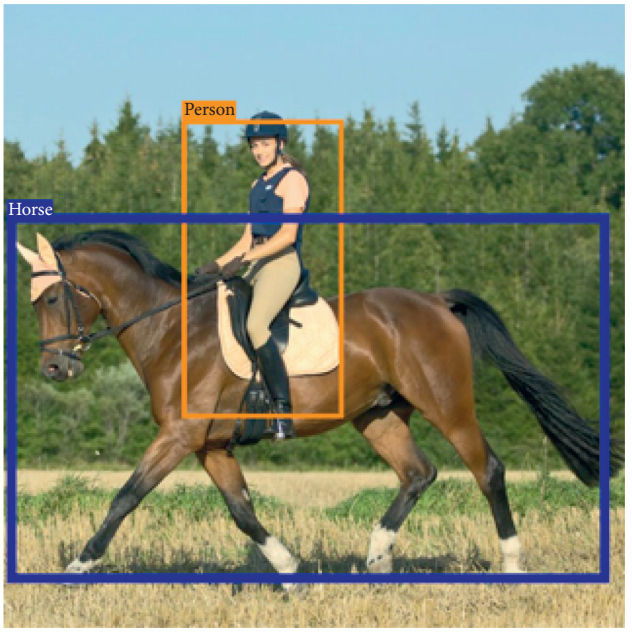


Figure 7: Testing results of our model on Pascal VOC 2007.

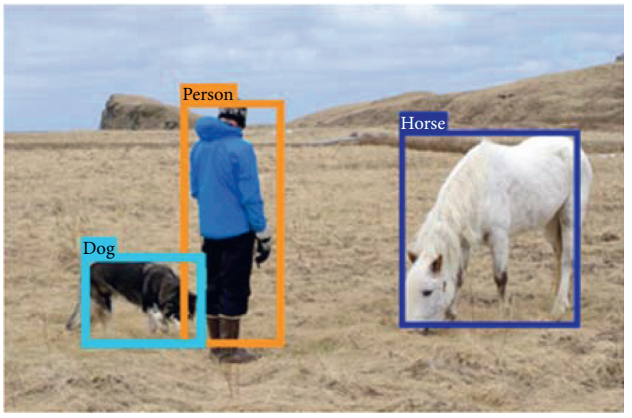


Figure 8: Testing results of our model on Pascal VOC 2012.

From the testing results, the robustness of the improved network is noticed; it classifies each class accurately and detects the desired class.

6. Conclusion

In this paper, we proposed YOLOv1 neural network based object detection by modifying loss function and adding spatial pyramid pooling layer and inception module with convolution kernels of $1 * 1$. The new network is trained on an end-to-end method, and the extensive experiment on a challenging Pascal VOC dataset, 2007/2012, shows the effectiveness of the improved new network, with the detection results being 65.6% and 58.7%, respectively. The results of the proposed network have been compared with those of R-CNN and YOLOv1, from which the effectiveness of the proposed method is demonstrated.

In the future, we expect to extend our work further to make our own benchmark dataset and a hybrid detector for small object detection.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding this paper.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2018YFC0831404 and the State Grid Corp of China Science and Technology Project "Research on Key Technologies of Knowledge Discovery Based ICT System Fault Analysis and Assisted Decision".

References

[1] A. Tiwari, A. Kumar, and G. M. Saraswat, "Feature extraction for object recognition and image classification," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, pp. 2278–0181, 2013.

[2] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2497–2504, New York, NY, USA, 2014.

[3] T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, "Fast, accurate detection of 100,000 object classes on a single machine," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1814–1821, New York, NY, USA, 2013.

[4] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[5] C.-J. Du, H.-J. He, and D.-W. Sun, "Object classification methods," in *Computer Vision Technology for Food Quality Evaluation*, pp. 87–110, Elsevier, Berlin, Germany, 2016.

[6] K. W. Eric, Li Yueping, N. Zhe, Y Juntao, L. Zuodong, and Z. Xun, "Deep fusion feature based object detection method for high resolution optical remote sensing images," *Applied Science*, vol. 34, 2019.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedigs of the International Conference on Computer Vision & Pattern Recognition (CVPR'05)*, pp. 886–893, Berlin, Germany, 2005.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, Las Vegan, NV, USA, 2016.

[10] Y. Zheng, C. Zhu, K. Luu, C. Bhagavatula, T. H. N. Le, and M. Savvides, "Towards a deep learning framework for unconstrained face detection," in *Proceedings of the 2016 IEEE 8th International Conference on BiometricsTheory, Applications and Systems (BTAS)*, IEEE, New York, NY, USA, pp. 1–8, 2016.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, New York, NY, USA, 2014.

[12] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Berlin, Germany, 2015.

[13] W. Liu, D. Anguelov, D. Erhan et al., "Single shot multibox detector," *European Conference on Computer Vision*, vol. 45, pp. 21–37, 2016.

[14] T.-Yi Lin, P. Dollár, R. B. Girshick et al., "Feature pyramid networks for object detection," *IEEE CVPR*, vol. 43, pp. 936–944, 2017.

[15] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," *Computer Vision-ECCV 2016*, vol. 43, pp. 21–37, 2016.

[16] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, C. Alexander, and C. Berg, "DSSD: Deconvolutional Single Shot Detector," *CoRR*, vol. 45, 2017.

[17] M. Lin, Q. Chen, and S. Yan, "Network in-network," 2013.

[18] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 61, pp. 91–99, 2015.

[20] Z. Zeng, J. Zhang, and X. Wang, "Place recognition an overview of vision perspective," *Applied Science*, vol. 15, 2018.

[21] X. Wan, "A comparative study of cross-lingual sentiment classification," in *Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, vol. 1, pp. 24–31, Macau, China, 2012.

[22] J. Gu and C. Lan, "Joint pedestrian and body part detection via semantic relationship learning app," *Journal of Machine Learning Research*, vol. 9, 2019.

[23] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[24] A. Humayun, F. Li, and J. M. R. I. G. O. R. Rehg, "Reusing inference in graph cuts for generating object regions," in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 336–343, Columbus, OJ, USA, June 2014.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep CNNs," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1097–1105, Lake Tahoe, NV, USA, December 2012.

[26] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," 2016.

[27] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle type classification using a semisupervised convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2247–2256, 2015.

[28] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 10, pp. 1797–1801, 2014.

[29] X. Chen, "Vehicle detection in satellite images by parallel deep convolutional neural networks," in *Proceedings of the 2013 2nd IAPR Asian Conference on Pattern Recognition (ACPR)*, vol. 45, pp. 181–185, New York, NY, USA, 2013.

[30] Y.-K. Park, J.-K. Park, H.-I. On, and D.-J. Kang, "Convolutional neural network-based system for vehicle front-side detection," *Journal of Institute of Control, Robotics and Systems*, vol. 21, no. 11, pp. 1008–1016, 2015.

[31] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," 2016.
[32] H. Wang, Y. Cai, and L. Chen, "A vehicle detection algorithm based on deep belief network," *The Scientific World Journal*, vol. 2014, 2014.
[33] K. Kim, S. Lee, J.-Y. Kim, M. Kim, and H.-J. Yoo, "A configurable heterogeneous multicore architecture with cellular neural network for realtimeobject recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 11, pp. 1612–1622, 2009.
[34] N. Sudha, A. R. Mohan, and P. K. Meher, "A self-configurable systolic architecture for face recognition system based on principal component neural network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 8, pp. 1071–1084, 2011.
[35] K. He, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 1904–1916, 2014.
[36] T.-Yi Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE ICCV*, vol. 43, pp. 2999–3007, 2017.
[37] J Liu, M. Shah, B. Kuipers, and S. Savarese, "Cross-view action recognition via view knowledge transfer," in *Proceedings of the CVPR 2011*, pp. 3209–3216, Colorado Springs, CO, USA, June 2011.
[38] L. Wu, S. C. Hoi, and N. Yu, "Semantics-preserving bag-of-words models and applications," *IEEE Transaction Image Processing*, vol. 19, pp. 1908–1920, 2010.
[39] R. Girshick, "Fasr r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Boston, MA, USA, June 2015.
[40] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," *CVPR*, vol. 45, 2006.
[41] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
[42] L. Maaten and G. Hinton, *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

Hindawi
Scientific Programming
Volume 2020, Article ID 1953461, 16 pages
https://doi.org/10.1155/2020/1953461

Research Article

High-Performance Machine Learning for Large-Scale Data Classification considering Class Imbalance

Yang Liu,[1] Xiang Li,[1] Xianbang Chen,[1] Xi Wang,[2] and Huaqiang Li[1]

[1]*College of Electrical Engineering, Sichuan University, Chengdu 610065, China*
[2]*State Grid Sichuan Economic Research Institute, Chengdu 610041, China*

Correspondence should be addressed to Huaqiang Li; lihuaqiang@scu.edu.cn

Received 23 December 2019; Revised 24 February 2020; Accepted 5 May 2020; Published 18 May 2020

Academic Editor: Rahman Ali

Currently, data classification is one of the most important ways to analysis data. However, along with the development of data collection, transmission, and storage technologies, the scale of the data has been sharply increased. Additionally, due to multiple classes and imbalanced data distribution in the dataset, the class imbalance issue is also gradually highlighted. The traditional machine learning algorithms lack of abilities for handling the aforementioned issues so that the classification efficiency and precision may be significantly impacted. Therefore, this paper presents an improved artificial neural network in enabling the high-performance classification for the imbalanced large volume data. Firstly, the Borderline-SMOTE (synthetic minority over-sampling technique) algorithm is employed to balance the training dataset, which potentially aims at improving the training of the back propagation neural network (BPNN), and then, zero-mean, batch-normalization, and rectified linear unit (ReLU) are further employed to optimize the input layer and hidden layers of BPNN. At last, the ensemble learning-based parallelization of the improved BPNN is implemented using the Hadoop framework. Positive conclusions can be summarized according to the experimental results. Benefitting from Borderline-SMOTE, the imbalanced training dataset can be balanced, which improves the training performance and the classification accuracy. The improvements for the input layer and hidden layer also enhance the training performances in terms of convergence. The parallelization and the ensemble learning techniques enable BPNN to implement the high-performance large-scale data classification. The experimental results show the effectiveness of the presented classification algorithm.

1. Introduction

Classification is one of the most effective approaches in enabling the analysis of the digital data in quite a number of academia and research fields, for example, the medical researches [1–6] and the power system researches [7–12]. Multiple applications including image processing, pattern recognition, and pattern matching benefit from the accurate and efficient classification algorithms [1–6]. Li et al. [13] implemented the medical image classification using convolutional neural network (CNN). Their classification strategy can automatically and efficiently learn the graphical characteristics of the interstitial lung disease. As a result, the strategy is able to supply accurate and efficient classification performances. Jiao et al. [14] also employed and improved CNN to classify the mental load data. The experimental results show the effectiveness of their classification algorithm. Yang and Shen [15] reviewed the load classification researches of the electrical power system. The paper pointed out that classification is an effective way of researching the complicated load behaviors, which significantly affects the power production and consumption. In order to implement the natural language processing (NLP), Zhang et al. [16] employed deep learning as the underlying methodology to execute the text classification. Their work is able to identify the semantic labels of the texts successfully.

In the recent years, a number of classification algorithm researches have been presented, which are mainly based on k-means [17], fuzzy c-means (FCM) [18], neural networks (NNs) [19], support vector machine (SVM) [20], and so on.

For example, Peng et al. [7] employed a number of clustering algorithms containing k-means, k-medoids, self-organizing maps (SOM) [21], and FCM to recognize the patterns of the electrical load data. Xu et al. [8] improved and optimized k-means algorithm using canopy algorithm. The authors claimed that the clustering accuracy and efficiency of their algorithm have been significantly enhanced. Niazmardi et al. [22] presented an improved FCM algorithm which can achieve accurate results for clustering the hyperspectral data. However, quite a number of researches [23–25] pointed out that the flaws, for example, the sensitivity of outliers, difficulty of clustering nonlinear-separable classes, and empirical values-based parameters, existing in the abovementioned unsupervised machine learning algorithms prevent them from being effectively used. As a result, the supervised machine learning algorithms, for example, the neural networks, have been widely employed in the classification researches [26]. Gu et al. [27] optimized the learning rate and inertial factor for the traditional BPNN. Their improved self-adaptive BPNN algorithm shows great performances in terms of data modeling. Li et al. [9] combined BPNN and FCM to implement the electrical load forecasting. The authors reported that the load forecasting accuracy can be significantly improved. Based on the conclusions of the researches [10,11,23–25], BPNN has been proved that it is quite suitable for the classification tasks. Although BPNN has a number of advantages, it still has flaws, for example, the slow convergence issue in its training caused by the sensitivity of initial weights, sensitivity of learning rate, gradient exploding, and gradient vanishing. However, several researches [28–33] suggested that batch-normalization [28] and ReLU [34] have great potentials to solve these issues.

Additionally, current digital data collection benefits from the developments of the smart meters, data communication systems, and data storage technologies, which results in sharply increasing of the data scale and data dimension. Tang et al. [35] pointed out that to overcome the issue, machine learning algorithms need focus on dimension reduction and sample selection techniques to improve the algorithm efficiencies. Farrell et al. [36] also proved that their improvements of the maximum entropy principle, random forest, and SVM can achieve satisfied performances for processing the high-dimensional data. Xu et al. [37] claimed that the data labeling and training efficiency are two main challenges for the large-scale data classification. Therefore, they presented a k-means and SVM-based strategy to implement the large-scale data classification. Their experimental results show that based on their strategy, the size of the training dataset can be reduced with maintaining the classification accuracy. Liu et al. [23] further pointed out that BPNN encounters extremely low efficiency when it classifies the large-scale and high-dimension data. The research also stated that the distributed computing could be a suitable solution to overcome the low-efficiency issue. Su et al. [12] employed the Hadoop framework [38] to implement the efficiency improvement for BPNN. The authors reported that their solution achieves satisfied precision for the data forecasting. Liu et al. [25] also presented a Spark-based distributed BPNN algorithm [39] which shows remarkable efficiency for classifying large-scale data. However, Liu et al. [23] pointed out that the algorithm decoupling-based parallelization of BPNN may generate a great number of iterations, which deteriorates the processing efficiency. The authors also claimed that the data separation-based parallelization may reduce the final classification accuracy. Therefore, to figure out a parallelized BPNN with high efficiency and accuracy is a valuable work.

It also should be noticed that due to multiple classes and uneven data distribution, the class imbalance issue [40] frequently exists in the training data. The issue could significantly affect the training effect, which further impacts the final classification accuracy. Zhang et al. [41] reported that the image recognition ability of the deep CNN declines in the case of unbalanced training data. Therefore, the authors presented a classification method which recognizes a query image by comparing distances between category centers of CNN features of the whole training dataset and the corresponding CNN feature of the query image. Li et al. [42] also pointed out that although CNN supplies very high performances, it is still suffering from the problem of class imbalance. By adding an extra class-imbalance aware regularization, the authors presented a new loss function enabling CNN more sensitive to the samples of the minority classes. Zhang et al. [43] also claimed that the imbalanced class issue leads to significant misclassification of deep belief network (DBN). To address the issue, the authors unequalize the misclassification costs between classes, and then the costs are applied to DBN to achieve the accurate classification. Although the efforts contributed by the abovementioned researches are able to solve the class imbalance issue, Han et al. [40] suggested that the sampling technique should be an effective way of rebuilding the samples for the minority class and further solves the class imbalance issue efficiently.

As a result, this paper presents a BPNN-based high-performance large-scale data classification method considering class imbalance. The paper firstly improves the Borderline-SMOTE [40] algorithm using Fréchet distance [44] to solve the potential class imbalance issue in the training data. Secondly, in order to solve the slow convergence issue in the training phase of BPNN, zero-mean [45], batch-normalization, and ReLU are employed to improve the input layer, hidden layer, and activation function. Thirdly, this paper also presents a MapReduce-based parallelization method for the improved BPNN. Based on the data separation and ensemble learning techniques, the parallelization of the improved BPNN can be implemented.

The rest of the paper is organized as follows: Section 2 presents the details of the methodologies for the BPNN improvements; Section 3 shows the experimental results; Section 4 concludes the paper.

2. Class Balance-Based Improved BPNN in Enabling Large-Scale Classification

This section firstly presents the Fréchet distance-based Borderline-SMOTE which is able to solve the class imbalance issue in the training dataset, and then, the section presents the details of BPNN improvement using zero-

mean, batch-normalization, and ReLU. Finally, the section presents the ensemble learning-based parallelization for the improved BPNN using the Hadoop framework.

2.1. Fréchet Distance and Borderline-SMOTE-Based Class Balance. Class imbalance issue may significantly impact the training of BPNN, which finally leads to the misclassification. Borderline-SMOTE [40] is proved to be an effective solution for balancing the classes in the training dataset. However, Borderline-SMOTE measures the similarity between data instances using Euclidean distance which lacks the ability for representing the shape features and sequential characteristics of the data instances. Therefore, this paper employs Fréchet distance [44] instead of Euclidean distance to measure the similarity between the data instances.

2.1.1. Fréchet Distance Computation for Two Data Instances. Let R^2 denote the metric space; $A, B: [0, 1] \longrightarrow R^2$ represent two continuous curves in the Fréchet space [46]; and $\alpha, \beta: [0, 1] \longrightarrow [0, 1]$ denote two continuous nondecreasing functions with independent variable t in the unit interval. Therefore, the Fréchet distance $\delta_{\mathrm{dF}}(A, B)$ is defined as the following equation:

$$\delta_{\mathrm{dF}}(A, B) = \inf_{t\in[0,1]} \max \|A(\alpha(t)) - B(\beta(t))\|_P, \quad (1)$$

where $||.||_P$ represents the Euclidean norm; inf represents the infimum of the set; $\alpha(0) = \beta(0) = 0$, $\alpha(1) = \beta(1) = 1$ [46].

The parameter t is continuous which cannot adapt to the computation for the discrete parameters. Therefore, the discrete Fréchet distance is presented by researches [47,48]. Let $P = (p_1, p_2, ..., p_u)$ and $Q = (q_1, q_2, \ldots, q_v)$ denote two discrete curves, and k denote the total number of the Fréchet permutations [47,48]; therefore, for one Fréchet permutation $W_j = \{(P_i, Q_i)\}$, $1 \le I \le u$, $1 \le j \le k$, the max value of the max distances in W_j can be represented by $d_F^{W_j}(P, Q) = \max_i \max_{(p,q)\in P_i\times Q_i} \mathrm{dis}(p, q)$. As a result, the discrete Fréchet distance for P and Q can be represented by the following equation:

$$d_F(P, Q) = \min_W d_F^W(P, Q). \quad (2)$$

Fréchet distance has better ability to represent the shape features and sequential characteristics of the data instances [44]. Therefore, it is employed by this paper to compute the nearest neighbors for the Borderline-SMOTE algorithm.

2.1.2. Borderline-SMOTE in Enabling Class Balance. Borderline-SMOTE is able to balance the classes for the imbalanced dataset [40]. It balances the dataset based on two remarkable advantages including the identification of the border between the major and minority classes, and the sample synthesis around the border.

Step 1. In one dataset T, for each point $p_i (i = 1, ..., \mathrm{pnum})$ in the minority class P, compute a set of m nearest neighbors using Fréchet distance (equation (2)). Inside the set, the number of the points belonged to the major class is m' $(0 \le m' \le m)$.

Step 2. If m' equals m which indicates the m nearest neighbors are majority examples, p_i is regarded as noise and neglected. Otherwise, if $0 \le m' \le m/2$, p_i has less chance to be misclassified, which does not need to be further processed. If $m/2 \le m' \le m$, p_i has higher chance to be misclassified. Therefore, for all p_i in the minority class, a borderline set $E = \{p_1', p_2', ..., p_{\mathrm{dnum}}'\}$, $0 \le \mathrm{dnum} \le \mathrm{pnum}$ can be achieved, where p_i' denotes a nearest neighbor of p_i.

Step 3. For each p_i' in E, compute a number of k nearest neighbors from the minority class P, and then a number of s points are randomly selected from the k neighbors. As a result, let $r_j (j = 1, 2..., s)$ denote a random value between 0 and 1, and p_j' denote one of the s neighbors. Therefore, a new instance synthetic$_j = p_i' + r_j \times (p_i' - p_j')$ can be ultimately synthesized.

A demonstration performed by Fréchet distance-based Borderline-SMOTE is shown in Figure 1. Firstly, the algorithm is able to identify the border between two imbalance classes. Secondly, new instances can be synthesized to balance the classes and further highlight the border to distinguish two classes. Therefore, if class imbalance issue exists in the training dataset of BPNN, the Fréchet distance-based Borderline-SMOTE has great potential to improve the training performance. However, it also should be noted that the values of two parameters k and m affect the performance of Borderline-SMOTE. Therefore, in the later algorithm evaluation sections, the optimal values of k and m are selected from a series of preexecuted experiments.

2.2. Improved BPNN Using Zero-Mean, Batch-Normalization, and ReLU. As an effective classification algorithm, BPNN has been successfully employed in quite a number of researches [10,11,26,49]. However, issues, for example, the sensitivity of initial weights, sensitivity of learning rate, gradient exploding, and gradient vanishing, are still needed to be handled. Therefore, this paper firstly employs zero-mean to normalize the input data instances. Secondly, batch-normalization and ReLU activation function are employed to overcome the convergence issues in the training phase.

2.2.1. Brief Introduction of the Standard BPNN. Figure 2 shows the architecture of a typical BPNN, which contains the input layer, several hidden layers, and the output layer. In the hidden and output layers, a number of neurons exist, respectively. In Figure 2, $x_1, \ldots, x_n$ represent the input of BPNN; w_{ij}, b_i, and g represent the weight, bias, and activation function of a neuron; a_i represents the output of a neuron in the hidden layer; y represents the output of BPNN ($a = y$ in the output layer). To facilitate the presentation, the aforementioned parameters are represented in the compact form of matrices X, W, b, A, and Y. In terms of the network training, two phases including feed forward and back propagation are employed. Basically, feed forward computes the input X using each layer of the network to achieve the

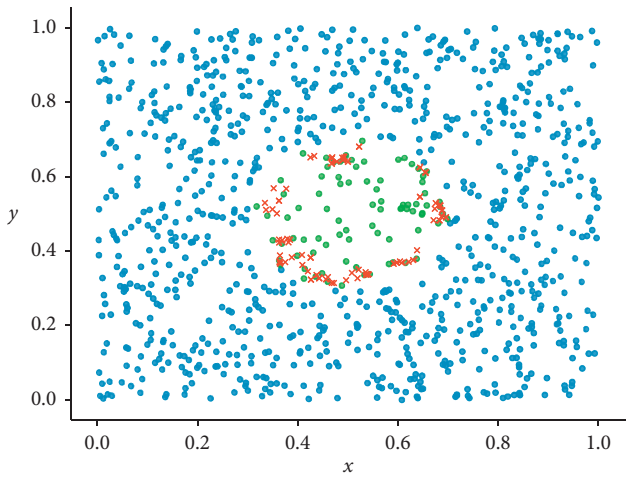


Figure 1: Borderline-SMOTE-based data instance synthesis (• majority class; • minority class; × synthetic data).

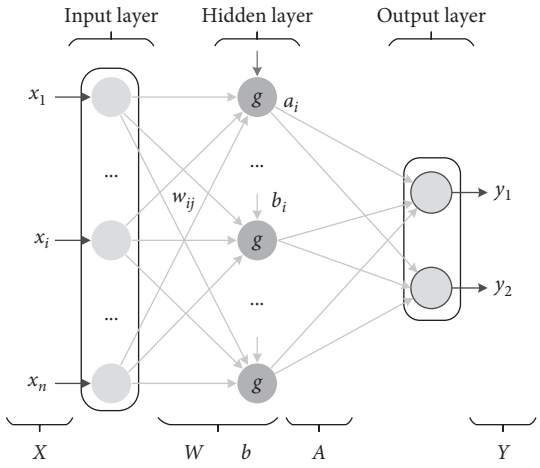


Figure 2: The architecture of a typical BPNN.

output Y according to $Y = g(W^T X + b)$. Back propagation employs the loss function *loss* to compute the variance J between the output Y and the actual value $Y\_$ according to $J = \text{loss}(Y, Y\_)$. The neural network chooses a proper optimization algorithm, for example, the stochastic gradient descent (SGD) [29] algorithm, to update W and b using $W = W - \alpha(\partial J/\partial W)$, $b = b - \alpha(\partial J/\partial b)$, where α is the learning rate [50].

In terms of classification using BPNN, firstly the training phase should be carried out. Let $\text{instance}_i = [a_1, a_2, \ldots, a_n]$ indicate the i^{th} data instance in the training dataset; a denote one feature of instance_i; and c_j denote the class that instance_i is belonged to. Firstly, each feature of instance_i is normalized. Secondly, BPNN inputs instance_i to run the feed forward to compute the output Y, and then, the coded c_j is regarded as the actual value $Y\_$ to run the back propagation to update the weights and biases. As long as all the training instances in the training dataset have been processed with several epochs and iterations, the training phase terminates. In the classification phase, let instance_k denote the k^{th} testing instance in the testing dataset. BPNN inputs instance_k and computes the output using only feed forward. The output is the class which instance_k is belonged to. As long as all the testing instances in the testing dataset have been processed, the classification phase terminates.

Figure 3 shows the BPNN improvements using zero-mean, batch-normalization, and ReLU. The details of the improvements are presented in the following sections.

2.2.2. Zero-Mean-Based Input Layer Improvement. Zero-mean is a normalization technique which is able to improve the data distribution to accelerate the gradient descent [45]. Let the input matrix X denote a number of batch-size input data instances in each iteration. Processed by the zero-mean layer shown in Figure 3, the zero-centered data instances become the actual input. Let the matrix X_{mean} denote the average value of the number of batch-size input data instances, and the matrix $X_{\text{zero-centered}}$ represent the zero-centered input; therefore, equations (3) and (4) indicate the computations for X_{mean} and $X_{\text{zero-centered}}$:

$$X_{\text{mean}} = \begin{bmatrix} \overline{x}_1 \\ \vdots \\ \overline{x}_{\text{batch-size}} \end{bmatrix}, \tag{3}$$

$$X_{\text{zero-centered}} = X - X_{\text{mean}}. \tag{4}$$

2.2.3. Batch-Normalization-Based Hidden Layer Improvement. Briefly, in the training phase of a multiple-hidden-layers neural network, the data distributions of different layers may vary, which is no longer independent and identically distributed (IID). Therefore, the internal covariate shift (ICS) occurs [28], which causes two main issues impacting the training. The first one is that the variation of the parameters in the current layer may cause the distribution variation of the input data in the last layer. As a result, the last layer has to tune itself to adapt to the distribution variation so that the learning performance is deteriorated. The second one is that the values of W and b may keep enlarging, which leads to larger value of $W \times x + b$ in each layer. Consequently, the gradient saturation of the activation function may occur. Therefore, the value of the updated gradient may be extremely small in the back propagation, which leads to the deterioration of the network convergence.

However, batch-normalization [28] employs the normalization operations to recover the input data distribution for each neuron in the hidden layers to the standard normal distribution $N(0, 1)$. Therefore, the activation function is able to work sensitively. Generally, in batch-normalization, the parameters mean μ and variance σ of the input data enable the normalization. Additionally, the tiny variations of μ and σ in each mini-batch can also improve the generalization of the neural network. The initial scale factor γ and the displacement factor β are adopted to implement the linear transformation. Both of the factors can be trained in batch-normalization. As a result, batch-normalization is capable of dealing with larger initial weights, larger learning rates,

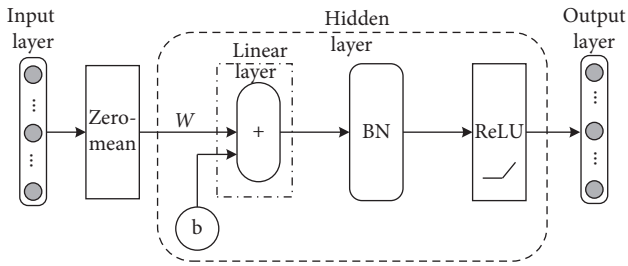


Figure 3: The improvements for the input, hidden, and output layers.

gradient issues, and overfitting issue, which significantly improves the training performance of the neural network [28].

This paper employs batch-normalization to improve the hidden layer of BPNN due to its remarkable advantages. It normalizes the linear output of the hidden layer, and then the output of batch-normalization is input into the nonlinear ReLU. The details of batch-normalization are shown in Algorithm 1.where D represents the output of the linear computation of the hidden layer; y_i denotes the output of batch-normalization, which is the input of the nonlinear ReLU; ε is a constant value to stabilize the training; μ_D and σ_x^2 represent the mean and variance of the input data instances; $\widehat{d}_i$ represents the standardized intermediate data. Similar to the weight and bias of BPNN, γ and β are also updated in the back propagation phase according to $\gamma = \gamma - \alpha(\partial L/\partial y_i)(\partial y_i/\partial \gamma)$ and $\beta = \beta - \alpha(\partial L/\partial y_i)(\partial y_i/\partial \beta)$, where L represents the back propagated loss function of the hidden layer.

2.2.4. ReLU Activation Function. To overcome the saturation of the commonly used *Sigmoid* and *Tanh* activation functions, as well as to improve the convergence of the network, this paper employs ReLU shown by equation (5) as the activation function for the neural network, which is able to handle the gradient issues [29]:

$$\mathrm{ReLU}(x) = \begin{cases} 0, & x < 0, \\ x, & x \geq 0, \end{cases} \tag{5}$$

where x presents the input of ReLU.

2.3. Ensemble Learning-Based Parallelization of BPNN. In order to implement the large-scale data classification, the Hadoop framework based on MapReduce computing model [51] is employed to parallelize the improved BPNN. This paper firstly separates the entire training dataset into a number of data chunks which are saved in HDFS (Hadoop Distributed File System), and then, each participated mapper initializes one sub-BPNN and inputs one data chunk, respectively. Each mapper trains its own classifier individually so that a number of different classifiers can be finally achieved in parallel. Therefore, for one testing data instance, these classifiers may generate different decisions. Finally, this paper also presents a weighted voting strategy to decide the ultimate classification result for the testing data instance.

2.3.1. Data Separation-Based Parallelization for the Improved BPNN. Although BPNN can be directly decoupled and parallelized in Hadoop, the research [23] suggested that the efficiency of the parallelized algorithm is extremely low due to the imperfect iteration support of the Hadoop framework. Therefore, the parallelization in this paper is based on the data separation. Firstly, the entire training dataset is separated into a number of data chunks using the random sampling. The data chunks are saved in HDFS. However, in each data chunk, the class imbalance issue may be intensified or alleviated due to the random sampling and the separation. Therefore, if the class imbalance exists in one data chunk, Borderline-SMOTE balances the classes, and then, a number of mappers start, each of which individually initializes one improved BPNN as a sub-BPNN and inputs one data chunk to train the parameters of the sub-BPNN. The workflow of each sub-BPNN in a mapper is shown in Figure 4.

Finally, as long as the training for each mapper terminates, multiple different classifiers can be achieved in parallel. In the classification phase, one testing instance is input into all the classifiers. As a result, different classifiers may generate different classification results. One reducer collects all the results from the mappers to form an aggregation, in which the weighted voting is carried out to achieve the final classification result for the testing instance.

2.3.2. Weighted Voting. In order to ensemble the multiple classification results into one final result in the reducer, this paper presents a classification reliability-based weighted voting, which can achieve the final classification result according to the reliabilities of the results from the multiple classifiers.

It is known that for different classes, each classifier has different classification accuracies. Therefore, let C represent the reliability matrix; n represent the number of classes in the training dataset; and $r = [r_1, r_2, ..., r_n]$ represent the classification accuracy for each class of the classifier. Consequently, *softmax* function [52] can be adopted to compute the reliability c_i for the i^{th} class of the classifier using the following equation:

$$c_i = \frac{e^{r_i}}{\sum_{j=1}^{n} e^{r_j}}. \tag{6}$$

Therefore, for a number of m classifiers, the reliability matrix C can be represented by the following equation:

$$C_{m\times n} = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mn} \end{bmatrix}. \tag{7}$$

The output coding of each classifier is based on one-hot encoding [44]. So, the output p of the classifier is the probability distribution of each class p_i ($i = 1, 2, \ldots, n$) indicated by the following equation:

$$p = [p_1, p_2, ..., p_n]. \tag{8}$$

Therefore, the outputs of m classifiers form a probability distribution matrix P shown by the following equation:

Input: a batch of input $D = \{d_1, d_2, ..., d_{\text{batch-size}}\}$
Initial scale and displacement factors: γ, β
Output: $\{y_i = \text{BN}_{\gamma,\beta}(d_i)\}$
$\mu_D = (1/\text{batch} - \text{size}) \sum_{i=1}^{\text{batch-size}} d_i$
$\sigma_D^2 = (1/\text{batch} - \text{size}) \sum_{i=1}^{\text{batch-size}} (d_i - \mu_D)^2$
$\widehat{d}_i = (d_i - \mu_D)/(\sqrt{\sigma_d^2 + \varepsilon})$
$y_i = \gamma \widehat{d}_i + \beta \equiv \text{BN}_{\gamma,\beta}(d_i)$,

Algorithm 1: Batch-normalization.

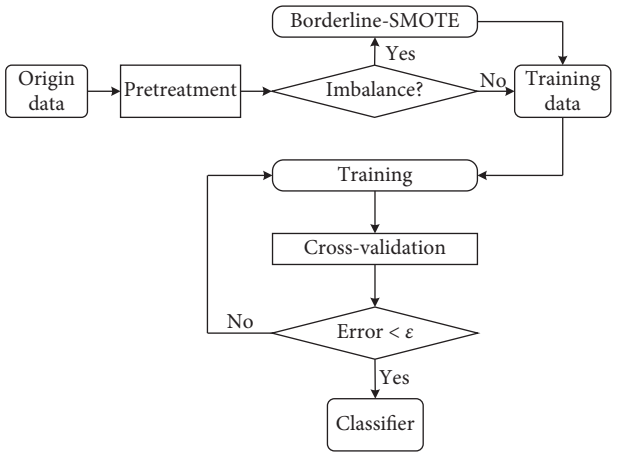


Figure 4: The data balancing and training processes for one sub-BPNN in a mapper.

$$P_{m\times n} = \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mn} \end{bmatrix}. \tag{9}$$

An intermediate matrix Q shown by equation (10) can be achieved by multiplying the reliability matrix C and the probability distribution matrix P:

$$Q_{m\times n} = \begin{bmatrix} c_{11}p_{11} & \cdots & c_{1n}p_{1n} \\ \vdots & \ddots & \vdots \\ c_{m1}p_{m1} & \cdots & c_{mn}p_{mn} \end{bmatrix}. \tag{10}$$

The sum r_i' of the elements in the i^{th} column in Q finally forms a weight matrix R denoted by the following equation:

$$R_{1\times n} = [r_1', r_2', ..., r_n']. \tag{11}$$

Ultimately, the final classification result class_label for the input data instance can be identified according to the following equation:

$$\text{class\_label} = \arg\max_i R. \tag{12}$$

Figure 5 shows the logical flow of the BPNN parallelization using Hadoop.

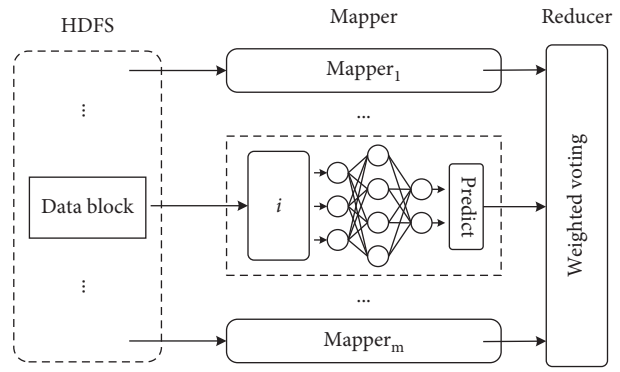


Figure 5: Parallelization of the improved BPNN using MapReduce.

3. Experimental Result

The experiments are organized into three parts. The first part evaluates the performances of the Fréchet distance-based Borderline-SMOTE using the randomly generated linearly inseparable two-dimensional semiannular dataset. The second part evaluates the performances of the improved BPNN in the standalone environment using the Iris dataset [53], Wine dataset [52], and Vehicle Silhouettes dataset [54]. The third part evaluates the performances of the parallelized improved BPNN in the Hadoop distributed computing environment. The details of the experimental environments are listed in Table 1.

3.1. Evaluation for Fréchet Distance-Based Borderline-SMOTE. A randomly generated dataset is employed to evaluate the Fréchet distance-based Borderline-SMOTE. The dataset has two classes, each of which contains 150 data instances, respectively. Based on the original dataset shown in Figure 6, an imbalanced dataset is generated, which is shown in Figure 7. Four class balance algorithms are implemented including Fréchet distance-based Borderline-SMOTE ($k = 5$, $m = 5$), random oversampling, random undersampling, and SMOTE. After the class balance for the imbalanced dataset, the improved BPNN carries out the classification using a number of 100 randomly selected training instances and a number of 90 testing instances. The classification results are shown in Table 2.

Table 2 indicates that without the class balance, the classification accuracy is only 66.67%. However, all the class balance algorithms can help to improve the classification

Table 1: Details of the experimental environment.

Standalone environment	Distributed environment
	CPU: Intel Core i5 2.5 GHz
CPU: Intel Core i5-8250U 1.6 GHz	MEM: 16 GB
MEM: 8 GB	OS: Ubuntu 16.04 64 bit
OS: Win 10 64 bit	Hadoop: 2.9.1
Python: 3.6.6	Python: 3.5.2
	Node no.: 4

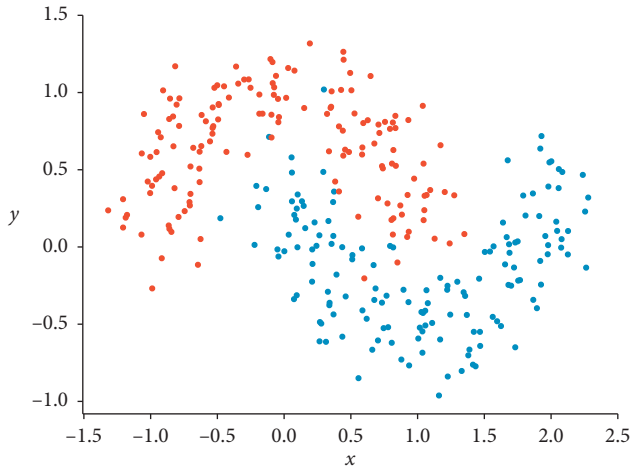


Figure 6: The randomly generated linearly inseparable two-dimensional semiannular dataset with 300 instances.

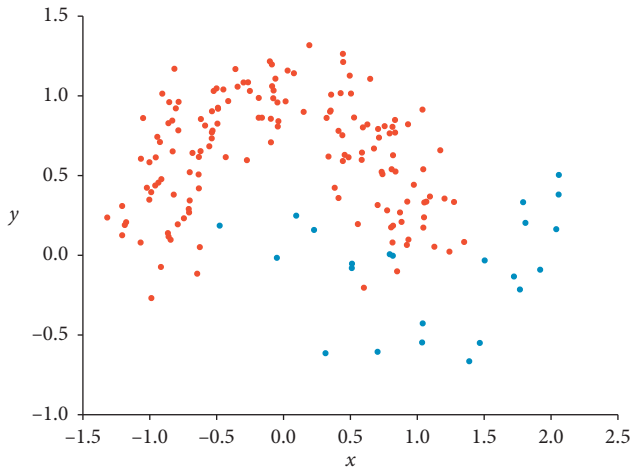


Figure 7: The imbalanced dataset based on the original dataset.

accuracy, among which the Fréchet distance-based Borderline-SMOTE significantly outperforms the other algorithms.

To further evaluate the potential of the Fréchet distance-based Borderline-SMOTE, a number of 20000 instances which averagely belonged to two classes are generated using the random sampling. Firstly, a number of 5000 instances are randomly selected from class 1, and then a number of 1000, 500, 50, and 10 instances are randomly selected from class 2, respectively, to compose 4 imbalanced training datasets. Balanced by the Fréchet distance-based Borderline-SMOTE ($k = 5$, $m = 5$), the improved BPNN is trained, and then carries out the classification. The number of the testing instances is 10000. The classification accuracies are listed in Table 3.

Table 2: The comparisons of the classification accuracy for different class balance algorithms.

Class balancing algorithm	Classification accuracy (%)
Borderline-SMOTE	95.556
Random oversampling	78.889
Random undersampling	74.444
SMOTE	86.667
Without balancing	66.667

Table 3 shows that the imbalanced training data significantly impacts the classification accuracy. However, benefitting from the Fréchet distance-based Borderline-SMOTE, the classification accuracy can be improved substantially. Table 3 also indicates that the imbalance ratio can impact the classification accuracy. The slightly imbalanced training dataset achieves better classification performance as long as it is balanced by the Fréchet distance-based Borderline-SMOTE. Contrarily, the extremely imbalanced training dataset leads to low classification accuracy even if it is balanced.

3.2. Evaluation for the Improved BPNN. This section evaluates the performances of the improved BPNN in the standalone environment. Iris dataset, Wine dataset, and Vehicle Silhouettes dataset are employed. The details of the training and testing instances are listed in Table 4. In terms of comparison, SVM, traditional BPNN, and self-adaptive BPNN [55] are also implemented.

Table 5 shows the average classification accuracy based on 50 times experiments for each algorithm using the Iris dataset. The traditional BPNN performs the lowest classification accuracy with the largest epochs. The presented improved BPNN and the self-adaptive BPNN perform the similar classification accuracy. However, in terms of the number of epochs, self-adaptive BPNN slightly outperforms the improved BPNN.

Tables 6 and 7 show the average classification accuracy based on 50 times experiments for each algorithm using the Wine dataset and Vehicle Silhouettes dataset. Firstly, the traditional BPNN also shows the worst performance. Secondly, in the Wine dataset-based experiments, although the self-adaptive BPNN slightly outperforms the improved BPNN in terms of epochs, the classification accuracy of the improved BPNN is higher than that of the self-adaptive BPNN. Thirdly, in the Vehicle Silhouettes dataset-based experiments, the improved BPNN performs the best in terms of accuracy and epochs.

Figures 8(a)–8(c) show the convergences of the traditional BPNN, the improved BPNN, and the self-adaptive BPNN using the Iris dataset, Wine dataset, and Vehicle Silhouettes dataset, respectively. For the simple Iris dataset, three algorithms perform relatively close convergences. However, for the complicated Wine dataset, both the

Table 3: The comparisons of the classification accuracy using Borderline-SMOTE.

Imbalance class ratio	Balanced dataset accuracy (%)	Imbalanced dataset accuracy (%)
5000 : 1000	98.09	73.15
5000 : 500	95.19	68.32
5000 : 50	80.53	56.20
5000 : 10	67.45	53.13

Table 4: The training instances and the testing instances of the datasets.

Dataset	Training instance number	Testing instance number
Iris	120 (class1: 40; class2: 40; class3: 40)	30 (class1: 10; class2: 10; class3: 10)
Wine	124 (class1: 36; class2: 52; class3: 36)	54 (class1: 23; class2: 19; class3: 12)
Vehicle	592 (Class1: 137; Class2: 155; Class3: 159; Class4: 141)	254 (Class1: 62 Class2: 62; Class3: 59; Class4: 71)

Table 5: The accuracy comparisons using the Iris dataset.

Algorithm	Testing dataset accuracy	Epochs
Improved BPNN	96.667	188
Self-adaptive BPNN	96.667	172
Traditional BPNN	93.333	342
SVM	96.667	/

Table 6: The accuracy comparisons using the Wine dataset.

Algorithm	Testing dataset accuracy	Epochs
Improved BPNN	94.370	723
Self-adaptive BPNN	91.444	537
Traditional BPNN	67.317	1531
SVM	96.296	/

Table 7: The accuracy comparisons using the Vehicle Silhouettes dataset.

Algorithm	Testing dataset accuracy (%)	Epochs
Improved BPNN	83.465	245
Self-adaptive BPNN	80.709	465
Traditional BPNN	74.803	993
SVM	44.488	/

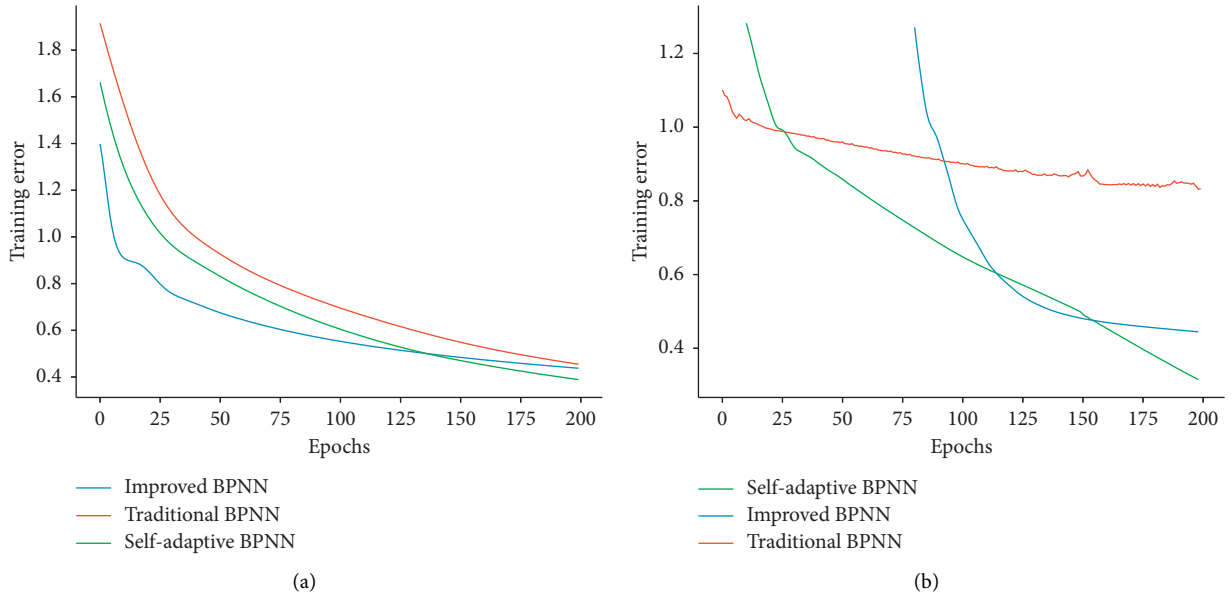
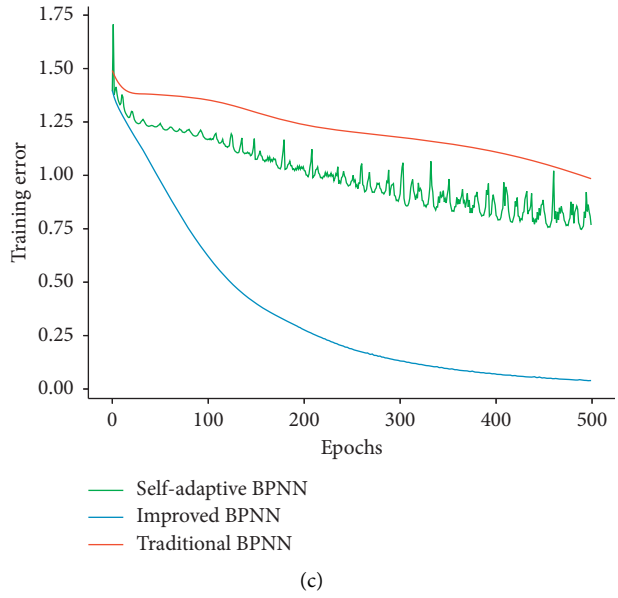


Figure 8: Continued.



(c)

Figure 8: (a) Comparison of the convergence using the Iris dataset. (b) Comparison of the convergence using the Wine dataset. (c) Comparison of the convergence using the Vehicle Silhouettes dataset.

improved BPNN and the self-adaptive BPNN significantly outperform the traditional BPNN. Until the error becomes small, the self-adaptive BPNN converges slightly faster than the improved BPNN. However, Figure 8(c) indicates that the convergence of the self-adaptive BPNN shows a number of fluctuations. The algorithm converges slowly and unstably. Contrarily, the convergence of the improved BPNN is fast and stable.

Figures 9(a)–9(c) show the max and min number of epochs in the 50 times experiments using the Iris dataset, Wine dataset, and Vehicle Silhouettes dataset, respectively. The improved BPNN outperforms the traditional BPNN and performs the similar performances to those of the self-adaptive BPNN based on Iris and Wine datasets. In terms of the Vehicle Silhouettes dataset, the improved BPNN performs the best.

Figures 10(a)–10(c) show the max and min classification accuracies in the 50 times experiments using the Iris dataset, Wine dataset, and Vehicle Silhouettes dataset, respectively. For the simple Iris dataset, the three algorithms perform closely. The presented improved BPNN slightly outperforms the other two algorithms. However, in terms of the Wine dataset, the traditional BPNN performs the worst. Although the classification accuracy of the self-adaptive BPNN can reach to 98%, its minimal accuracy is only 55%, which indicates the unstable performance of the algorithm. Contrarily, the improved BPNN performs the highest accuracy and the most stable performances. In terms of the Vehicle Silhouettes dataset, the improved BPNN also supplies the best accuracies.

To further indicate the performance of the improved BPNN, the statistical indices of evaluating the epoch and accuracy using three datasets for 50 times experiments are listed in Tables 8–10.

In the Iris dataset-based experiments, in terms of the epoch and accuracy tests, the improved BPNN and the self-adaptive BPNN perform similarly, though the improved BPNN shows slightly higher variance of the epoch, and also, both of them outperform the traditional BPNN.

In the Wine dataset-based experiments, although the traditional BPNN performs the least mean and variance of the epoch, it shows the lowest mean accuracy with the highest variance. Contrarily, the improved BPNN significantly outperforms the self-adaptive BPNN in terms of both the epoch and accuracy.

In the Vehicle Silhouettes dataset-based experiments, the traditional BPNN shows the worst performance. The improved BPNN outperforms the self-adaptive BPNN in terms of accuracy. For the epoch, the improved BPNN shows the least mean, and its variance is slightly higher than that of the self-adaptive BPNN.

Figures 11(a)–11(c) indicate that for the improved BPNN, the classification accuracy is impacted by batch size, and also, in terms of different datasets, the optimal batch size which leads to the highest accuracy is different. Therefore, a proper batch size is able to improve the classification accuracy for the presented improved BPNN.

Figures 12(a) and 12(b) indicate the numbers of epochs, nonconvergence, and overfitting of the traditional BPNN and the improved BPNN with varying learning rates (lr = 0.1, 0.01, 0.001, and 0.0001; 50 times experiments for each lr).

Firstly, Figure 12(a) shows that if the learning rate is small (lr = 0.0001), the traditional BPNN cannot converge totally, which results in the loss of the experimental result. However, Figure 12(b) shows that even if the learning rate is 0.0001, the improved BPNN can still converge. Secondly, for each learning rate, respectively, the number of exceptions (nonconvergence and overfitting) of the improved BPNN is

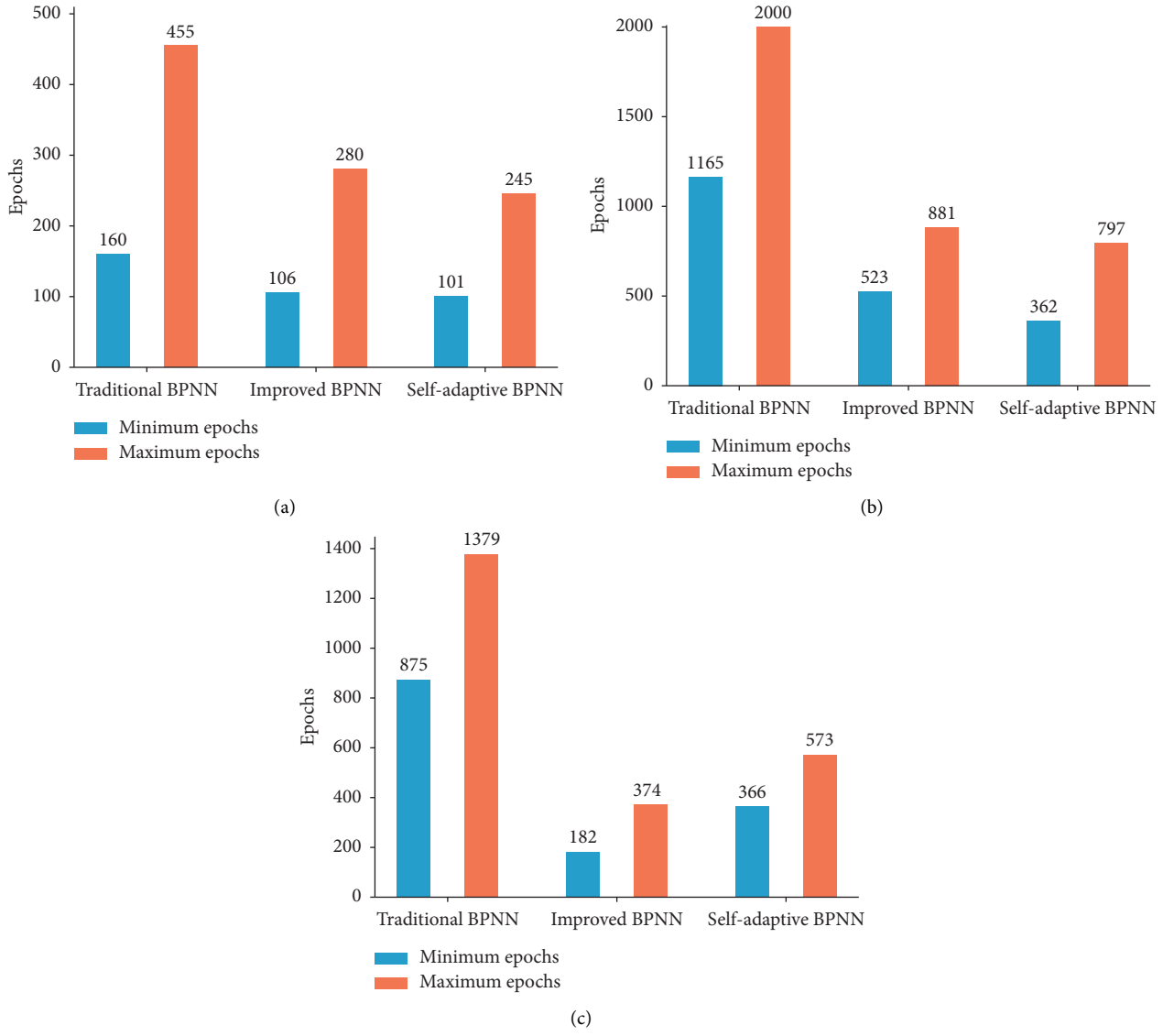


Figure 9: (a) The max and min number of epochs in 50 times experiments using the Iris dataset. (b) The max and min number of epochs in 50 times experiments using the Wine dataset. (c) The max and min number of epochs in 50 times experiments using the Vehicle Silhouettes dataset.

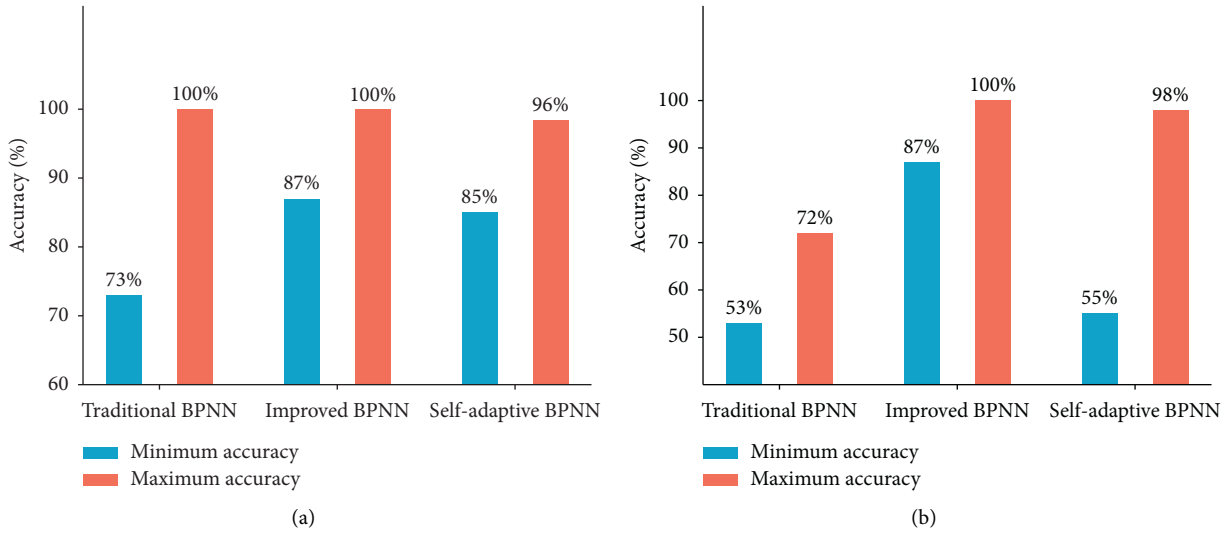
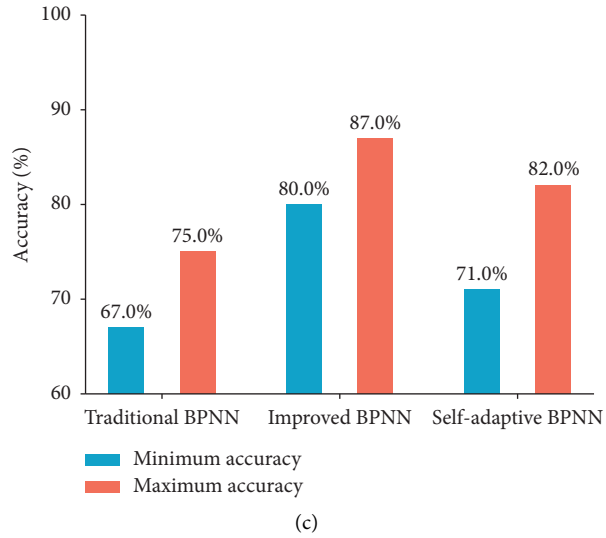


Figure 10: Continued.



(c)

Figure 10: (a) The max and min classification accuracies in 50 times experiments using the Iris dataset. (b) The max and min classification accuracies in 50 times experiments using the Wine dataset. (c) The max and min classification accuracies in 50 times experiments using the Vehicle Silhouettes dataset.

Table 8: The statistical indices of evaluating the epoch and accuracy using the Iris dataset for 50 times experiments.

Algorithm	Epoch		Accuracy	
	Mean	Variance	Mean (%)	Variance
Improved BPNN	188.82	2557.62	96.7	9.23
Traditional BPNN	342	4096.43	93.3	63.27
Self-adaptive BPNN	172.3	1093.96	96.7	11.73

Table 9: The statistical indices of evaluating the epoch and accuracy using the Wine dataset for 50 times experiments.

Algorithm	Epoch		Accuracy	
	Mean	Variance	Mean (%)	Variance
Improved BPNN	723.26	6266.69	94.4	28.31
Traditional BPNN	536.8	5974.37	67.3	372.87
Self-adaptive BPNN	1531	31924.12	91.4	55.2

Table 10: The statistical indices of evaluating the epoch and accuracy using the Vehicle Silhouettes dataset for 50 times experiments.

Algorithm	Epoch		Accuracy	
	Mean	Variance	Mean (%)	Variance
Improved BPNN	245	3243.19	83.5	21.06
Traditional BPNN	993	23542.08	74.8	187.21
Self-adaptive BPNN	465	2806.74	80.7	35.19

Table 11: The details of the training instances and the testing instances.

Dataset	Training instance number	Testing instance number
Iris	80 (class1: 36; class2: 7; class3: 37)	45 (class1: 14; class2: 18; class3: 13)
Wine	79 (class1: 36; class2: 7; class3: 36)	54 (class1: 23; class2: 19; class3: 12)
Vehicle	450 (Class1: 30; Class2: 140; Class3: 140; Class4: 140)	254 (Class1: 62; Class2: 62; Class3: 59; Class4: 71)

less than that of the traditional BPNN. Thirdly, for each learning rate, respectively, the improved BPNN performs a smaller number of epochs than that of the traditional BPNN. The experimental results prove that the improvements done by this paper can significantly improve the convergence of BPNN.

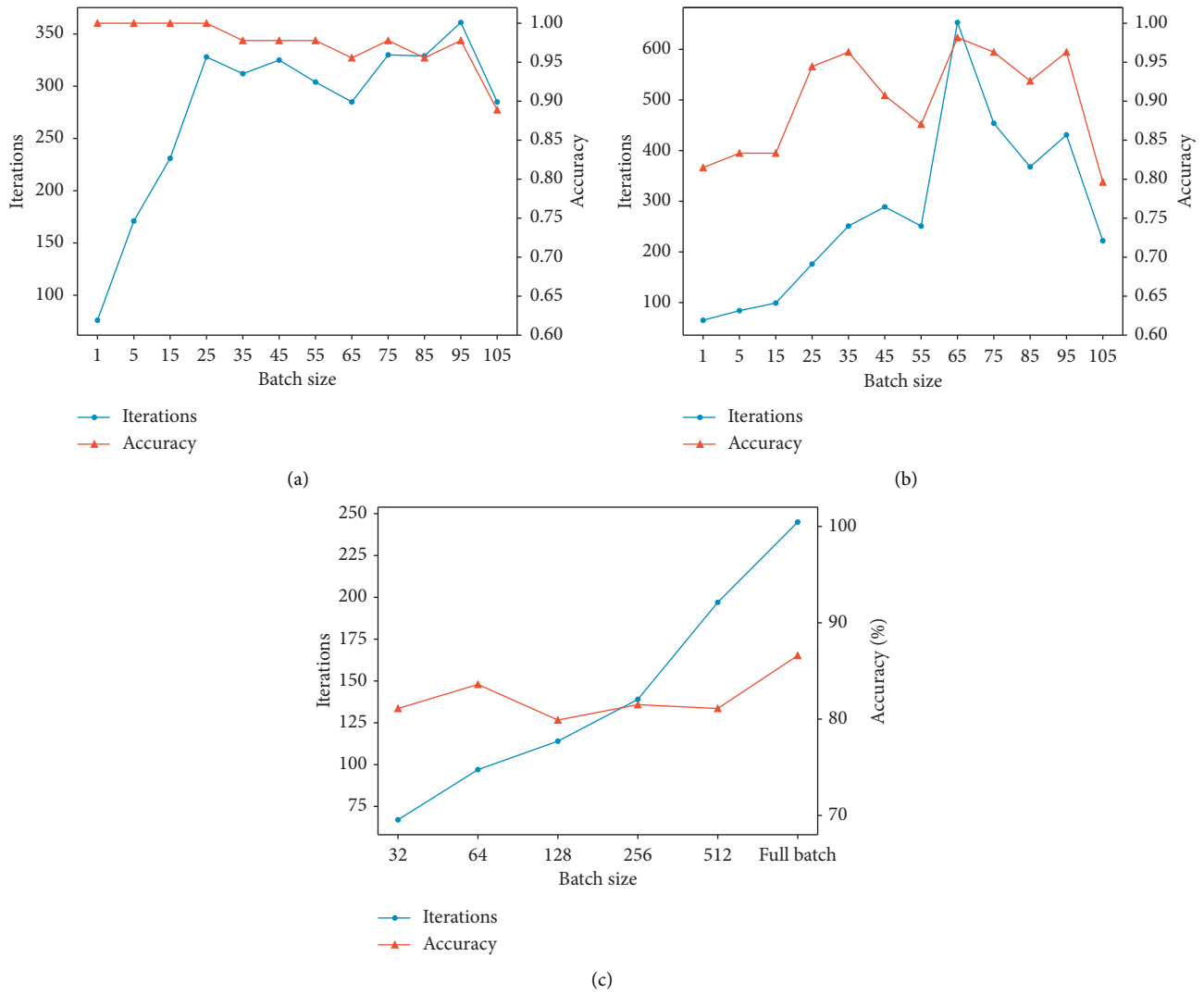


Figure 11: (a) Impact of batch sizes on classification accuracy of the improved BPNN using the Iris dataset. (b) Impact of batch sizes on classification accuracy of the improved BPNN using the Wine dataset. (c) Impact of batch sizes on classification accuracy of the improved BPNN using the Vehicle Silhouettes dataset.

In next experiments, imbalanced training datasets are generated based on the original Iris dataset, Wine dataset, and Vehicle Silhouettes dataset. The details of the training and testing instances are listed in Table 11. The imbalanced training datasets are firstly processed by the Fréchet distance-based Borderline-SMOTE ($k = 5$, $m = 10$), and then the improved BPNN carries out the classifications based on both the balanced and imbalanced training data for 50 times. The average classification accuracies are shown in Figures 13(a)–13(c).

Both Figures 13(a) and 13(b) indicate that the imbalanced training datasets significantly impact the training of the improved BPNN, which finally leads to severe misclassifications. However, balanced by the Fréchet distance-based Borderline-SMOTE, the network can be correctly and sufficiently trained. Therefore, the classification accuracies can be greatly improved. However, due to the complicated attributes of the instances of the Vehicle Silhouettes dataset, although the classification based on the balanced dataset also outperforms the imbalanced dataset, its average accuracy is only 79.53%. This point indicates that the dimension of the data may severely impact the performance of the class balance algorithm.

3.3. Evaluation for the Parallelized Improved BPNN in Hadoop Cluster. Firstly, the original Iris dataset is employed to evaluate the classification accuracy of the parallelized improved BPNN. A number of 105 instances are the training instances which will be separated for the parallelized training. The other 45 instances are the testing instances. A number of three mappers start in parallel. Each mapper initializes one sub-BPNN. The standalone BPNN and the parallelized long short-term memory network (LSTM) are also implemented in terms of comparison. Especially, the configuration of the parallelized LSTM is as the same as that of the parallelized BPNN. The number of the training

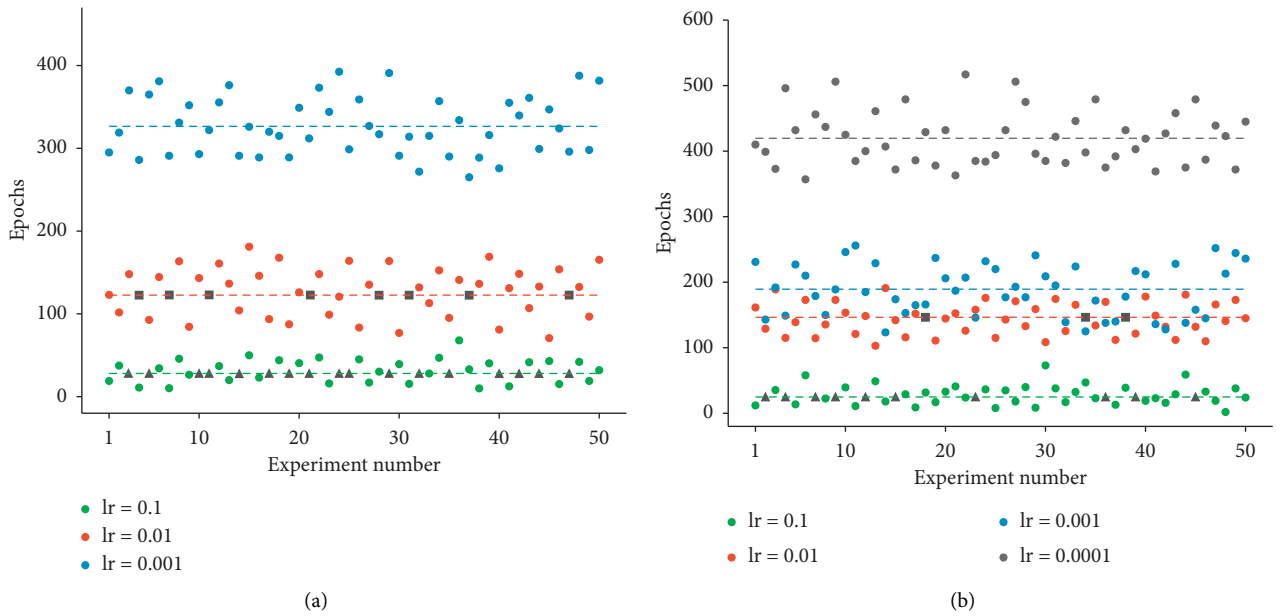


Figure 12: (a) The convergence of the traditional BPNN with various learning rates (▲: nonconvergence, ■: overfitting). (b) The convergence of the improved BPNN with various learning rates (▲: nonconvergence, ■: overfitting).

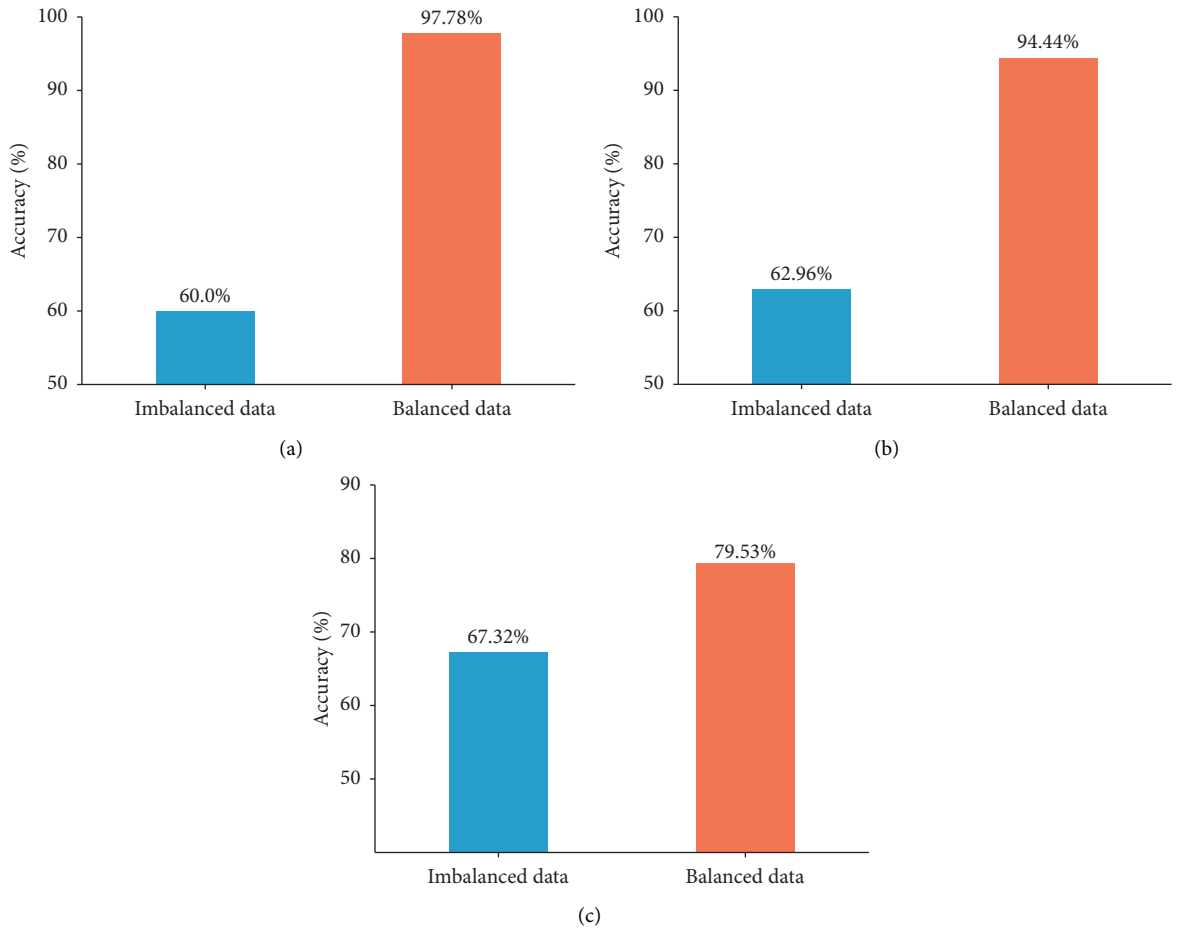


Figure 13: (a) Comparisons of the classification results using the Iris dataset. (b) Comparisons of the classification results using the Wine dataset. (c) Comparisons of the classification results using the Vehicle Silhouettes dataset.

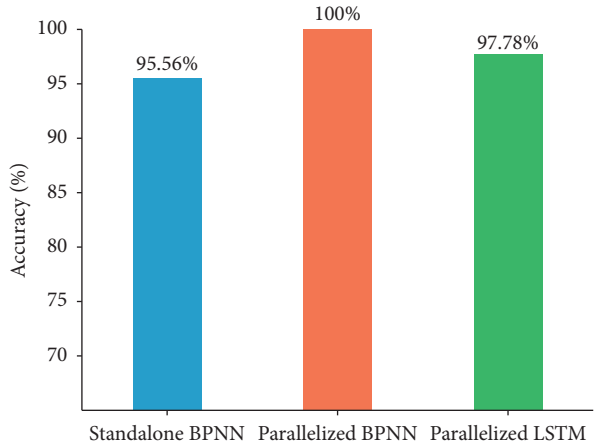


Figure 14: The accuracy comparison for the standalone BPNN, the parallelized BPNN, and the parallelized LSTM.

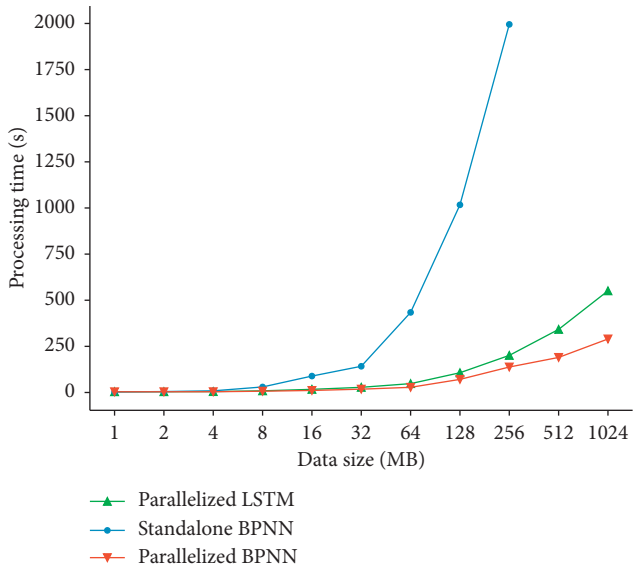


Figure 15: Efficiency comparison for the standalone BPNN, the parallelized BPNN, and the parallelized LSTM.

instances is 105 and the number of the testing instances is 45 for the standalone BPNN.

Figure 14 indicates that the parallelized BPNN outperforms the standalone BPNN and the parallelized LSTM. Benefitting from the weighted voting, the potential accuracy loss issue in the data separation is properly handled. Actually, the classification accuracies for the 3 sub-BPNNs are 96.88%, 94.44%, and 91.89%, respectively. However, the accuracy for the aggregated result can reach to 100%. It further proves that the weighted voting can help to aggregate the weak classifiers to a strong classifier. Additionally, the data separation may intensify the class imbalance issue of the training dataset in each sub-BPNN. However, the Fréchet distance-based Borderline-SMOTE can effectively solve the issue, which guarantees the classification accuracy. Moreover, the parallelized LSTM performs slightly worse than the parallelized BPNN. It is well known that LSTM is suitable for processing the time-series data. However, the Iris dataset employed for the experiments is not related to time series, which indicates that the type of the dataset may affect the performance of the parallelized LSTM in terms of accuracy.

In terms of the classification efficiency evaluation, this paper duplicates the size of the Iris dataset from 1 MB to 1024 MB. The cluster starts 16 mappers in parallel. The processing times of the standalone BPNN, the parallelized BPNN, and the parallelized LSTM are listed in Figure 15.

Figure 15 indicates that the performances of three algorithms are quite close with smaller data sizes. Due to the overhead of the Hadoop cluster, the standalone BPNN even outperforms the parallelized BPNN and the parallelized LSTM. However, along with the increasing data sizes, because of the computing resource limitations of the standalone environment, the processing time of the standalone BPNN increases sharply. Contrarily, due to less computations, the parallelized BPNN outperforms the parallelized LSTM slightly and processes the large volume of data relatively efficiently.

4. Conclusion

In order to serve the classifications for large-scale data, this paper presents a parallelized improved BPNN algorithm. The parallelization is based on the data separation, and the parallelization is implemented using the Hadoop framework. To overcome the classification accuracy loss issue caused by the separation, the weighted voting is presented to improve the classification accuracy. Based on the experimental results, the parallelization shows effectiveness for dealing with the large-scale data. However, there are other two issues existing. The first is that the class imbalance issue in the training dataset significantly impacts the training effect of BPNN, which finally leads to the deterioration of the classification accuracy. Therefore, this paper presents the Fréchet distance-based Borderline-SMOTE algorithm in enabling the class balance. According to the experimental results, the balanced training dataset can tremendously improve the classification accuracy. The second is that the convergence issue may exist in BPNN. Therefore, this paper improves the input layer, hidden layer, and activation function of BPNN by employing zero-mean, batch-normalization, and ReLU, respectively. Based on the comparisons to the traditional BPNN and the self-adaptive BPNN, the presented improved BPNN has great potential to serve the classification tasks accurately and efficiently.

Data Availability

The data of the models and algorithms used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This research was supported by the State Grid Science and Technology Project (SGAH0000TKJS1900091).

References

[1] G. Wang, Z. Deng, and K.-S. Choi, "Tackling missing data in community health studies using additive LS-SVM classifier," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 2, pp. 579–587, 2018.

[2] M. Skubic, R. D. Guevara, and M. Rantz, "Automated health alerts using in-home sensor data for embedded health assessment," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 3, pp. 1–11, 2015.

[3] L. Chen, X. Li, Q. Z. Sheng et al., "Mining health examination records—a graph-based approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2423–2437, 2016.

[4] M. Huang, H. Han, H. Wang, L. Li, Y. Zhang, and U. A. Bhatti, "A clinical decision support framework for heterogeneous data sources," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 6, pp. 1824–1833, Nov. 2018.

[5] S. Huda, J. Yearwood, H. F. Jelinek, M. M. Hassan, G. Fortino, and M. Buckland, "A hybrid feature selection with ensemble classification for imbalanced healthcare data: a case study for brain tumor diagnosis," *IEEE Access*, vol. 4, pp. 9145–9154, 2016.

[6] Y. Tseng, X. Ping, J. Liang, P. Yang, G. Huang, and F. Lai, "Multiple-time-series clinical data processing for classification with merging algorithm and statistical measures," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 3, pp. 1036–1043, 2015.

[7] X. Peng, W. Lai, and Y. Chen, "Application of clustering analysis in typical power consumption profile analysis," *Power System Protection and Control*, vol. 42, no. 19, pp. 68–72, 2014.

[8] Y. Xu, G. Li, K. Guo, S. Guo, and W. Lin, "Research on parallel clustering of power load based on improved *k*-means algorithm," *Computer Engineering and Applications*, vol. 53, pp. 260–265, 2017.

[9] Z. Li, B. Zhou, and N. Lin, "Classification of daily load characteristics curve and forecasting of short-term load based on fuzzy clustering and improved BP algorithm," *Power System Protection and Control*, vol. 40, no. 3, pp. 56–60, 2012.

[10] O. Noureldeen and I. Hamdan, "A novel controllable crowbar based on fault type protection technique for DFIG wind energy conversion system using adaptive neuro-fuzzy inference system," *Protection and Control of Modern Power Systems*, vol. 3, no. 1, pp. 328–339, 2018.

[11] A. A. Majd, H. Samet, and T. Ghanbari, "*k*-NN based fault detection and classification methods for power transmission systems," *Protection and Control of Modern Power Systems*, vol. 2, no. 2, pp. 359–369, 2017.

[12] X. Su, T. Liu, H. Cao et al., "A multiple distributed BP neural networks approach for short-term load forecasting based on Hadoop framework," *Proceedings of the CSEE*, vol. 37, pp. 4966–4972, 2017.

[13] Q. Li, W. Cai, X. Wang, Y. Zhou, D. Feng, and M. Chen, "Medical image classification with convolutional neural network," in *Proceedings of the 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pp. 844–848, Singapore, December 2014.

[14] Z. Jiao, X. Gao, Y. Wang, J. Li, and H. Xu, "Deep convolutional neural networks for mental load classification based on EEG data," *Pattern Recognition*, vol. 76, pp. 582–595, 2018.

[15] S. Yang and C. Shen, "A review of electric load classification in smart grid environment," *Renewable and Sustainable Energy Reviews*, vol. 24, pp. 103–106, 2013.

[16] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," 2015, https://arxiv.org/abs/1509.01626.

[17] C. Deng, Y. Liu, L. Xu et al., "A MapReduce-based parallel *K*-means clustering for large-scale CIM data verification," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 11, pp. 3096–3114, 2016.

[18] E.-H. Kim, S.-K. Oh, and W. Pedrycz, "Design of reinforced interval type-2 fuzzy C-means-based fuzzy classifier," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 5, pp. 3054–3068, 2018.

[19] X. Wang, Y. Guo, Y. Wang, and J. Yu, "Automatic breast tumor detection in ABVS images based on convolutional neural network and superpixel patterns," *Neural Computing and Applications*, vol. 31, no. 4, pp. 1069–1081, 2019.

[20] Q. Jiang and F. Chang, "A novel rolling-element bearing faults classification method combines lower-order moment spectra and support vector machine," *Journal of Mechanical Science and Technology*, vol. 33, no. 4, pp. 1535–1543, 2019.

[21] T. Kohonen, "Physiological interpretation of SOM," in *Self-Organizing Maps*, vol. 30, Springer, Berlin, Germany, 1997.

[22] S. Niazmardi, S. Homayouni, and A. Safari, "An improved FCM algorithm based on the SVDD for unsupervised hyperspectral data classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 2, pp. 831–839, 2013.

[23] Y. Liu, J. Yang, Y. Huang, L. Xu, S. Li, and M. Qi, "MapReduce based parallel neural networks in enabling large scale machine learning," *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 297672, 13 pages, 2015.

[24] Y. Liu, W. Jing, and L. Xu, "Parallelizing backpropagation neural network using MapReduce and cascading model," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 2842780, 11 pages, 2016.

[25] Y. Liu, L. Xu, and M. Li, "The parallelization of back propagation neural network in MapReduce and spark," *International Journal of Parallel Programming*, vol. 45, no. 4, pp. 760–779, 2017.

[26] N. Almaadeed, A. Aggoun, and A. Amira, "Speaker identification using multimodal neural networks and wavelet analysis," *IET Biometrics*, vol. 4, no. 1, pp. 18–28, 2015.

[27] D. Gu, Q. Ai, C. Chen, and S. Shen, "Application of adaptive neural network in dynamic load modeling," *Proceedings of the CSEE*, vol. 16, pp. 31–36, 2007.

[28] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448–456, Lille, France, July 2015.

[29] G. Wang, G. B. Giannakis, and J. Chen, "Learning ReLU networks on linearly separable data: algorithm, optimality, and generalization," *IEEE Transactions on Signal Processing*, vol. 67, no. 9, 2019.

[30] V. Thakkar, S. Tewary, and C. Chakraborty, "Batch normalization in convolutional neural networks-a comparative study with CIFAR-10 data," in *Proceedings of the 2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, Kolkata, India, January 2018.

[31] A. N. Abbasi and M. He, "Convolutional neural network with PCA and batch normalization for hyperspectral image classification," in *Proceedings of the 2019 IEEE International Geoscience and Remote Sensing Symposium IGARSS-2019*, Yokohama, Japan, July–August 2019.

[32] S. Wu, G. Li, L. Deng et al., "L1-Norm batch normalization for efficient training of deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 7, pp. 2043–2051, 2018.

[33] K. Tachibana and K. Otsuka, "Wind prediction performance of complex neural network with ReLU activation function," in *Proceedings of the 2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, Nara, Japan, September 2018.

[34] K. Hara, D. Saito, and H. Shouno, "Analysis of function of rectified linear unit used in deep learning," in *Proceedings of the 2015 International Joint Conference on Neural Networks*, pp. 1–8, IEEE, Killarney, Ireland, July 2015.

[35] T. Tang, S. Chen, M. Zhao, W. Huang, and J. Luo, "Very large-scale data classification based on *K*-means clustering and multi-kernel SVM," *Soft Computing*, vol. 23, no. 11, pp. 3793–3801, 2019.

[36] A. Farrell, G. Wang, S. A. Rush et al., "Machine learning of large-scale spatial distributions of wild turkeys with high-dimensional environmental data," *Ecology and Evolution*, vol. 9, no. 10, pp. 5938–5949, 2019.

[37] X. Xu, T. Liang, J. Zhu, D. Zheng, and T. Sun, "Review of classical dimensionality reduction and sample selection methods for large-scale data processing," *Neurocomputing*, vol. 328, pp. 5–15, 2019.

[38] Apache Hadoop, 2019, http://hadoop.apache.org/.

[39] Apache Spark, 2019, http://spark.apache.org/.

[40] H. Han, W. Wang, and B. Mao, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning," in *Advances in Intelligent Computing*, pp. 878–887, Springer, Berlin, Germany, 2005.

[41] Y. Zhang, L. Shuai, Y. Ren, and H. Chen, "Image classification with category centers in class imbalance situation," in *Proceedings of the 2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Nanjing, China, May 2018.

[42] F. Li, S. Li, C. Zhu, X. Lan, and H. Chang, "Class-imbalance aware CNN extension for high resolution aerial image based vehicle localization and categorization," in *Proceedings of the 2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, Chengdu, China, June 2017.

[43] C. Zhang, K. Tan, and R. Ren, "Training cost-sensitive deep belief networks on imbalance data problems," in *Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, Canada, July 2016.

[44] B. Aronov, S. Har-Peled, C. Knauer, Y. Wang, and C. Wenk, "Fréchet distance for curves, revisited," in *Lecture Notes in Computer Science*, pp. 52–63, Springer, Berlin, Germany, 2006.

[45] W. D. Mulder, G. Molenberghs, and G. Verbeke, "An interpretation of radial basis function networks as zero-mean Gaussian process emulators in cluster space," *Journal of Computational and Applied Mathematics*, vol. 363, pp. 249–255, 2020.

[46] D. Jiao, H. Wang, J. Zhu, Z. Chi, and S. Zeng, "EV battery SOH diagnosis method based on discrete Fréchet distance," *Power System Protection and Control*, vol. 44, no. 12, pp. 68–74, 2016.

[47] J. Zhu, Z. Huang, and X. Peng, "Curve similarity judgment based on the discrete Fréchet distance," *Journal of Wuhan University of Technology*, vol. 55, no. 2, pp. 227–232, 2009.

[48] C. Chen, G. Huang, Y. Fan, J. Wu, and X. Wang, "Short-term load forecasting based on discrete Fréchet distance and LS-SVM," *Power System Protection and Control*, vol. 42, no. 5, pp. 142–147, 2014.

[49] D. Fan, J. Yang, J. Zhang et al., "Effectively measuring respiratory flow with portable pressure data using back propagation neural network," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 6, Article ID 1600112, pp. 1–12, 2018.

[50] Y. Liu, X. Chen, L. Xu, H. Li, and M. Li, "A resource aware parallelized back propagation neural network in enabling efficient large-scale digital health data processing," *IEEE Access*, vol. 7, 2019.

[51] J. Dean and S. Ghemawat, "MapReduce," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[52] Wine Dataset, 21019, http://archive.ics.uci.edu/ml/datasets/Wine.

[53] Iris Dataset, 2019, https://archive.ics.uci.edu/ml/datasets/Iris.

[54] Vehicle Silhouettes Dataset, 2020, https://archive.ics.uci.edu/ml/datasets/Statlog+(Vehicle+Silhouettes).

[55] X. Yang, X. Y. Jin, and J. F. Shen, "A PVC identification method of ECG signal based on improved BPNN," *Applied Mechanics and Materials*, vol. 738-739, pp. 578–581, 2015.

Hindawi
Scientific Programming
Volume 2020, Article ID 9758049, 10 pages
https://doi.org/10.1155/2020/9758049

Research Article

A Digital Camera-Based Rotation-Invariant Fingerprint Verification Method

Sajid Khan,[1] Dong-Ho Lee,[2] Asif Khan,[3] Ahmad Waqas,[4] Abdul Rehman Gilal,[4] and Zahid Hussain Khand[4]

[1]*Center of Excellence for Robotics, Artificial Intelligence and Blockchain, Sukkur IBA University, Sukkur, Pakistan*
[2]*School of Electrical Engineering, Hanyang University ERICA Campus, Ansan, Republic of Korea*
[3]*Department of Electrical Engineering, Sukkur IBA University, Sukkur, Pakistan*
[4]*Department of Computer Science, Sukkur IBA University, Sukkur, Pakistan*

Correspondence should be addressed to Dong-Ho Lee; dhlee77@hanyang.ac.kr

Received 3 January 2020; Accepted 29 April 2020; Published 15 May 2020

Academic Editor: Asad M. Khatak

Fingerprint registration and verification is an active area of research in the field of image processing. Usually, fingerprints are obtained from sensors; however, there is recent interest in using images of fingers obtained from digital cameras instead of scanners. An unaddressed issue in the processing of fingerprints extracted from digital images is the angle of the finger during image capture. To match a fingerprint with 100% accuracy, the angles of the matching features should be similar. This paper proposes a rotation and scale-invariant decision-making method for the intelligent registration and recognition of fingerprints. A digital image of a finger is taken as the input and compared with a reference image for derotation. Derotation is performed by applying binary segmentation on both images, followed by the application of speeded up robust feature (SURF) extraction and then feature matching. Potential inliers are extracted from matched features by applying the M-estimator. Matched inlier points are used to form a homography matrix, the difference in the rotation angles of the finger in both the input and reference images is calculated, and finally, derotation is performed. Input fingerprint features are extracted and compared or stored based on the decision support system required for the situation.

1. Introduction

Image processing has many applications [1–3], including biometric registration and verification. Biometric registration and verification are performed to restrict unauthorized people from entering a specific building or wing, to keep track of users of service by registering their fingerprints in a database, to take automatic attendance of employees, and to store data along with biometric features in a database for issuing passports or national identity cards.

Biometric verification and registration using image processing techniques such as face detection, fingerprint detection, and retina or iris detection have been widely researched [4–8]. Face detection-based techniques [9, 10] store facial features to be compared during the registration and verification process. Face verification and registration provides accuracy at the cost of high complexity and extensive allocation of system memory. Accuracy demands that issues such as scale difference, posture, illumination, and the opening and closing of the eyes and mouth be addressed, further increasing complexity.

Iris-based recognition and verification [7, 11, 12], on the other hand, is done by applying segmentation in the region of the iris to extract its radius. This is followed by feature extraction. These techniques are used along with fingerprint verification at airports and other high-risk buildings to provide accurate identification. Iris-based recognition and verification provide good accuracy, but the required equipment is costly.

Fingerprint verification and registration is a widely researched area [13–23] in which work has focused on algorithms and techniques to store and compare fingerprint

features obtained using a scanner. Fingerprint-based biometric verification and registration are widely used in banks [19, 20] and national databases for maintaining passport and national identity card information [24].

By using different approaches, such as combining fingerprint registration with odor analysis [21] and applying a ridgelet transform [22], researchers have improved existing algorithms. Complexity can be reduced by comparing a small number of features instead of entire images. However, these methods require the use of sensors (fingerprint readers) that are not always available. This can make performing the registration and verification process difficult.

Solutions that avoid the need of sensors have been proposed [24–27]. These techniques focus on processing images captured using ordinary digital cameras. Hiew et al. [24–26] applied skin color-based segmentation followed by morphological processing to extract regions of interest. Short-time Fourier transform analysis (STFT) is then applied, followed by ridge orientation calculation and core point detection. The detection and matching of features such as bifurcations and ridges are not discussed. However, detection and matching can be performed using the same fingerprint techniques used in the algorithms applied to input obtained from sensors.

Sankaran et al. [27] proposed a method similar to that discussed in this paper that performs skin color segmentation followed by the region of interest (ROI) extraction. Image enhancement is applied followed by ScatNet-based feature extraction. Finally, feature-based fingerprint registration/matching is performed.

When features are extracted from a digital image of a fingerprint, scanners are not needed. However, orientation alignment is an issue that is inadequately addressed in the algorithms proposed by Hiew et al. and Sankaran et al. The fingerprint in the input image must be vertically aligned when conventional methods are used. When rotation of the finger is present, a fingerprint mismatch will occur even if features of that person are present in the database.

Therefore, this paper proposes a rotation and scale-invariant technique that takes an image captured using an ordinary digital camera as the input and applies skin color-based segmentation to the input image and a reference image to extract regions of interest from both images. SURF [28] algorithm is applied to the segmented binary images, and the detected features from both images are matched. Inliers are extracted from the matched features using the M-estimator [29] for accurate calculation of the rotation angle using a homography matrix [30]. The input image is derotated to obtain an image with a vertically aligned finger. The vertically aligned input image is then subjected to image enhancement techniques, such as unsharp masking and contextual blockwise Fourier domain filtering. Otsu thresholding is applied for binarization. Then, nonmaximal suppression is used to obtain ridges having a thickness of one pixel. Features such as bifurcations and ridge endings are extracted from the thin ridges to be stored in a database for registration or compared with the features of registered fingerprints for verification. Simulation results show that the proposed method provides an accuracy of 100 percent if the suggested parameter range is applied.

This paper is organized as follows: Section 2 discusses the proposed methodology, Section 3 contains the results and discussion, Section 4 presents the conclusions and the direction of future work.

2. Methodology

The objective of this work is to propose a cost-effective rotation and scale-invariant alternative to the processing of fingerprint images obtained using scanners. A flowchart of the algorithm is shown in Figure 1. The major steps of the proposed algorithm are the derotation of the input image, image enhancement, binarization, thinning, minutiae extraction, and matching. The steps shown in the flowchart are explained in detail in this section.

2.1. Image Acquisition. In this study, image acquisition was performed using several smartphones: Samsung Galaxy S2, iPhone 3s, Galaxy S6, and iPhone 8. Images of fingers were captured with the focus on their ridges.

2.2. Image Derotation. A flowchart depicting the derotation process is shown in Figure 2. An image containing a finger aligned along the y-axis is selected as a reference image, as seen in Figure 3(a). Any input finger image requiring registration or verification is considered an input image and is subjected to the derotation process so that the fingerprint registration/matching process can be done smoothly. Figure 3(b) shows an example of an input image that requires rotational alignment for accurate feature matching.

2.2.1. Skin Color-Based Segmentation. Both the input and reference images are subjected to adaptive skin color-based image segmentation [31]. The images shown in Figures 3(c) and 3(d) show the results of this segmentation on the reference and input images. The input image shown in Figure 3(b) is particularly complex because the color of the background is similar to the color of the skin. Therefore, noise is detected as part of the finger. The white region in Figure 3(d) is the region of interest (ROI) and will be used in upcoming sections.

2.2.2. Input Image Scale Adjustment. Both input and reference images usually are of different dimensions. Therefore, the fingers in both images might have a huge scale difference. In that case, scale variance might produce problems while detecting and comparing features. In order to provide a scale-invariant derotation, scales of both input and reference fingers should nearly be the same. If the number of pixels belongs to the finger in the segmented reference image, it is denoted by N_{ref} and if the number of pixels belongs to ROI, it is denoted by N_{input}. Scaling factor is calculated using (1) as

$$S_{\text{ref}}^{\text{input}} = \frac{N_{\text{ref}}}{N_{\text{input}}}. \quad (1)$$

If $S_{\text{ref}}^{\text{input}} < 0.9$, both the intensity and segmented input images are scaled down by a factor of $S_{\text{ref}}^{\text{input}}$, whereas if

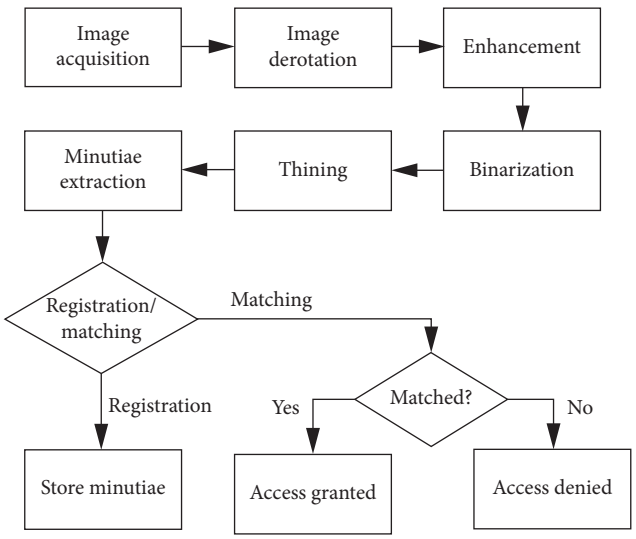


Figure 1: Flowchart of the proposed algorithm.

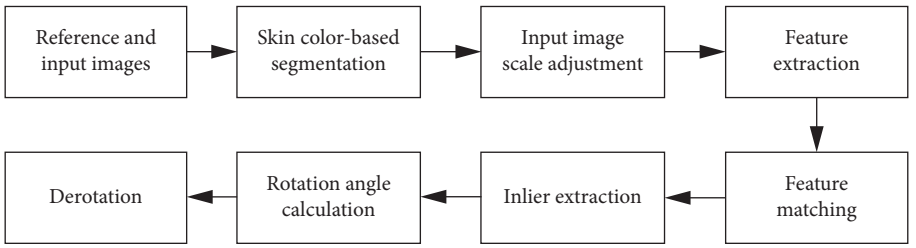


Figure 2: Flowchart of the derotation process.

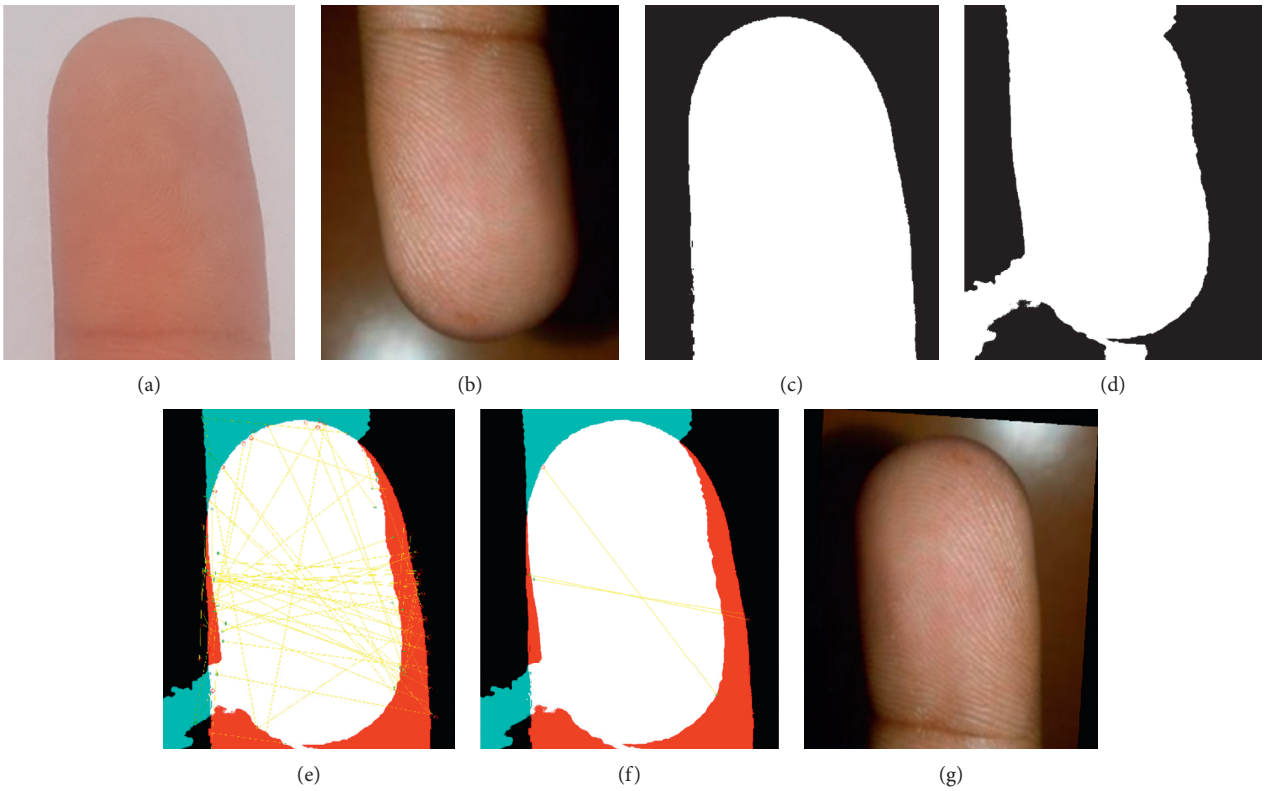


Figure 3: Stages of rotation alignment: (a) reference intensity image, (b) input intensity image, (c) segmented reference image, (d) segmented input image, (e) matched features including outliers, (f) matched inliers, and (g) derotated input image.

$S_{\text{ref}}^{\text{input}} > 1.1$, the images are interpolated by a factor of $S_{\text{ref}}^{\text{input}}$ using simple bilinear interpolation.

2.2.3. Feature Extraction and Matching. Intensity images are converted into binary images for time-efficient feature matching. Since finger shapes are all similar, features detected along their boundaries can provide better estimations of geometrical differences. For feature detection, the SURF [28] method is applied to the segmented binary images to obtain boundary features for both images. SURF is selected for its robustness and reasonable match rate for two same images with different angle of rotation [29]. After obtaining features, nearest neighbor-based feature matching by computing the pairwise distance between features is performed. Then, feature matching provides matched features that contain truly matched inliers as well as erroneously matched outliers. Figure 3(e) shows all the matched features including both inliers and outliers.

2.2.4. Inlier Extraction and Derotation of Image. To extract inliers from the matched points, an M-estimator [29] algorithm is applied. Figure 3(f) shows the inliers extracted from all matched features using the M-estimator. It can be seen that extracted inliers represent true matches that can be used to estimate the angle of rotation between accurately matched inlier points.

Once several potential inliers are extracted, the angle of rotation between the matched inliers must be determined. A homography matrix [30] is calculated for this purpose. For the two sets of matched points, $\mathbf{D} = (x_i, y_i)$ for the reference image and $D' = (x_i', y_i')$ for the input image. Then, the relation between them is as given in the following equation:

$$\begin{aligned} D' &= HD, \\ \text{or } H &= D'D^{-1}, \end{aligned} \tag{2}$$

where

$$\mathbf{D} = \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix}, \tag{3}$$

The resultant matrix in (3) is of the form given in (4) as

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} \varepsilon\cos(\theta) & -\sin(\theta) & t_x \\ \varepsilon\sin(\theta) & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

Therefore, the angle of rotation can be calculated using $h_1 2$ or $h_2 2$ using (5) as

$$\begin{aligned} \theta &= \sin^{-1}(-h_{12}), \\ \theta &= \cos^{-1}(-h_{22}), \end{aligned} \tag{5}$$

where θ can be used to derotate the input image to obtain an image with a vertically aligned finger. Figure 3(g) shows the derotation of the input image after the calculation of $\theta = -176.29^{\circ}$. Figure 4 shows the alignment method applied to various input images showing fingers at different angles of rotation. It can be seen that the proposed algorithm predicted the angles of rotation accurately and that the fingers in the derotated images are aligned vertically irrespective of their initial alignment.

2.3. Image Enhancement. Edge-based image enhancement is applied to the grayscale image to improve the visibility of ridges on the finger. Edge-based enhancement is performed in two stages. First, unsharp masking [32] is applied to the image. This algorithm subtracts an image from a blurred version of itself and enhances those regions which provide a good response after subtraction. To calculate the blurred image, a Gaussian filter is convolved with the original grayscale image using the following equation:

$$I_{\text{gray}} * G_{\sigma}, \tag{6}$$

where

$$G_{\sigma} = \frac{1}{\sqrt{\pi\sigma}} e^{-x^2/2\sigma^2}. \tag{7}$$

For the second stage of image enhancement, a contextual blockwise Fourier domain filter [33] is applied to the grayscale image by dividing it into a 16×16 overlapping block. These blocks are then filtered in the Fourier domain using a frequency and orientation-selective filter with parameters based on the estimated local ridge orientation and frequency. The outcome of the Fourier domain-based enhancement is another image denoted as I_{enhanced}. I_{enhanced} is multiplied with a skin-segmented binary image so that only the region of interest (ROI) is processed further.

Figure 5 shows the effects of this method on the derotated input image. The image in Figure 5(a) shows ridges that are not visible. The results of the unsharp masking can be seen in Figure 5(b) showing improved visibility of the ridges. Figure 5(c) shows the image once it has been converted to grayscale. Figure 5(d) shows that application of Fourier-based image enhancement further sharpens unclear edges and other details. Figure 5(e) shows the extracted ROI.

2.4. Image Binarization and Edge Thinning. Features such as ridge endings and bifurcations can be accurately extracted when ridges having a thickness of one pixel are detected. To detect thin ridges, all ridges are first detected by applying Otsu thresholding [34] to the enhanced region of interest. This will result in a binary image with ridge pixels represented in white and the remaining pixels in black. However, this will result in the detection of thick ridges. To limit the results to thin ridges, nonmaximal suppression [35] is applied. This method sets all the ridge pixels in a certain direction to zero if they do not have a maximum gradient value and retains ridge pixels with the maximal value. This results in conversion of the thick ridges to ridges with a thickness of only one pixel. Figure 6(a) shows Otsu thresholding being applied to the enhanced region of interest. The detected ridges are thick and thus cannot be used for feature detection. Application of nonmaximal

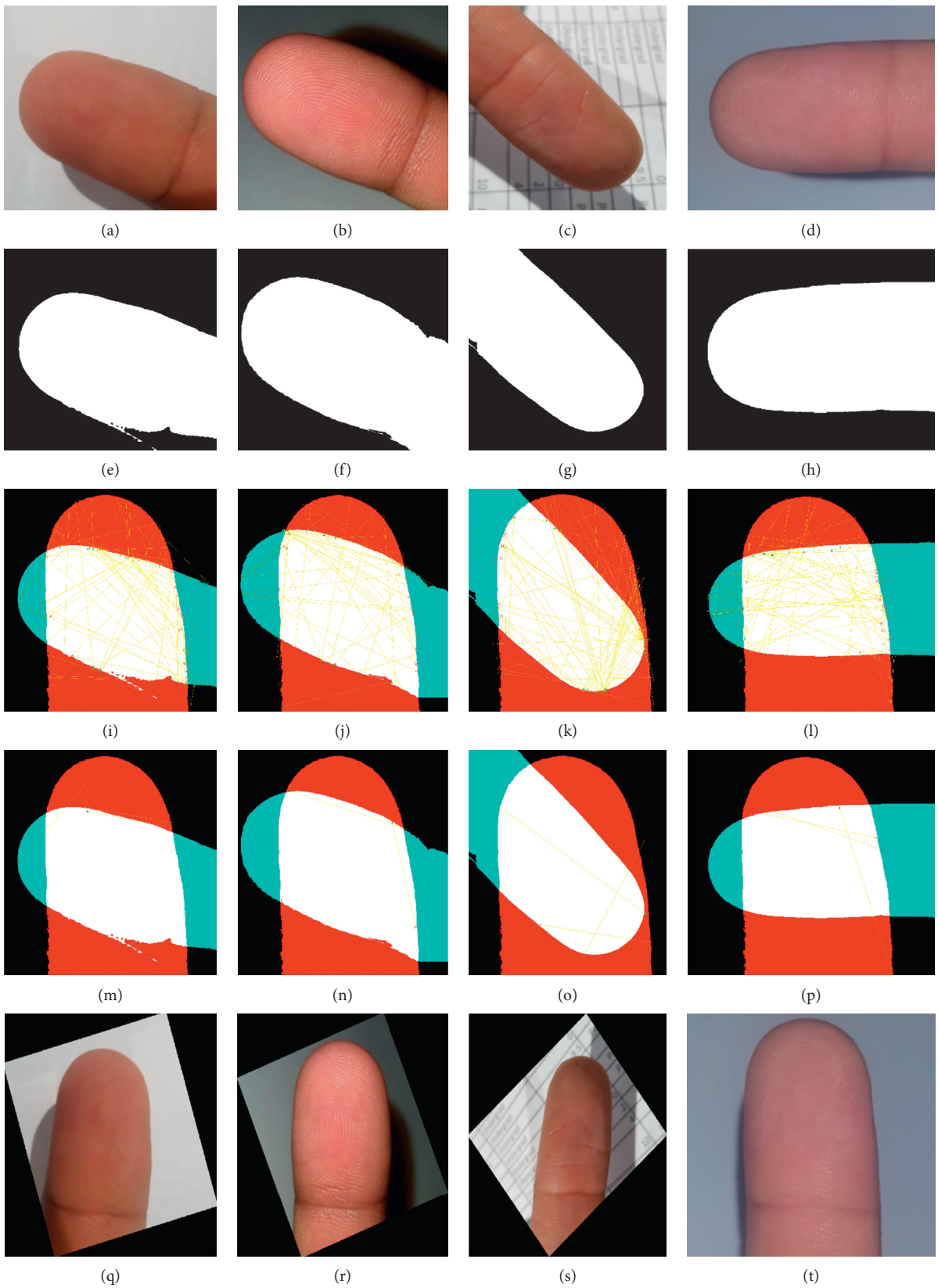
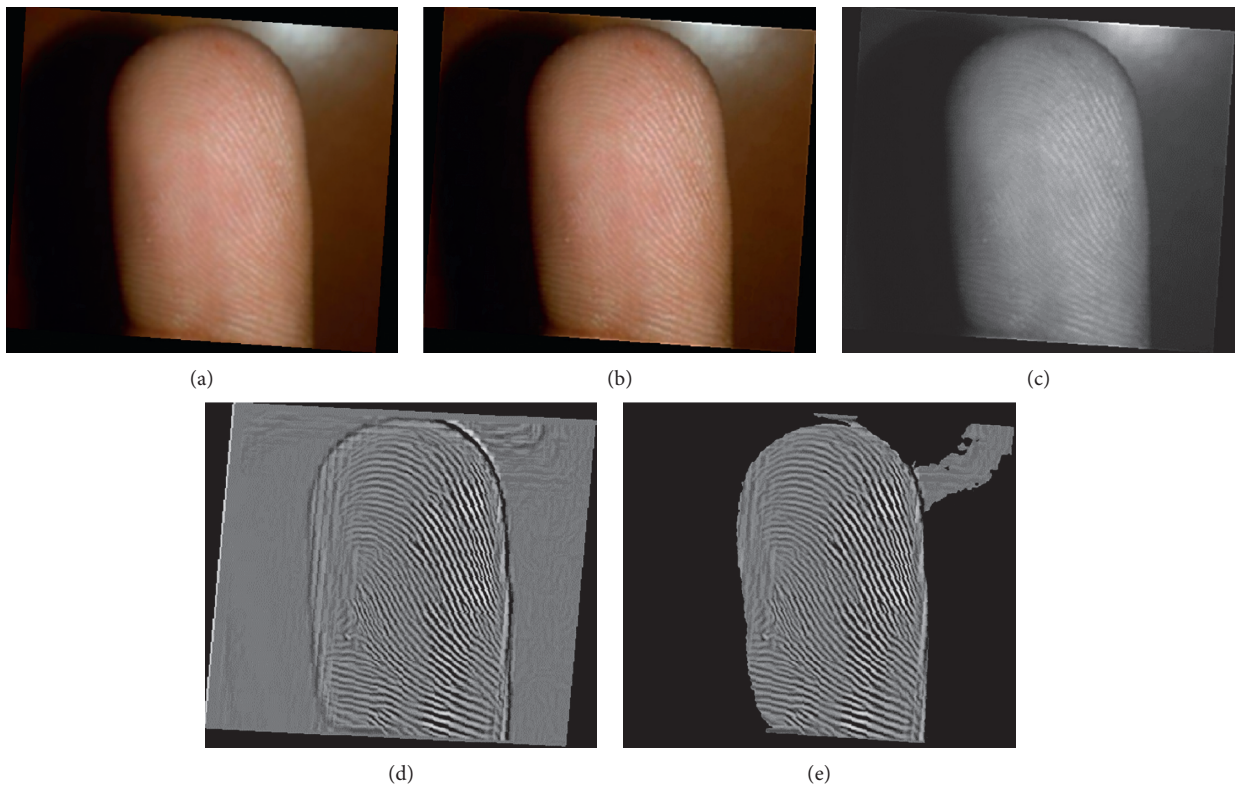


Figure 4: Stages of rotation alignment method applied to four images: (a–d) input intensity image, (e–h) segmented input image, (i–l) matched features including outliers, (m–p) matched inliers, and (q–t) derotated input image.



(a) (b) (c) (d) (e)

Figure 5: Application of preprocessing on the fingerprint image: (a) original input image, (b) unsharp masking, (c) grayscale conversion, (d) blockwise Fourier-based enhancement, and (e) extracted region of interest.

(a) (b) (c)

Figure 6: Applying edge thinning operations to an enhanced image: (a) binary segmentation, (b) nonmaximal suppression, and (c) minutiae extraction.

suppression results in more desirable thin ridges, as seen in Figure 6(b).

2.5. Minutiae Extraction. Minutiae are features that are used for biometric verification. Instead of storing fingerprint images in the database, minutiae are stored and compared. The most common method used for minutiae extraction is the crossing number (CN) [36].

For extracting minutiae, the 3×3 window shown in Figure 7 is moved across each pixel to calculate its crossing number using the following equation:

$$\mathrm{CN}(x, y) = \frac{1}{2} \sum_{i=1}^{8} \left| p_i - p_{i-1} \right|, \tag{8}$$

where p_i denotes the pixels around the center pixel at location (x, y). Using CN, the ridge pixel can then be classified as a ridge ending, bifurcation, or isolated point using Table 1. Figure 6(c) shows bifurcations and ridge endings represented by blue and red squares, respectively.

For each bifurcation and ridge ending, the Euclidean distance $D_{x,y}$ between its spatial coordinate (x, y) and $(x_{\mathrm{ROI}}^{\min}, y_{\mathrm{ROI}}^{\min})$, the normalized gradient value $G^{x,y}$, and the

P_4	P_3	P_2
P_5	(x,y)	P_1
P_6	P_7	P_8

Figure 7: 3×3 window used in minutiae extraction.

Table 1: Ridge pixel classifications using the crossing number.

CN	Decision
0	Isolated point
1	Ridge ending
2	Continuing ridge
3	Bifurcation
4	Crossing

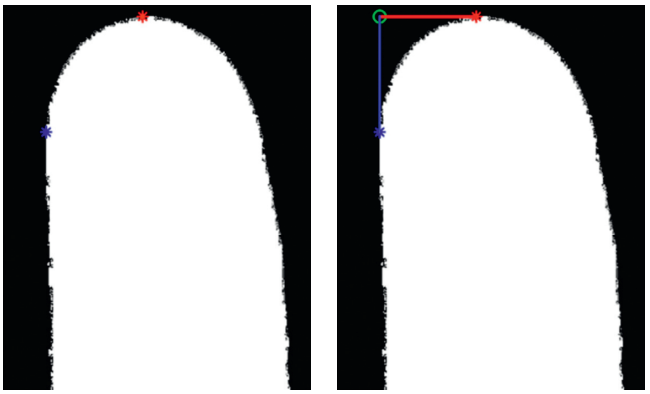
gradient orientation $\alpha^{x,y}$ are stored in $\mathbf{S}_{\mathbf{BF}}^{\mathbf{in}}$ and $\mathbf{S}_{\mathbf{RE}}^{\mathbf{in}}$, respectively, for storage or comparison. Hence, every bifurcation and ridge ending point is stored in $\mathbf{S}_{\mathbf{BF}}^{\mathbf{in}}$ or $\mathbf{S}_{\mathbf{RE}}^{\mathbf{in}}$ in the form of 3D point $(D_{\mathrm{BF}}^{x,y}, G_{\mathrm{BF}}^{x,y}, \alpha_{\mathrm{BF}}^{x,y})$ and $(D_{\mathrm{RE}}^{x,y}, G_{\mathrm{RE}}^{x,y}, \alpha_{\mathrm{RE}}^{x,y})$, respectively. The values donated by $x_{\mathrm{ROI}}^{\min}$ and $y_{\mathrm{ROI}}^{\min}$ are x-axis and y-axis points taken from two ROI points corresponding to the minimum x-axis and y-axis values. Figure 8 shows an example of calculation of $(x_{\mathrm{ROI}}^{\min}, y_{\mathrm{ROI}}^{\min})$ that is represented by a green circle. This was determined using the minimum x-axis ROI point represented by a blue cross and the minimum y-axis ROI point represented by a red cross.

2.6. Fingerprint Registration and Matching. Extracted minutiae are either stored or compared with minutiae already existing in the database as required by the use case. Suppose that the ridge endings extracted from the input image are represented by the set $\mathbf{S}_{\mathbf{BF}}^{\mathbf{in}}$ and that bifurcations are represented by $\mathbf{S}_{\mathbf{RE}}^{\mathbf{in}}$ as follows:

$$\mathbf{S}_{\mathbf{RE}}^{\mathbf{in}} = \left\{\left(D_{\mathrm{RE1}}^{\mathrm{in}}, G_{\mathrm{RE1}}^{\mathrm{in}}, \propto_{\mathrm{RE1}}^{\mathrm{in}}\right), \left(D_{\mathrm{RE2}}^{\mathrm{in}}, G_{\mathrm{RE2}}^{\mathrm{in}}, \propto_{\mathrm{RE2}}^{\mathrm{in}}\right), \ldots, \left(D_{\mathrm{REN}_{\mathrm{RE}}^{\mathrm{in}}}^{\mathrm{in}}, G_{\mathrm{REN}_{\mathrm{RE}}^{\mathrm{in}}}^{\mathrm{in}}, \propto_{\mathrm{REN}_{\mathrm{RE}}^{\mathrm{in}}}^{\mathrm{in}}\right)\right\}, \tag{9}$$

$$\mathbf{S}_{\mathbf{BF}}^{\mathbf{in}} = \left\{\left(D_{\mathrm{BF1}}^{\mathrm{in}}, G_{\mathrm{BF1}}^{\mathrm{in}}, \propto_{RE1}^{\mathrm{in}}\right), \left(D_{\mathrm{BF2}}^{\mathrm{in}}, G_{\mathrm{BF2}}^{\mathrm{in}}, \propto_{\mathrm{BF2}}^{\mathrm{in}}\right), \ldots, \left(D_{\mathrm{BFN}_{\mathrm{BF}}^{\mathrm{in}}}^{\mathrm{in}}, G_{\mathrm{BFN}_{\mathrm{BF}}^{\mathrm{in}}}^{\mathrm{in}}, \propto_{\mathrm{BFN}_{\mathrm{BF}}^{\mathrm{in}}}^{\mathrm{in}}\right)\right\}, \tag{10}$$

where $\mathbf{S}_{\mathbf{RE}}^{\mathbf{in}}$ and $\mathbf{S}_{\mathbf{BF}}^{\mathbf{in}}$ are sorted in an ascending order according to the calculated Euclidean distance such that $D_{\mathrm{RE}(k)}^{\mathrm{in}} \leq D_{\mathrm{RE}(k+1)}^{\mathrm{in}}$ and $D_{\mathrm{BF}(l)}^{\mathrm{in}} \leq D_{\mathrm{BF}(l+1)}^{\mathrm{in}}$ for all $k \in 1, 2, \ldots, N_{\mathrm{RE}}^{\mathrm{in}}$ and $l \in 1, 2, \ldots, N_{\mathrm{BF}}^{\mathrm{in}}$. These sets must be compared with the stored features of a number of fingerprints N_{DB} stored in the database, as follows:



(a) (b)

Figure 8: Determination of point $(x_{\mathrm{ROI}}^{\min}, y_{\mathrm{ROI}}^{\min})$: (a) minimum x-axis and y-axis ROI points and (b) $(x_{\mathrm{ROI}}^{\min}, y_{\mathrm{ROI}}^{\min})$ determination using two minimum axis points.

$$\mathbf{S}_{\mathbf{RE}}^{\mathbf{DB}} = \left\{S_{\mathrm{RE}}^{\mathrm{DB1}}, S_{\mathrm{RE}}^{\mathrm{DB2}}, \ldots, S_{\mathrm{RE}}^{\mathrm{DBN}_{\mathrm{RE}}^{\mathrm{DB}}}\right\}, \tag{11}$$

$$\mathbf{S}_{\mathbf{BF}}^{\mathbf{DB}} = \left\{S_{\mathrm{BF}}^{\mathrm{DB1}}, S_{\mathrm{BF}}^{\mathrm{DB2}}, \ldots, S_{\mathrm{BF}}^{\mathrm{DBN}_{\mathrm{BF}}^{\mathrm{DB}}}\right\}, \tag{12}$$

where

$$S_{\mathrm{RE}}^{\mathrm{DB}i} = \left\{\left(D_{\mathrm{RE1}}^{\mathrm{DB}i}, G_{\mathrm{RE1}}^{\mathrm{DB}i}, \propto_{\mathrm{RE1}}^{\mathrm{DB}i}\right), \left(D_{\mathrm{RE2}}^{\mathrm{DB}i}, G_{\mathrm{RE2}}^{\mathrm{DB}i}, \propto_{\mathrm{RE2}}^{\mathrm{DB}i}\right), \ldots, \left(D_{\mathrm{REN}_{\mathrm{RE}}^{\mathrm{DB}i}}^{\mathrm{DB}i}, G_{\mathrm{REN}_{\mathrm{RE}}^{\mathrm{DB}i}}^{\mathrm{DB}i}, \propto_{\mathrm{REN}_{\mathrm{RE}}^{\mathrm{DB}i}}^{\mathrm{DB}i}\right)\right\}, \tag{13}$$

$$S_{\mathrm{BF}}^{\mathrm{DB}i} = \left\{\left(D_{\mathrm{BF1}}^{\mathrm{DB}i}, G_{\mathrm{BF1}}^{\mathrm{DB}i}, \propto_{\mathrm{RE1}}^{\mathrm{DB}i}\right), \left(D_{\mathrm{BF2}}^{\mathrm{DB}i}, G_{\mathrm{BF2}}^{\mathrm{DB}i}, \propto_{\mathrm{BF2}}^{\mathrm{DB}i}\right), \ldots, \left(D_{BFN_{\mathrm{BF}}^{\mathrm{DB}i}}^{\mathrm{DB}i}, G_{BFN_{\mathrm{BF}}^{\mathrm{DB}i}}^{\mathrm{DB}i}, \propto_{BFN_{\mathrm{BF}}^{\mathrm{DB}i}}^{\mathrm{DB}i}\right)\right\}. \tag{14}$$

$\mathbf{S}_{\mathbf{RE}}^{\mathbf{DBi}}$ and $\mathbf{S}_{\mathbf{BF}}^{\mathbf{DBi}}$ are also sorted in an ascending order according to the Euclidean distances $D_{\mathrm{RE}m}^{\mathrm{DB}i}$ and $D_{\mathrm{BF}n}^{\mathrm{DB}i}$ where $m \in \{1, 2, \ldots, N_{\mathrm{RE}}^{\mathrm{DB}i}\}$ and $n \in \{1, 2, \ldots, N_{\mathrm{BF}}^{\mathrm{DB}i}\}$. Suppose that $\mathbf{S}_{\mathbf{RE}}^{\mathbf{in}}$ and $\mathbf{S}_{\mathbf{BF}}^{\mathbf{in}}$ are to be compared with the stored features of the ith fingerprint having feature sets $\mathbf{S}_{\mathbf{RE}}^{\mathbf{DBi}}$ and $\mathbf{S}_{\mathbf{BF}}^{\mathbf{DBi}}$. First, $N_{\min}^{\mathrm{RE}}$, $N_{\min}^{\mathrm{BF}}$, $N_{\max}^{\mathrm{RE}}$, and $N_{\max}^{\mathrm{BF}}$ are to be determined using the following equations:

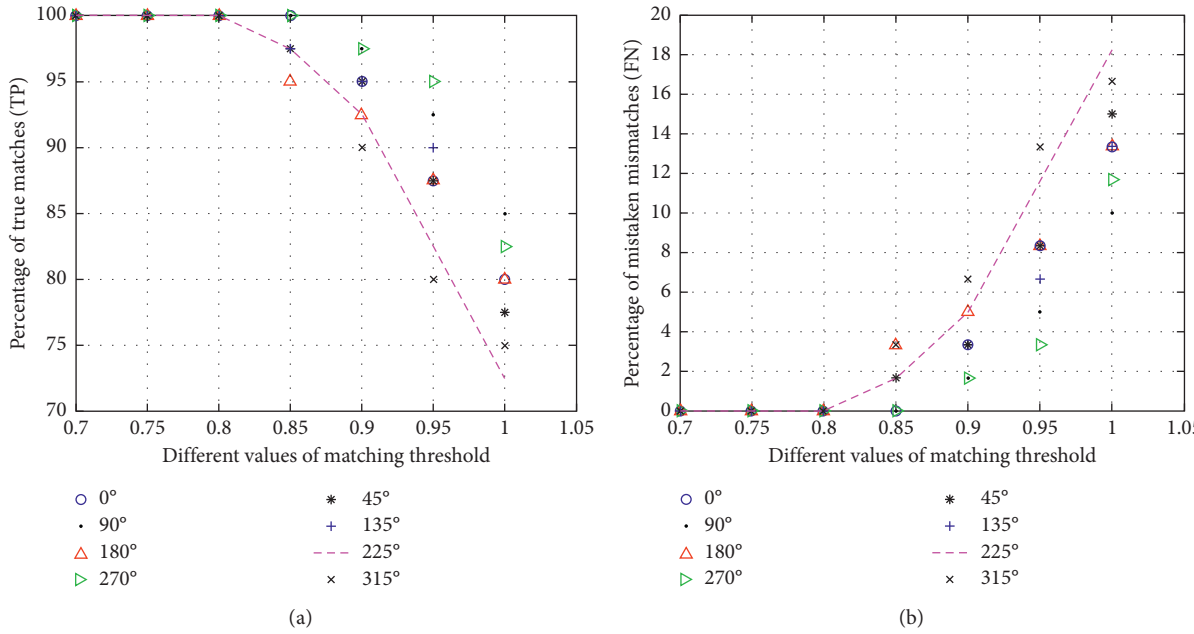


Figure 9: Percentage of (a) TP and (b) FN for different values of t_{match}.

$$N_{\text{min}}^{\text{RE}} = \min\left(N_{\text{RE}}^{\text{in}}, N_{\text{RE}}^{\text{DB}i}\right), \tag{15}$$

$$N_{\text{min}}^{\text{BF}} = \min\left(N_{\text{BF}}^{\text{in}}, N_{\text{BF}}^{\text{DB}i}\right), \tag{16}$$

$$N_{\text{max}}^{\text{RE}} = \max\left(N_{\text{RE}}^{\text{in}}, N_{\text{RE}}^{\text{DB}i}\right), \tag{17}$$

$$N_{\text{max}}^{\text{BF}} = \max\left(N_{\text{BF}}^{\text{in}}, N_{\text{BF}}^{\text{DB}i}\right), \tag{18}$$

where $N_{\text{RE}}^{\text{in}}$ and $N_{\text{RE}}^{\text{DBi}}$ are the number of ridge endings extracted from the input image and the ith image features stored in the database and $N_{\text{BF}}^{\text{in}}$ and $N_{\text{BF}}^{\text{DB}i}$ are the number of extracted bifurcations. The input fingerprint features are compared with the ith fingerprint features if the condition in (19) is justified:

$$\begin{aligned} N_{\text{min}}^{\text{RE}} &\geq 0.8 N_{\text{max}}^{\text{RE}}, \\ N_{\text{min}}^{\text{BF}} &\geq 0.8 N_{\text{max}}. \end{aligned} \tag{19}$$

For matching, all 3D bifurcation and ridge ending points of the input fingerprint and the ith fingerprint from the database and input fingerprint are stored in $\mathbf{M}_{\mathbf{RE}}^{\mathbf{ref}}$, $\mathbf{M}_{\mathbf{BF}}^{\mathbf{ref}}$, $\mathbf{M}_{\mathbf{RE}}^{\mathbf{in}}$, and $\mathbf{M}_{\mathbf{BF}}^{\mathbf{in}}$ respectively. $\mathbf{M}_{\mathbf{RE}}^{\mathbf{ref}}$ is compared with $\mathbf{M}_{\mathbf{RE}}^{\mathbf{in}}$, and $\mathbf{M}_{\mathbf{BF}}^{\mathbf{ref}}$ is compared with $\mathbf{M}_{\mathbf{BF}}^{\mathbf{in}}$. The steps for matching are

(a) If $\mathbf{M}_{\mathbf{BF}}^{\mathbf{in}}$ has fewer bifurcations, store $\mathbf{M}_{\mathbf{BF}}^{\mathbf{in}}$ in $\mathbf{M}_{\mathbf{BF}}^{\mathbf{min}}$ and $\mathbf{M}_{\mathbf{BF}}^{\mathbf{ref}}$ in $\mathbf{M}_{\mathbf{BF}}^{\mathbf{max}}$ and vice versa. Initialize bifurcation matching score score_{BF} to zero.

(b) Compare each bifurcation 3D point in $\mathbf{M}_{\mathbf{BF}}^{\mathbf{min}}$ at the pth position with all bifurcation points at locations $p+t$ in $\mathbf{M}_{\mathbf{BF}}^{\mathbf{max}}$ where $t \in \{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$.

(c) If $|G_{p+t}^{\text{max}} - G_{p}^{\text{min}}| < T_1$ and $|\alpha_{p+t}^{\text{max}} - \alpha_{p}^{\text{min}}| < T_2$ is true for any value of t, then increment score_{BF} and move to the next bifurcation point for matching. Here, G_{p}^{min} and α_{p}^{min} are the magnitude and orientation of the normalized gradient of bifurcation of $M_{\text{BF}}^{\text{min}}$ at position p, and G_{p+t}^{max} and α_{p+t}^{m} max are the magnitude and orientation of $\mathbf{M}_{\mathbf{BF}}^{\mathbf{max}}$ at any of the eleven locations in Step b.

(d) Repeat Steps a–c, comparing ridge endings and updating score_{RE} to track the number of ridges matched.

(e) If $(\text{score}_{\text{BF}} + \text{score}_{\text{RE}}/N_{\text{min}}^{\text{BF}} + N_{\text{min}}^{\text{RE}}) > t_{\text{match}}$, two fingerprints are said to be matched.

3. Results and Discussion

MATLAB 2018a was used in this study. A value of $\sigma = 2.5$ was used, and a Gaussian window of size $(2\sigma + 1) \times (2\sigma + 1), T_1 = .06$, and $T_2 = 10^{\circ}$ was used throughout the work. There was no dataset available for evaluation purposes, so features from the fingerprints of 40 people were stored in a database. Features obtained from digital images of the fingers of 100 people, including those of registered people, were compared with the features already stored in the database in order to evaluate the performance of the proposed algorithm. Every person provided eight images of their finger at approximately 90°, 135°, 180°, 225°, 270°, 315°, 0°, and 45°.

The performance of the proposed method was evaluated by analyzing the performance of the derotation and matching methods. It was not possible to calculate the exact errors in the calculation of the rotation angles, so error calculation was done manually for ten random inputs for each set of rotation angles.

According to this manual evaluation, the errors fell in the range from ~−2.5 to ~2.5° for all cases. The error range for the angles 0°, 180°, and 270° was from ~−1.5° to

1.5°. For a rotation angle of 90°, the error was approximately 0°.

True positives, true negatives, false positives, and false negatives were calculated for different values of the matching threshold t_{match}. True positives, true negatives, false positives, and false negatives are given in (20)–(23) as

$$\text{TP} = \frac{n_{\text{correctMatches}} \times 100}{n_{\text{ActualMatches}}}, \tag{20}$$

$$\text{TN} = \frac{n_{\text{correctMissMatches}} \times 100}{n_{\text{ActualMismatches}}}, \tag{21}$$

$$\text{FP} = \frac{n_{\text{mistakenMatches}} \times 100}{n_{\text{total}}}, \tag{22}$$

$$\text{FN} = \frac{n_{\text{mistakenMisMatches}} \times 100}{n_{\text{ActualMismatches}}}. \tag{23}$$

Figure 9 shows a comparison of the percentages of true matches (TP) and mistaken mismatches (FN). It can be seen from Figure 9 that a value of $0.7 \leq t_{\text{match}} \leq 0.8$ successfully verified all 40 of the registered fingerprints irrespective of their angle of rotation. However, for values of $t_{\text{match}} > 0.8$, mistaken mismatches occurred in the range from ~(2% – 18%). True mismatches (TN) occurred in 100% of the cases, and mistaken matches (FP) occurred in 0% of all cases.

4. Conclusions and Future Work

This paper proposes a fingerprint detection and verification method that uses a digital image of a finger as an input instead of a sensor. The input fingerprint image can be of any resolution and depends on the quality of the digital camera. Often, the finger in the image is not vertically aligned. Therefore, the input image is derotated by applying skin color-based binary segmentation, followed by downscaling or upscaling of the input image with a scale calculated using the segmented input image and a reference image. The angle of rotation of the input finger is then calculated by applying SURF-based feature extraction on both binary images. The features detected in both images are matched, and outliers are removed from the matched points using the M-estimator. The angle between the matched inliers is calculated using a homography matrix, and the input image is derotated by the calculated angle. After derotation, enhancement techniques including unsharp masking and Fourier-based blockwise enhancement are applied to increase the visibility of the ridges in the input image. Image enhancement is followed by the application of Otsu thresholding to segment the ridge regions, and then nonmaximal suppression is applied for edge thinning. A crossing number technique is used to extract ridge endings and bifurcations. This is followed by the proposed Euclidean distance-based matching to detect a fingerprint match or mismatch based on a matching score. The proposed method is a novel method, and simulation results show that if the matching threshold is kept within a suggested range, accuracy of 100% can be achieved.

Scar removal-based fingerprint detection using digital images and edge-preserving image interpolation techniques could extend the proposed method and is being considered for future work.

Abbreviations

STFT:	Short-time Fourier transform analysis
ROI:	Region of interest
SURF:	Speeded up robust features
M-estimator:	Maximum-likelihood estimator
CN:	Crossing number
TP:	True positives
TN:	True negatives
FP:	False positives
FN:	False negatives.

Data Availability

The data used to support the findings of this study can be obtained by contacting the corresponding author through e-mail: dhlee77@hanyang.ac.kr.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Sajid Khan performed all the simulations and worked on the solution of most of the problems under the supervision of Dong-Ho Lee. Asif Khan provided helped in finalization of the content of manuscript and writing of the manuscript. Ahmad Waqas and Abdul Rehman Gilal collected the fingerprints for experiments and validation. Zahid Hussain Khand helped in the understanding of mathematical models and equations that are used in this research.

References

[1] S. Khan and D. H. Lee, "An adaptive dynamically weighted median filter for impulse noise removal," *EURASIP Journal on Advances in Signal Processing*, vol. 2017, no. 1, pp. 1–14, 2017.

[2] S. Khan, D.-H. Lee, M. A. Khan, A. R. Gilal, and G. Mujtaba, "Efficient edge-based image interpolation method using neighboring slope information," *IEEE Access*, vol. 7, pp. 133539–133548, 2019.

[3] S. Khan and D. Lee, "Efficient deinterlacing method using simple edge slope tracing," *Optical Engineering*, vol. 54, no. 10, pp. 103108–103111, 2015.

[4] M. J. Burge and K. W. Bowyer, *Handbook of Iris Recognition*, Springer Publishing Company, Berlin, Germany, 2013.

[5] D. S. Moon, K. Y. Moon, Y. O. O. Jang-Hee et al., "Fingerprint verification method and apparatus with high security," U.S. Patent and Trademark Office, Washington, DC, USA, U.S. Patent No. 8,699,799, 2014.

[6] H. Guesmi, H. Trichili, A. M. Alimi, and B. Solaiman, "Fingerprint verification system based on curvelet transform

and possibility theory," *Multimedia Tools and Applications*, vol. 74, no. 9, pp. 3253–3272, 2015.

[7] K. B. Raja, R. Raghavendra, V. K. Vemuri, and C. Busch, "Smartphone based visible iris recognition using deep sparse filtering," *Pattern Recognition Letters*, vol. 57, pp. 33–42, 2015.

[8] R. Jafri and H. R. Arabnia, "A survey of face recognition techniques," *Journal of Information Processing Systems*, vol. 5, no. 2, pp. 41–68, 2009.

[9] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," *BMVC*, vol. 1, no. 3, pp. 1–12, 2015.

[10] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, "Deepface: closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708, Columbus, OH, USA, June 2014.

[11] H. Rai and A. Yadav, "Iris recognition using combined support vector machine and Hamming distance approach," *Expert Systems with Applications*, vol. 41, no. 2, pp. 588–593, 2014.

[12] A. I. Ismail, H. S. Ali, and F. A. Farag, "Efficient enhancement and matching for iris recognition using SURF," in *Proceedings of the Information Technology: Towards New Smart World*, pp. 1–5, Riyadh, Saudi Arabia, February 2015.

[13] Z. M. Kovacs-Vajna, "A fingerprint verification system based on triangular matching and dynamic time warping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1266–1276, 2000.

[14] X. Luo, J. Tian, and Y. Wu, "A minutiae matching algorithm in fingerprint verification," *Pattern Recognition*, vol. 4, no. 15, pp. 833–836, 2000.

[15] R. Cappelli, D. Maio, D. Maltoni, J. L. Wayman, and A. K. Jain, "Performance evaluation of fingerprint verification systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 3–18, 2006.

[16] Y. He, J. Tian, X. Luo, and T. Zhang, "Image enhancement and minutiae matching in fingerprint verification," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1349–1360, 2003.

[17] S. Khan, A. Waqas, M. A. Khan, and A. W. Ahmad, "A camera-based fingerprint registration and verification method," *International Journal of Computer Science and Network Security*, vol. 18, no. 11, pp. 26–31, 2018.

[18] K. Delac and M. Grgic, "A survey of biometric recognition methods," in *Proceedings of the Elmar-2004. 46th International Symposium on Electronics in Marine*, pp. 184–193, Zadar, Croatia, June 2004.

[19] P. Mahajan, M. Malekar, A. More, A. Wairagade, and B. Mahalakshmi, "Secured internet banking using fingerprint authentication," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 4, no. 3, pp. 6199–6208, 2016.

[20] R. Priya, V. Tamilselvi, and G. P. Rameshkumar, "A novel algorithm for secure internet banking with finger print recognition," in *Proceedings of the International Conference on Embedded Systems (ICES)*, pp. 104–109, Coimbatore, India, July 2014.

[21] D. Baldisserra, A. Franco, D. Maio, and D. Maltoni, "Fake fingerprint detection by odor analysis," in *Proceedings of the International Conference on Biometrics*, pp. 265–272, Hong Kong, China, January 2006.

[22] S. B. Nikam and S. Agarwal, "Ridgelet-based fake fingerprint detection," *Neurocomputing*, vol. 72, no. 10–12, pp. 2491–2506, 2009.

[23] H. Choi, R. Kang, K. Choi, A. T. B. Jin, and J. H. Kim, "Fake-fingerprint detection using multiple static features," *Optical Engineering*, vol. 48, no. 4, Article ID 047202, 2009.

[24] B. Y. Hiew, A. B. J. Teoh, and O. S. Yin, "A secure digital camera based fingerprint verification system," *Journal of Visual Communication and Image Representation*, vol. 21, no. 3, pp. 219–231, 2010.

[25] B. Y. Hiew, A. B. Teoh, and Y. H. Pang, "Digital camera based fingerprint recognition," in *Proceedings of the IEEE International Conference on Telecommunications and Malaysia International Conference on Communications*, pp. 676–681, Penang, Malaysia, May 2007.

[26] B. Y. Hiew, A. B. Teoh, and D. C. Ngo, "Automatic digital camera based fingerprint image preprocessing," in *Proceedings of the International Conference on Computer Graphics, Imaging and Visualisation*, pp. 182–189, Sydney, Australia, July 2006.

[27] A. Sankaran, A. Malhotra, A. Mittal, M. Vatsa, and R. Singh, "On smartphone camera based fingerphoto authentication," in *Proceedings of the International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–7, Arlington, VA, USA, September 2015.

[28] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[29] D. E. Tyler, "A distribution-free M-Estimator of multivariate scatter," *The Annals of Statistics*, vol. 15, no. 1, pp. 234–251, 1987.

[30] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 666–673, Kerkyra, Greece, September 1999.

[31] S. L. Phung, D. Chai, and A. Bouzerdoum, "Adaptive skin segmentation in color images," in *Proceedings of the ICASSP*, pp. 353–356, Hong Kong, China, April 2003.

[32] G. Ramponi, N. K. Strobel, S. K. Mitra, and T. H. Yu, "Nonlinear unsharp masking methods for image contrast enhancement," *Journal of Electronic Imaging*, vol. 5, no. 3, pp. 353–367, 1996.

[33] S. Chikkerur, C. Wu, and V. Govindaraju, "A systematic approach for feature extraction in fingerprint images," in *Biometric Authentication*, pp. 344–350, Springer, Berlin, Germany, 2004.

[34] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[35] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[36] L. Wieclaw, "A minutiae-based matching algorithms in fingerprint recognition systems," *Journal of Medical Informatics & Technologies*, vol. 13, pp. 65–72, 2009.

Hindawi
Scientific Programming
Volume 2020, Article ID 6597316, 24 pages
https://doi.org/10.1155/2020/6597316

Research Article

Fuzzy Multicriteria Decision-Making Approach for Measuring the Possibility of Cloud Adoption for Software Testing

Sikandar Ali,[1,2] **Niamat Ullah**,[3] **Muhammad Faisal Abrar**,[4] **Zhongguo Yang**,[5] **and Jiwei Huang**[1,2]

[1]*Department of Computer Science and Technology, China University of Petroleum, Beijing 102249, China*
[2]*Beijing Key Laboratory of Petroleum Data Mining, China University of Petroleum, Beijing, Beijing 102249, China*
[3]*Department of Computer Science University of Buner, Buner 17290, Pakistan*
[4]*Department of Computer Software Engineering, University of Engineering and Technology, Mardan 23200, Pakistan*
[5]*School of Information Science and Technology, North China University of Technology, Beijing, China*

Correspondence should be addressed to Jiwei Huang; huangjw@cup.edu.cn

Received 20 October 2019; Revised 27 December 2019; Accepted 9 January 2020; Published 30 April 2020

Guest Editor: Rahman Ali

To reduce costs and improve organizational efficiency, the adoption of innovative services such as Cloud services is the current trend in today's highly competitive global business venture. The aim of the study is to guide the software development organization (SDO) for Cloud-based testing (CBT) adoption. To achieve the aim, this study first explores the determinants and predictors of Cloud adoption for software testing. Grounded on the collected data, this study designs a technology acceptance model using fuzzy multicriteria decision-making (FMCDM) approach. For the stated model development, this study identifies a list of predictors (main criteria) and factors (subcriteria) using systematic literature review (SLR). In the results of SLR, this study identifies seventy subcriteria also known as influential factors (IFs) from a sample of 136 papers. To provide a concise understanding of the facts, this study classifies the identified factors into ten predictors. To verify the SLR results and to rank the factors and predictors, an empirical survey was conducted with ninety-five experts from twenty different countries. The application value in the industrial field and academic achievement of the present study is the development of a general framework incorporating fuzzy set theory for improving MCDM models. The model can be applied to predict organizational Cloud adoption possibility taking various IFs and predictors as assessment criteria. The developed model can be divided into two main parts, ranking and rating. To measure the success or failure contribution of the individual IFs towards successful CBT adoption, the ranking part of the model will be used, while for a complete organizational assessment in order to identify the weak area for possible improvements, the assessment part of the model will be used. Collectively, it can be used as a decision support system to gauge SDO readiness towards successful CBT.

1. Introduction

Over the past decades, software applications are becoming complex, dynamic, component based, and distributed, therefore making software testing a challenging task [1]. Industry reports show that software of 20,000 lines of code might take seven weeks of uninterrupted execution for a thorough regression test [2]. Many software development organizations (SDOs) consider test process automation to shorten development cycle and decrease testing costs [3]. Moreover, it is time-consuming and laborious to conduct some types of testing manually like regression testing, while through test automation, they can be performed efficiently [4]. Once tests have been automated, they can be run repeatedly and quickly. Test automation also proves to be an economical method for the regression testing of software with lengthy maintenance spell [4]. However, this type of testing demands huge investment from an SDO on various testing tools and related resources which may go out of use over a period of time because technology and application advance over time. Since SDO is supposed to retain pace with the industrial changes, SDO finds it hard and unfeasible to upgrade them every time [2].

Due to the essential value-added feature of Cloud Computing (CC), paradigms such as on large resource pool,

self-service on demand, measured service, broad network access, and rapid elasticity application development, deployment, and testing in the CC environments have been greater than before. This results in the reduction of delivery time, increased productivity, and capital and operational cost saving. Besides numerous benefits, it also postures several challenges for the quality of service (QoS) assurance including high-availability, multitenancy issues, elasticity, privacy, and security [5]. CC is regarded by many authors as a paradigm shift in computing, in lieu of an important change in the way computing services are offered, developed, attained, and paid for [6]. In practice, in this model, a Cloud service provider (CSP), serving the public, uses a pool of computing services via multitenancy. Potential clients can acquire and use these services over the Internet with automated user-friendly interfaces. Using the pay-as-you-go model, the service consumption is automatically metered [7].

At present, Cloud services are offered in at least three different models, i.e., Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). These services are typically organized in one of the four infrastructures which are public, private, hybrid, or community Cloud settings. In the near past, numerous business organizations have shown a growing interest in the adoption of Cloud services to support various corporate functions [8]. CC is one of the five most persuasive technologies across the globe and is ranked the third widely invested significant IT service in 2013 [9]. According to the latest 2018 McKinsey report, over the next three years, organizations will make a fundamental shift from developing IT to utilizing IT [10]. Companies of all sizes are moving to off-premise Cloud services, among a significant move are noticed in large enterprises. The market share of CC is increasing fast, 30% small and medium businesses (SMBs) have already acquired five or more Cloud services while 60% have acquired a minimum of one Cloud service [11]. While cost reduction is often perceived to be the main factor in adopting CC services, the report shows that quality is a key predictor for Cloud adoption, while compliance and security remain key inhibitors for acceptance, specifically for large organizations [10].

As stated above there are three basic service models of CC, i.e., IaaS, PaaS, and SaaS. However, when we want to conduct Cloud testing, there emerges a new service model called Software Testing-as-a-Service (STaaS) or simply TaaS. A new term, known as TaaS or STaaS, has become a hot topic of discussion in different research groups as well as IT businesses and the Cloud industry [12]. The term of "TaaS" was originally coined and properly introduced by Tieto (http://www.tieto.com/) in 2009 in Denmark. Afterward, TaaS has been widely accepted by both industry and academic communities owing to its advantages in utility style service models, scalable testing environment, demanded testing services, and cost reduction [12]. This model has been assimilated into commercial products like Sogeti and Tieto. TaaS in a CC environment is deliberated as a new service and business model where a TaaS service provider creates a software testing environment and provides them as-a-testing service to accomplish testing tasks and activities for SDO to test application software under-test (SUT) or a web-based system running over a Cloud infrastructure [12]. In connection to software testing, the terms "testing in the Cloud," "Cloud-based testing (CBT)," "Cloud testing," "STaaS," and "TaaS" are now widely used as synonyms.

To cope with the challenges of software testing; employing CC technology would be the best choice. While many organizations are adopting CC with careful assurance, testing seems to be one of the areas where they are willing to be more optimistic to move. Adoption of CC for software testing process is considered to be a safe decision because testing data does not contain any sensitive business information [2]. However, the adoption of CC for software testing is still in the initial stage [13]. The adoption rate of CC depends on the identification of novel factors. Before taking any important adoption decision, the significant factors for Cloud adoption should be ranked. Studies focusing on the adoption of CC for software testing are still lacking. Furthermore, conducting testing in the CC environment is neither cost-effective nor it is the best possible solution to all testing problems. Furthermore, not every type of software is the best candidate for testing in the Cloud [14].

SDO concerned about CBT must be conscious about the characteristics and types of software that are suitable to be tested in the Cloud environment, which enables them to better employ CBT [15]. Unit testing (particularly large regression suites), performance testing, and high-volume automated testing are the ideal choices to be migrated to the Cloud environment [14]. Unlike traditional web-based software testing, TaaS on Cloud infrastructure has several distinct requirements and unique features [12]. Some of the requirements from different perspectives are (1) the applications or SUT must be accessible online. The SUT might be a SaaS application or non-SaaS application. Additionally, this takes into account different levels of testing, for example regression or performance testing; (2) diverse deployment models of the Cloud such as private, public, hybrid, or community are used to host testing platforms and infrastructure; (3) testing of the Cloud itself [16].

Moreover, it is not straightforward to move testing to the Cloud; several quantitative and qualitative factors influence the decision to adopt CC. For example, all the traditional artifacts must be relocated to a new platform while still adhering to the original quality attributes and functionality. Questions like why tests are migrated to the Cloud, which techniques will be suitable, which test needs to be migrated, and when to migrate to the Cloud demand to be answered before proceeding with the CBT [17]. When transferring testing to the Cloud platform, the tools that are used in the course of the testing need to be switched to an innovative Cloud environment while still coinciding with the development environment. These artifacts consist of the test environment (such as testbeds and tools), test plans, testing method, test cases, test results and its documentation, and so on. Equal rewards should be reaped provided that high risks are posed while moving application testing to the CC platform [14].

1.1. Aims and Objectives. Specifically, this study explores the determinants and predictors of Cloud adoption in the

context of software testing and develops a Cloud testing adoption assessment model (CTAAM) for decision-making towards CC adoption for software testing, specifically known as CBT. CTAAM was developed based on the input from experts' judgment using fuzzy multicriteria decision-making (FMCDM) approach. For the proposed model development, we performed a systematic literature review (SLR) for data collection by applying our customized search strings [18]. From results of the SLR study, we identify seventy influential factors (IFs) from a sample of 136 papers. To validate the SLR findings and to rank the IFs, an empirical survey was conducted in the software testing industry with ninety-five experts from 20 different countries. Based on the collected data, we performed a two-phase analysis based on FMCDM methods. We develop CTAAM model based on the two-phase analysis results. For evaluation of the CTAAM two case studies were conducted in SDO. To rank the IFs, we used our developed FMCDM based framework. For this purpose, we have developed an evaluation scale with 133 items based on previous researches.

1.2. Contribution of the Study. This study contributes positively to the academic literature on the Cloud adoption in the context of software testing by identifying various criteria and is the foundation of Cloud adoption for future research in this field. Unlike other researchers, we formulate Cloud adoption as MCDM problem. Since several qualitative and quantitative factors influence decision towards Cloud adoption indicating that adoption is a kind of MCDM problem. Furthermore, unlike other researchers, we have combined the fuzzy set theory with MCDM to better handle vagueness, uncertainty, expert heterogeneity, human biasness, and subjectivity. In detail, in our framework model, the linguistic terms with parallel values in triangular fuzzy numbers (TFNs) format are used to translate the linguistic term. To handle expert heterogeneity, this study makes CTAAM capable of accumulating the decision of several experts by means of various predictors and factors as measurement criteria.

The model will do an organization assessment based on various IFs and predictors as evaluation criteria and can be applied to predict organizational Cloud adoption possibility taking various IFs and predictors as assessment criteria. The developed model can be divided into two main parts ranking and rating. To measure the success or failure contribution of the individual IFs towards successful CC adoption the ranking part of the model will be used while for a complete organizational assessment in order to identify the weak area for possible improvements the assessment part of the model will be used. Collectively, it can be used as a decision support system (DSS) to gauge SDO readiness towards successful CC adoption for CBT. The industrial contribution of the model is that it can be used as an assessment tool for SDO vendors and will indicate their weakness using an extended fuzzy version of the Motorola instrument specifically extended for this study. The models developed through FMCDM approach can handle uncertainty and vagueness in the expert judgments but they cannot identify the weak and strong areas of an organization. Therefore, our framework model in this study uses Motorola assessment tool [19, 20] in the case study. Similar to our previous studies [21–23], this study also prolonged the Motorola evaluation tool to a fuzzy environment by suggesting the TFNs scale for its three dimensions. The existing Motorola guidelines [19, 20] were appropriate for a 5-point Likert scale only. We update the guidelines to 7-point Likert and introduce TFNs instead of even numbers (0, 2, 4, 6, and 10). We have designed seven point linguistic scales for assessing the importance weights of 133 items. Scale development is another contribution.

Theoretical contribution of the fuzzy study is the design and structure of a general framework to improve multicriteria decision models. Other researchers and practitioners can use the proposed model structure and procedure as a template to develop FMCDM models and framework for decision-making, organizational capability improvement, and assessment for enterprises, organization, or companies in the other relevant fields. Moreover, the ranking part of the CTAAM can be adopted to design a standalone prioritization or weighting tool and to rank the impact of different risk or success factors while the assessment part of the CTAAM can be adopted to develop mini assessment tool. Collectively, it can be adopted as a DSS.

1.3. Paper Outline. The rest of the paper is organized as follows. In the second section, a brief review of the studies conducted in the domain of CBT followed by quick motivation for using fuzzy set theory is presented. Section 3 describes the research methodologies for data collection and analysis. This is then followed by results obtained through ranking and rating in Section 4. Section 5 discusses the results while Section 6 discusses the study limitations. Finally, Section 7 concludes the paper.

2. Background and Motivation

Cloud testing is the process of software testing based on CC technology [5]. Cloud testing can be testing the Cloud itself or TaaS over Cloud. In this study, we focus on testing applications in the Cloud, i.e., TaaS. A group of expert architects and performance testers from the UK leading web load and performance testing company (http://www.CloudTesting.com/) initiated Cloud-based testing service for load and performance testing of website and Web applications [24].

The present literature around CBT is in the form of reports and industrial white papers. However, as the academic research on CBT gains popularity, scientific research has also been growing [22, 25–29]. Researchers examine specific types of testing [28], its architecture [30], applicability through case studies [31], experiments [31], and industrial survey [25]. Other related research, focusing on overviewing [32], surveying [25, 33], and summarizing the existing work [34].

The survey paper [35] poses some questions to the CC research community, which involve suitable solutions to software testing. Fernandes and Gemmer [36] have defined

the CC paradigm and its implications for IT organizations. Bertolino et al. [37] identify various references and buzzwords. Bai et al. [24] suggest new developments in Cloud testing tools like SaaS testing, cross-Cloud testing, real-time results processing and dynamic adaptive testing, dependability testing, service-level-agreement conformance testing, and security and reliability testing. Nachiyappan and Justus [38] give an overview of Cloud testing tools. In a comparative study, they discuss various challenges to CC testing tools. Titinen [39] in their master thesis identify critical factors for software testing tools selection. They also classify testing tools into test management, execution, comparison, framework, and measurement tools. Murthy and Suma [40] conduct a study on CC testing tools, and Devasena et al. [41] propose a load testing tool for Cloud.

However, this research area is still immature [26]. For the time being, a vast majority of the preceding researches addresses the testing of the core architecture of the Cloud. Furthermore, the majority of the preceding work focuses on Cloud service evaluation. Conversely, customization of those services regarding software testing has been largely ignored in the preceding literature. What's more, few studies focus on Cloud adoption [42–48] but they are not conducted form software testing, the only exception are [22, 25]. A recent survey on testing in the Cloud [49] identified lack of a Cloud testing adoption model that guides decision-maker, when and how to move testing to the Cloud, based on various decision factors as a research gap. Therefore, our study will be an attempt to bridge some of the most important research gaps by developing a research model that is adaptable and scalable for SDO.

2.1. Motivation and Novelty. The decisions on whether to adopt Cloud as a testing tool are essential since incorrect decisions might consequence in a loss in terms of resources and efforts. A shortage of knowledge and information about a certain technology makes it uncertain and vague. Thus, unavoidably, most of the critical decisions are made under nondeterministic conditions while any carelessness may result in the dire consequences. Models and frameworks have been recommended to support the decision-makers for CC adoption, incorporating various aspects such as demand behaviors, business objectives, and quality of services, migration cost, and technical challenges. For instance, an MCDM model for ranking and selecting Cloud services includes six factors, namely, accountability, costs, assurance, agility, performance, and security [50]. Although there are numerous models for decision-making [42–48, 51–55] but they are not conducted form software testing, the only exceptions are [14, 22, 25]. Furthermore, most of them [42–48, 51–53] are not based on the fuzzy set theory. Furthermore, they are proposed for either service selection [46] or Cloud provider selection [46]. Interestingly, none of them is proposed for software testing or TaaS. The only exception is found in [14], in which authors propose a decision framework called SMART-T. SMART-T originated from Service-Oriented Architecture (SOA) Migration, Adoption, and Reuse Technique (SMART) [56] where *T* shows testing. SMART-T contains three parts: business drivers, technical factors, and operational results. SMART-T is developed based on business drivers and technical factors. Additionally, it is not developed based on the fuzzy set theory. Moreover, we did not find any study from the perspective of CC adoption, handling vagueness, or uncertainty in experts' decision-making.

Our study will contribute to the existing literature on Cloud adoption and software testing. Unlike the preceding researchers, we formulate Cloud adoption as the MCDM problem. Since several qualitative and quantitative factors influence in decision towards Cloud adoption indicating that adoption is a kind of MCDM problem. Compared with the preceding studies, our study has the following significance. *Firstly*, very few studies on CBT have been conducted. *Secondly*, no compressive study on Cloud adoption for software can be found in the relevant literature. Therefore, this paper is the initial effort to provide new valuable discernments on the adoption of CC technology for software testing, which is considered to be an emerging research perspective in automated testing. *Thirdly*, besides the importance of CBT, very limited evidence in the relevant literature can be found that discuss aspects of CBT adoption through systematic theoretical evaluation and models. In fact, the application of operational research models and theories, especially technology adoption theories and the MCDM approaches for Cloud adoption in the context of software testing remains unexplored. Consequently, proposing a Cloud testing adoption assessment model (CTAAM) and linking various influential factors to its core constructs of organizational adoption intention is deliberated as a valuable tool, which is the fundamental innovation of this study. *Lastly*, previous studies have only utilized MCDM [45–47, 51, 57] approach in ranking the influential factors and its dimension and predicting the organizations' intention to adopt Cloud technologies but they are not conducted for software testing, the only exception is [25]. Besides, we did not find any study, handling vagueness or uncertainty in experts' decision-making as well as causal relations among the identified factors simultaneously from the perspective of CC adoption. To handle expert heterogeneity, this study develops a model capable of aggregating the decision of multiple experts by using multiple predictors and factors as measurement criteria. In our proposed framework, multiple experts can participate in both the ranking survey and rating assessment case study. Additionally, unlike the preceding studies, this study provides the experts with a self-effacing linguistic scale as presented in column 1 of Tables 1 and 2.

2.2. Application of Fuzzy Set Theory to Multicriteria Decision-Making (MCDM). In real-world decision problems experts typically have to decide with imprecise, vague, or incomplete data [58]. Vagueness exists in the natural language terms, such as good or best, better than, smaller than, considerable, important, significant, not implemented, partially implemented, fully implemented, achieving, achieved, outstanding, qualified, marginally qualified, etc. [59, 60]. Vagueness or fuzziness in data may be due to poor boundaries of scale

Table 1: Linguistic scale for ranking of items.

Linguistic terms	Corresponding scale
Not contributing at all	(0.0, 0.0, 0.1)
Contributing sometime	(0.0, 0.1, 0.3)
Slightly contributing	(0.1, 0.3, 0.5)
Not sure	(0.3, 0.5, 0.7)
Moderately contributing	(0.5, 0.7, 0.9)
Strongly contributing	(0.7, 0.9, 1.0)
Extremely contributing	(0.9, 1.0, 1.0)

Table 2: Linguistic scale for rating of items.

Linguistic term	Corresponding scale
Very poor implementation	(0.0, 0.0, 0.1)
Poor implementation	(0.0, 0.1, 0.3)
Weak implementation	(0.1, 0.3, 0.5)
Fair implementation	(0.3, 0.5, 0.7)
Marginally implemented	(0.5, 0.7, 0.9)
Fully implemented	(0.7, 0.9, 1.0)
Outstanding	(0.9, 1.0, 1.0)

[61]. All real numbers greater than one is a familiar example of a class of objects where boundaries are not clear or poorly defined [58]. To cope with the fuzziness usually articulated in decision data rising from the qualitative independent judgment of the experts due to the uncertainty, incomplete or vague data, the scale based on crisp numbers might be inadequate to model the real-world MCDM problems [62].

Furthermore, the influential factors are not likewise significant as the influential factors may be quite different regarding their contribution to success [59]. Thus, it is more forthright and realistic for experts to rank the IFs for example "factor F extremely contributing towards CC adoption" rather than to guess "factor "*F*" 70% contributing towards CC adoption". Converting heterogeneous information to a single form might result in a loss of important information [63].

In practice, decision-making usually requires subjective data provided by the expert [64]. It is obvious that different experts have different preference structures, knowledge levels, and complex decision capabilities based on their experiences [65, 66]; therefore, a good decision-making model must tolerate vagueness or ambiguity in expert judgment [60, 65]. Numerous researchers have recommended incorporating fuzzy set theory [54, 55, 58–67] dispensing subjectivity in decision-making by human experts. Fuzzy statistics in FMCDM methods are represented by fuzzy numbers via fuzzy sets. In fuzzy sets, the degree of association is allocated to objects in their universe of discourse. An object in this approach can fit some sophisticated classes of objects with only a partial association [67].

2.3. Fuzzy Multicriteria Decision-Making (FMCDM). Zadeh [67] is the first researcher to incorporate "fuzzy sets theory" and proposed the MCDM approach for subjective decision-making. It has been widely adopted to model uncertainties in human decision-making. It also meritoriously resolves uncertainties in collecting data for MCDM. Several researchers have identified that the subjective vagueness of expert judgment can be dispensed by integrating the fuzzy set theory [59, 60, 65, 66] with MCDM. In a fuzzy set, linguistic terms are used to denote fuzzy variables that are then translated to the corresponding numerical values using the predefined linguistic scale [65, 66].

2.4. Fuzzy Numbers and Fuzzy Sets. A fuzzy set has a membership function which assigns a score of association to objects within its universe of information between 0 and 1. Figure 1 shows the x coordinates of the three vertices l lower, m central, and, u upper of $\mu A(\alpha)$ in a fuzzy set A. Let U is a universal set having items $\{\alpha\}$ then, a fuzzy set $\widetilde{A}$ in the universe of information U defined by its membership function is as follows:

$$\mu_A(\alpha)\colon U \longrightarrow [0,1], \tag{1}$$

which assigns to each $\{\alpha\}$ a score of association to $\widetilde{A}$ in the interval [0, 1].Here $\mu_A(\alpha)$ = score of membership of α in $\widetilde{A}$, assign values in range of 0 to 1, i.e.,

$$\mu\overline{A}(\alpha) \in [0,1]. \tag{2}$$

A fuzzy set can also be represented by a continuous membership function $\mu\overline{A}(\alpha)$.

$$\mu\overline{A}(\alpha) = \begin{Bmatrix} 0, & \text{if } \alpha \le 1, \\ \dfrac{\alpha - 1}{m - l}, & \text{if } l \le \alpha \le m, \\ \dfrac{u - \alpha}{u - m}, & \text{if } m \le \alpha \le u, \\ 0, & \text{if } \alpha \ge u, \end{Bmatrix}. \tag{3}$$

2.5. Linguistic Scale. The conventional scaling approaches face difficulties in labeling the criteria that are overly complicated or hard to define [67]. To counter the consequences of fuzziness the linguistic scale is used. Linguistic terms with parallel TFNs offer practical means for dispensing incidents of subjectivity. The appropriateness of substitutes versus criteria and the importance weights of criteria are assessed using a linguistic scale with corresponding fuzzy numbers [59, 60]. Several researchers [21–23, 59, 60, 65, 66] recommend the linguistic scale for describing such situations.

Inspired by the above-reported studies, in this paper, the linguistic terms with parallel TFNs scores are used to reproduce the information. Precisely, we have designed seven Likert linguistic scales for ranking IFs via 133 items. Seven Likert linguistic scales as presented in Table 1 were used to rank the significance of the identified factors via its items found via SLR in phase 2.

Likewise, seven linguistic scales as presented in Table 2 grounded on Motorola evaluation tool are designed. The case study companies were asked to rate the level of

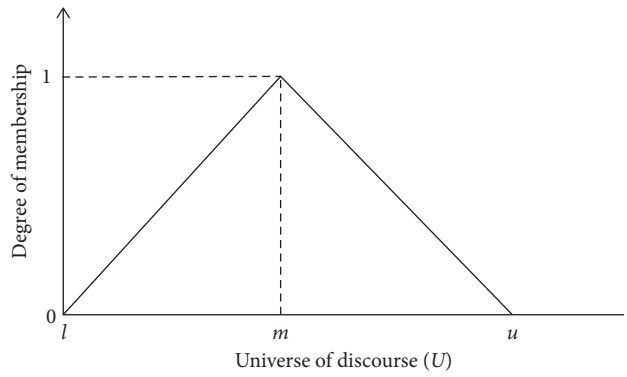


Figure 1: Coordinates of Triangular fuzzy numbers.

implementation of items across the three dimensions of Motorola evaluation tool [21], i.e., results, deployment, and approach as presented in Table 3.

3. Research Methodology

In terms of the proposed research model, we have developed it by including empirical research methodology based on a mixture of quantitative and qualitative research techniques for data collection and a two-phase FMCDM approach for analysis as deliberated in the following subsections.

Phase#1: to identify critical influential factors (CIFs), for data collection at the early stage, SLR was used.

Phase#2: to rank the identified CIFs using fuzzy FMCDM and at the same time to validate the SLRs findings, a questionnaire survey was carried out.

Phase#3: based on the inputs from the questionnaire survey and SLR the CTAAM was developed in phase 3.

Phase#4: to assess the CTAAM through the FMCDM approach and at the same time to evaluate its effectiveness in real-world environment case study was executed.

3.1. Systematic Literature Review (SLR). This study conducts SLR [68] to find the critical factors of the adoption of CC for software testing. Before conducting the SLR, we design a review plan known as a protocol. We have published the SLR protocol with initial results in a conference paper [18].

3.1.1. Review Plan. It reduces researchers' bias and enhances review accuracy and repeatability [68]. Particularly, it outlines rationale for the study, search strategies, study questions, literature inclusion/exclusion and quality assessment checklists, and data extraction and synthesis plans [68]. Figure 2 illustrates the main phases of the SLR process.

Construction of Search Terms. In terms of the searching phase, the scope of the study is acknowledged as follows:

Population. Primary studies on "Cloud-based testing," "testing in the Cloud," "Cloud testing," "testing-as-a-service," and "TaaS."

Intervention. Published primary studies reporting success factors for Cloud-based software testing.

Outcomes of Relevance. List of factors related to the CBT adoption.

Experimental Design. Any forms of empirical study.

The below example shows the aforementioned parts:

[What factors/challenges] ⟶ "INTERVENTION"

In [Cloud software testing] ⟶ "POPULATION"

Which impacts upon

[Adoption of Cloud] ⟶ "OUTCOMES OF RELEVANCE"

KEYWORDS_ABSTRACT_TITLE

(Software OR Application OR program) AND (Test* OR Validat* OR verification OR maintenance OR accepting) AND (Cloud OR Cloud computing OR TaaS OR Testing as a service) AND (factors OR motivators OR drivers OR elements OR parameters OR characteristics OR adopt* decision OR criteria OR "lesson learned")

Search plan: We used an automatic search method for publication selection because manual search is not easy for digital libraries where the numbers of available documents are over several thousands. The final search string is given below.

Literature Selection: In this step, the investigators have to list sources to be searched, inclusion, and exclusion and quality criteria.

Selection of sources: Following the guidelines for SLR [68], we selected seven search engines/databases related to the study subject to run an exhaustive search [68]. Databases are chosen based on the characteristics, such as coverage of indexed articles (journals, conferences, or books), availability of the study, and importance to the software engineering area. Following are the seven databases, we selected.

(1) ACM_digital library http://www.acm.org/
(2) Scholar_Google https://scholar.google.com
(3) IEEE_Xplore-http://ieeexplore.ieee.org/
(4) Cite_Seer-http://citeseer.ist.psu.edu/
(5) Science_Direct-http://www.sciencedirect.com/
(6) Springer_Link-http://link.springer.com/
(7) Wiley_online library-https://onlinelibrary.wiley.com/

Inclusion and Exclusion Criterion: An inclusion and exclusion criterion is defined to verify the importance and usefulness of each study. The relevant criteria are listed in Table 4.

Quality Criteria: The quality checks are listed in Table 5. The checklist will be scored based on three possible answers; yes, partial, and no. If any of the checks are not relevant to any articles, then it will be excluded from the evaluation of that particular study only. Those studies that will not provide the basic information about their research methods will score less than 50% in quality assessment and will be excluded.

Table 3: Key evaluation dimensions of Motorola assessment tool.

Score	Approach	Deployment	Results
Very poor	(i) Items not evident OR (ii) No ability to implement OR (iii) No management indebtedness of the need OR	(i) No department of the organization can use the items OR (ii) No department of the organization express interest	(i) No change
Poor	(i) Poor organizational skills OR (ii) Poor management recognition of the need OR (iii) Poor organizational commitment	(i) Some department of the organization uses the item OR (ii) Some department of the organization express interest	(i) Ineffective
Weak	(i) Supportive action for the items starts OR (ii) Some departments of the organization can instrument the items OR (iii) Management starts recognizing the needs	(i) Fragmented or inconsistent use of item OR (ii) Implemented in some departments of the organization OR (iii) Item use subject to experimenting in some departments of the organization	(i) Inconsistent results OR (ii) Spotty results OR (iii) Indication of effectiveness only for some departments of the organization
Fair	(i) Process for items putting into practice defined (ii) Numerous supportive actions for the items take place OR (iii) Widespread commitment but no full assurance by management	(i) Item use subject to experimenting in some major departments of the organization OR (ii) Uniformity in use or less fragmented use OR (iii) Used in some major departments of the organization OR	(i) Unpredictable results for other departments of the organization OR (ii) Reliable and positive results for some important departments OR
Marginally Implemented	(i) Supportive items took in place OR (ii) Some managers become preemptive and assure to implement OR (iii) Items implementation across every part of the organization OR	(i) Item use subject to experimenting in several departments of the organization OR (ii) Item used in many departments of the organization OR (iii) Item mostly stable across various departments of the organization	(i) Positive quantifiable results in maximum departments OR (ii) Constantly positive results over time throughout the organization
Implemented	(i) Item established as an indispensable part of the process OR (ii) Most of the administration is proactive OR (iii) Supporting items facilitates and encourages the use of practice OR (iv) The entire management is committed OR	(i) Item use subject to experimenting in nearly every department of the organization OR (ii) Installed in nearly every department of the organization OR (iii) Constant use across nearly every department of the organization	(i) Constantly positive outcomes over time across nearly every department OR (ii) Positive measurable outcomes in nearly all departments
Outstanding	(i) Management provides enthusiastic leadership and commitment OR (ii) Organizational excellence in the practice recognized even outdoor the firm	(i) Item use subject to experimenting in every department of the organization OR (ii) Consistent use with the passage of time throughout the organization OR (iii) Consistent setup in every part of the organization	(i) Requirement exceeded OR (ii) Guidance sought by others OR (iii) Constantly world-class outcomes

Data extraction: Primary reviewers will independently review all papers. In the case of needs for guidance, the secondary reviewers will be approached.

Execution of the review: The review was conducted from June 2017 to September 2018. To support the protocol trial, we used the EndNote X7 tool for the control group of primary studies and paper selection and storage. Moreover, Microsoft Access was used to manage the data extraction and quality assessment.

3.2. Empirical Survey. To rank the identified CIFs using FMCDM and at the same time to validate the SLR's findings, a questionnaire survey was carried out in the software development industry. Questionnaire survey serves as an appropriate method of gathering tacit quantitative and qualitative facts [69]. The process of a questionnaire survey can be divided into two phases, designing and sampling. In the design phase, a set of questions are designed to be answered by the sample. In the sampling phase, the investigator discovers, lists, selects, and invites suitable experts to contribute in the questionnaire survey [70]. Both are described briefly in the below subsections.

3.2.1. Implementing Online Questionnaire Survey. Surveys can be developed using either commercial or self-developed tools. We have used Google drive where we typed

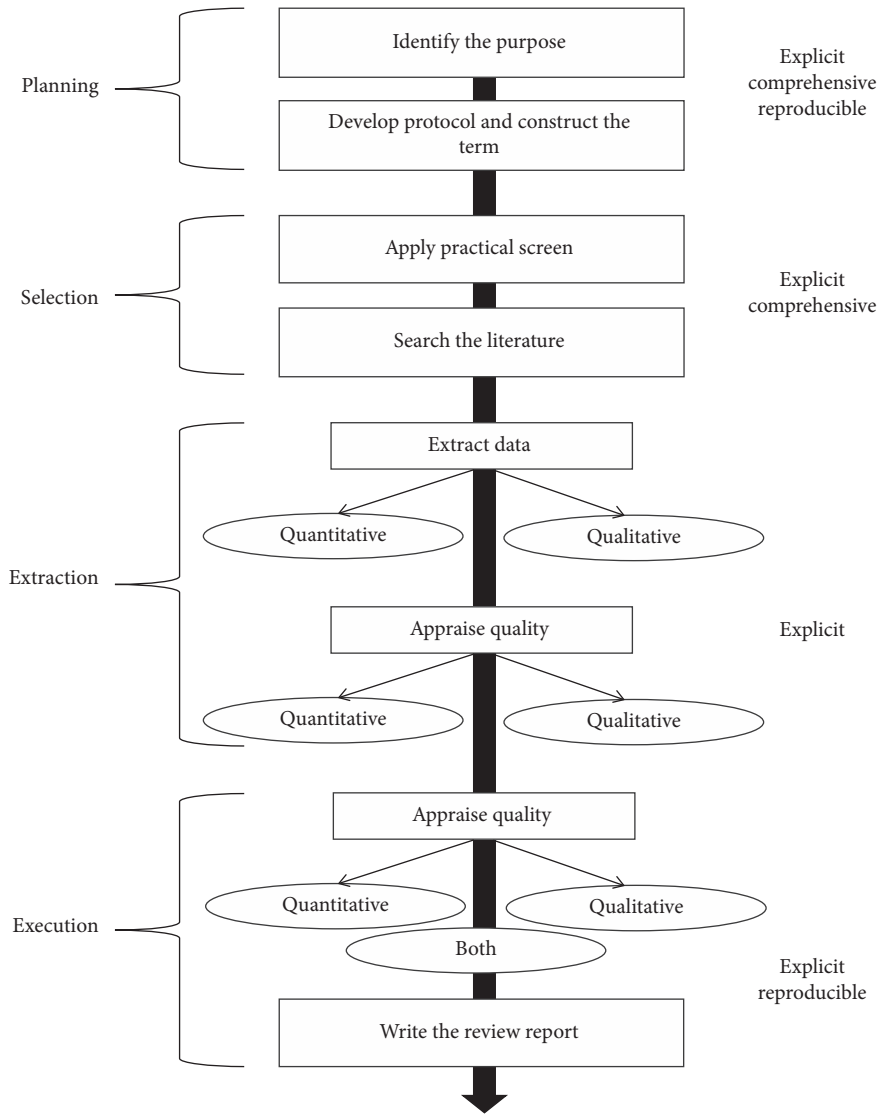


Figure 2: Various stages in the SLR process.

questions and required scripts and web pages came out as results. It is a commercially available online application. It also helps in collecting responses from participants, because it facilitates resending of the link through e-mail in case of no response. The questionnaire design was piloted through six members of our laboratory and necessary changes were made accordingly.

3.2.2. Data Gathering. Survey inquiry is deliberated as a suitable method of gathering tacit qualitative and quantitative data [69]. The questions of the questionnaire are of two types: open-ended, also called subjective, and close-ended, called objective. The subjective questions allow a variety of answers from the respondents, while for objective, only one choice can be chosen from the available choices. This method of data gathering assists in reducing the threat of bias relating to the investigator's prejudices. It encourages the respondent to give her/his view regarding a specific question [69, 70]. Before a questionnaire, each participant was sent a questionnaire invitation letter. This letter outlined the main theme of the questionnaire survey, the expected duration, and measures that could be taken to ensure privacy and confidentiality. We concentrate on obtaining the linguistic weight for the influential factors (items) because not all the factors are equally important for the adoption of CC in connection to software testing.

3.3. Analysis Approach. For analysis purposes, we use the FMCDM framework presented in the next section.

3.3.1. FMCDM Framework for Measuring the Possibility of Cloud Adoption for Software Testing. This paper aims to increase empirical knowledge of CC adoption for software testing. Recently, the use of MCDM approaches has been increasing in information science and software engineering. In this paper, we successfully adopt FMCDM for the evaluation and assessment of CBT in the software industry. For

Table 4: Inclusion criteria/exclusion criteria.

Inclusion criteria	
S. no	
1	Studies that describe factors for cloud-based software testing (CBT)
2	Studies that describe the evaluation of tools, methods, processes, and techniques for CBT.
3	Studies that describe advice/strength/weakness of the CBT framework
4	Studies that describe lessons learned in adopting CBT tools/services
5	Studies that describe techniques/process for migrating software testing to cloud platform
6	Studies that describe criteria for what and when to shift software testing to cloud platform
7	Studies that describe solutions to the challenges in CBT testing domain
8	Studies that evaluate CBT service provider practical capabilities for automatic software testing
Exclusion criteria	
1	Studies written in a language other than English
2	Studies that do not mention practices/solutions/challenges/tools/methods or techniques in automatic software testing in the Cloud
3	Studies conducted in a domain other than software engineering
4	Studies with less than four pages of contents
5	Studies published in more than one sources
6	Studies which get a low score on the quality checklist

Table 5: Quality criteria.

S. no	Quality criteria
1	Is it clear how the factors that influence the adoption are identified?
2	Are the findings supported by extensive data?
3	Are the objectives clearly stated?
4	Are the experimental or observational units adequately described?
5	Is the paper based on based on some standard theory or model?
6	Are negative findings presented?
7	Are the implications clearly stated?
8	Are all the data proposed in the data extraction form possible to extract?
9	If there is a control, what are these variables that may affect study results?
10	Is the study context well defined?
11	Are the testing strategies formally described?
12	Is there a link between interpretation along with conclusion and data?

the stated research model development, we have incorporated a five-phase empirical research methodology using a mix of qualitative and quantitative research techniques for data collection and MCDM approach for model development and evaluation. An FMCDM framework model as shown in Figure 3 was developed to rank and rate the identified alternative criteria by assembling multiple experts in decision-making as deliberated below.

Phase#1: Identification of the Influential Factors via SLR. To find the influential factors for the adoption of CC for CBT, the review protocol was executed to select papers through five distinct stages. In the first stage, after the selection of the databases, the search string was applied to the selected sources and we got 1,650 research articles for further process. The number of returned papers in each database is shown in Table 6. In the second stage, we read the titles and keywords and apply inclusion criteria. In the third stage, we read the abstracts and conclusions and apply exclusion criteria. At stage-3, we also excluded 29 studies that were repeated across different databases. In the fourth phase, we read the full text of the remaining articles and applied quality criteria along with inclusion/exclusion, which results in the studies to be described in the review. After the data extraction and execution stage, we extract a list of quotes from the sample of 136 papers, where each quote described a list of influential factors. The primary reviewers in conjunction with the secondary reviewers reviewed these quotes to classify these quotes into different categories. Initially, 82 categories were identified; these were reviewed and merged into 70 different factors.

3.3.2. Classification of the Identified Factors into Different Dimensions. This study presents the participants with the ease to understand linguistic terms, parameterized by TFNs to precise the individual disagreement or agreement about the importance of identified predictors (main criteria) and factors (subcriteria). We are concerned about finding the ranks because all the criteria (factors and predictors) are not equally important. To rank the predictors, we divided it into small factors called subcriteria.

After data collection through SLR, the identified factors (also called subcriteria in this section) are framed into twelve major variables, ten independent (also called main criteria in this section), one mediator, and one dependent variable. Out of seventy influential factors, eleven were identified to have negative effect on the Cloud adoption, these are term as barriers. The fifty-nine factors are distributed into ten main

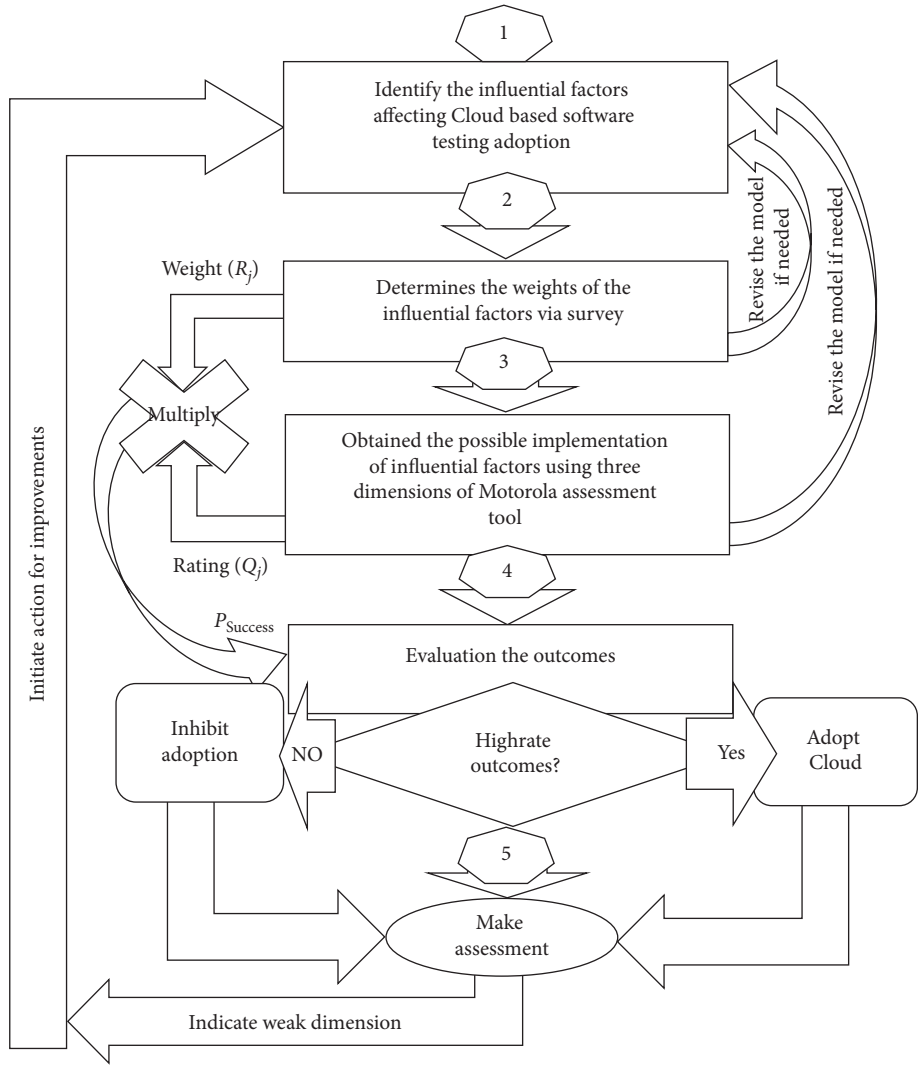


Figure 3: FMCDM model for measuring the possibility of successful Cloud adoption for software testing.

Table 6: Papers included/excluded during different phases.

Search venue	After phase 1	After phase 2	After phase 3	After phase 4
Science direct	177	83	80	14
ACM	123	55	50	17
IEEE xplore	842	250	229	76
Springer link	77	32	21	13
CiteSeer	63	07	06	03
Google scholar	400	58	21	12
Wiley	13	03	03	01
Total	1650	533	464	136

variables that are supposed to positively affect the adoption decision while the 11 barriers collectively form one variable that is supposed to negatively affect adoption decision (Figure 4).

3.3.3. Scale Development and Validation. The questions and the items to measure factors for this study have been borrowed from the preceding research on CC adoption. The scales used to collect the survey data about various predictors and factors used for this study are illustrated in Appendix-A in Supplementary materials. The finest means for assuring content validity is to select items from the earlier confirmed scale [71]. Therefore, in this study, we try to borrow items from preceding research to show generalized insights. To further improve the questionnaire design and contents, the questionnaire was sent to two experts in this field and two experts in academia conducting research in the relevant field. Their comments were incorporated accordingly.

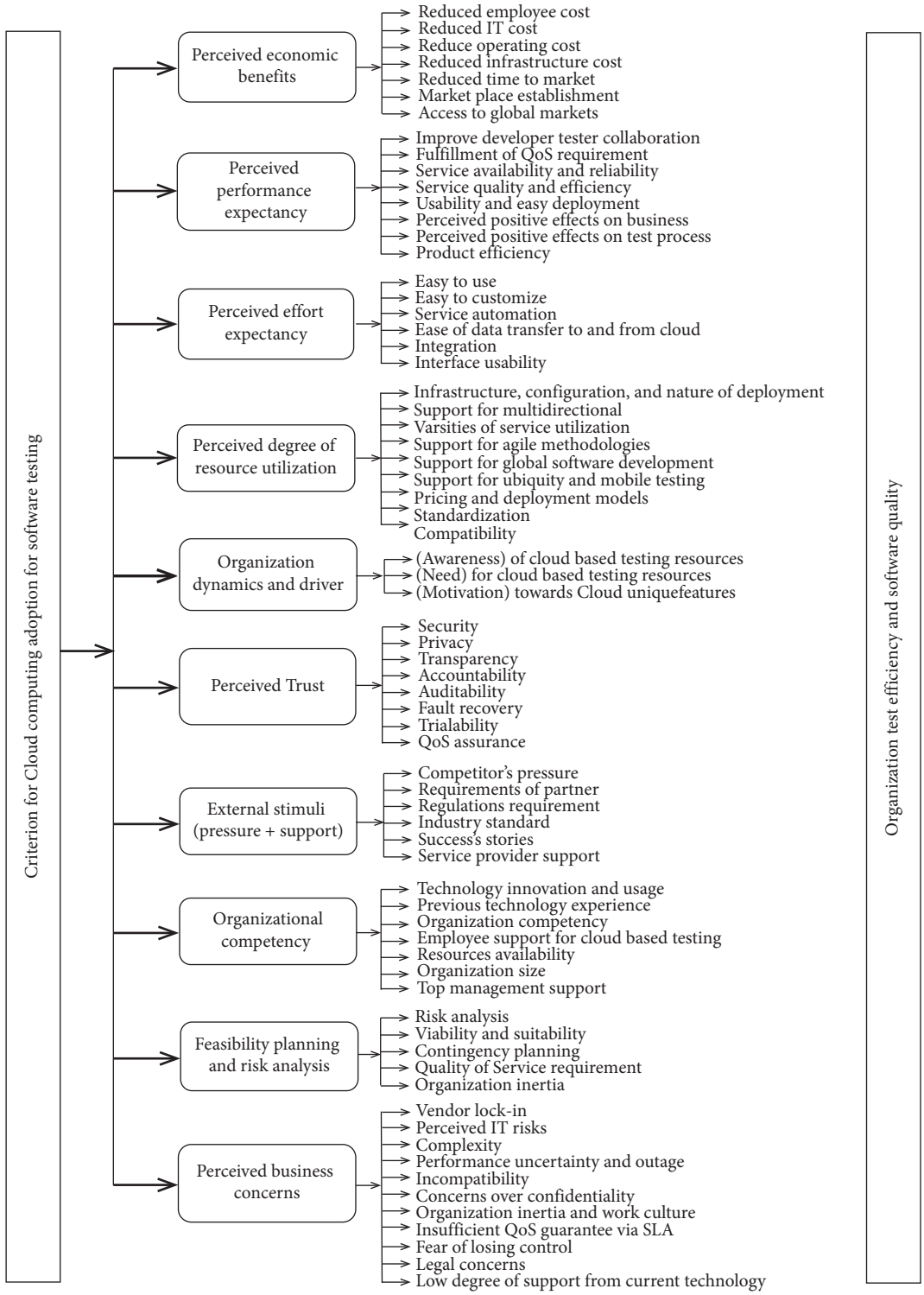


Figure 4: A conceptual model of Cloud adoption: predictors and factors.

Based on the outcomes of SLR, we have identified ten main criteria and seventy subcriteria as illustrated in Figure 4. These criteria were measured through one hundred and thirty-three items in an expert-based questionnaire survey. The items are presented in Appendix-A in Supplementary materials.

Phase#2: Weighting the Influential Factors via Empirical Survey. To rank the influential factors and validate the findings of SLR, an empirical questionnaire survey was performed. Survey investigation is considered an appropriate method for collecting tacit quantitative and qualitative data [69]. SLR results were used as input to the

questionnaire. The importance weights of main and subcriteria are assessed through a linguistic scale denoted by TFNs [59, 60]. In this study, we incorporate the fuzzy scores both in the weighting survey (see Table 1) and implementation case study (Table 2). Specifically, based on TFNs as shown in the second column of Tables 1 and 2, we translated the expert opinions for processing through the FMCDM framework.

3.3.4. Obtaining the Significant Weight of the Influential Factors via FMCDM. The process for computing the significant weights of items is described in the following steps:

Step 1: Generate a decision matrix $\widetilde{A}$ for the significance weights of items ($I_b, b = 1, 2, 3, \ldots, n$). The participants of the survey ($E^a, a = 1, 2, 3, \ldots, m$) are then asked to give their subjective judgment about the significant weight of each item ($\widetilde{\alpha}_b^a$ where I represents item, b represents item's number, and a represents an expert's number) through the linguistic variables as given in Table 1, e.g.,

$$\widetilde{A} = \begin{array}{c} \\ I_1 \\ I_2 \\ I_3 \\ \vdots \\ I_n \end{array} \begin{array}{c} \begin{array}{cccccc} E^1 & E^2 & E^3 & \cdots & E^m \end{array} \\ \begin{bmatrix} \widetilde{\alpha}_1^1 & \widetilde{\alpha}_1^2 & \widetilde{\alpha}_1^3 & \cdots & \widetilde{\alpha}_1^m \\ \widetilde{\alpha}_2^1 & \widetilde{\alpha}_2^2 & \widetilde{\alpha}_2^3 & \cdots & \widetilde{\alpha}_2^m \\ \widetilde{\alpha}_3^1 & \widetilde{\alpha}_3^2 & \widetilde{\alpha}_3^3 & \cdots & \widetilde{\alpha}_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \widetilde{\alpha}_n^1 & \widetilde{\alpha}_n^2 & \widetilde{\alpha}_n^3 & \cdots & \widetilde{\alpha}_n^m \end{bmatrix} \end{array}, \quad a = 1, 2, \ldots, m; b = 1, 2, \ldots, n. \tag{4}$$

where m represents the total number of experts and n represents the total number of items and, $\widetilde{\alpha}_b^a = (l\widetilde{\alpha}_b^a, m\widetilde{\alpha}_b^a, u\widetilde{\alpha}_b^a)$ represents the fuzzy weight of the bth items given by ath respondents.

Step 2: As the independent assessment of each expert, vary according to his or her, perception, knowledge, role, and experience of the subject matter. Thus, we calculate the average score to cumulative the fuzzy importance of each item by m experts.

$$\widetilde{\omega}_b = \frac{1}{m}\left[\sum_{a=1}^{m} \widetilde{\alpha}_b^a\right], \tag{5}$$

where $\widetilde{\omega}_b = (l\omega_b, m\omega_b, u\omega_b)$ represents the cumulative fuzzy importance weight of the bth item.

Step 3. The cumulative fuzzy importance $\widetilde{\omega}_b$ is used to calculate the best nonfuzzy performance (BNF) value BNP_{W_b}. BNP_{W_b} can be obtained using

$$\mathrm{BNP}_{W_b} = \frac{[(uw_b - lw_b) + (mw_b - lw_b)]}{3} + lw_b, \tag{6}$$

where BNP_{W_b} denotes the best nonfuzzy performance (BNP) score for the TFN $\widetilde{\omega}_b$ while W_b is the significant weight of the bth item in the crisp numeral format. The local ranks will be obtained through W_b.

Step 4. After the defuzzification of TFNs in step 4, crisp numbers are calculated and normalized through (7) while the items global ranks R_b will be obtained through equation (7):

$$R_b = \frac{W_b}{\sum_{b=1}^{n} W_b}, \tag{7}$$

where R_b denotes the normalized importance of the bth item such that $\sum_{b=1}^{n} R_b = 1$.

Step 5. Since each item measures different aspects of the subcriteria, we incorporate the average score method to cumulative the best fuzzy importance W_f of each subcriteria (factors).

$$W_f = \frac{\sum_{b=1}^{k} W_b}{k}, \tag{8}$$

where W_b represents the BNP weight of each item in the particular factor (subcriteria), k represents the total number of items measuring that factors while $\sum_{b=1}^{k} W_b$ is the aggregate BNP weight of all the items in that factor.

3.3.5. Computing the Significant Weight of the Predictor via FMCDM

Step 6. Using W_b (BNP weight of items), we can obtain W_p, BNP weight of predictor using

$$W_p = \sum_{b=1}^{h} W_b, \tag{9}$$

where W_b represents the BNP weight of each item in the particular predictor, while h represents the total number of items in that predictor. We also obtain and normalize the crisp number for each predictor and each factor within predictor using equations (10) and (11):

$$R_f = \frac{W_f}{\sum_{f=1}^{h} W_f}, \tag{10}$$

where R_f represents the normalized significance weight of the bth factor such that $\sum_{f=1}^{h} R_f = 1$.

$$R_p = \frac{W_f}{\sum_{b=1}^{n} W_b}, \tag{11}$$

where $\sum_{b=1}^{n} W_b = W_t$ sum of the weight of all items and R_p represents the normalized significance weight of the bth predictor such that $\sum_{b=1}^{n} R_b = 1$.

Phase#3: Rating of the Influential Factors via Case Study. To assess the CTAAM through MCDM approach and at the same time evaluate its effectiveness in real-world

environment, two case studies were executed. Case study can provide adequate evaluation in real-world industrial environments [72]. Similar to the questionnaire survey, here we also translate the expert's judgment to TFNs format but unlike them, we use the extended version of the Motorola evaluation tool [21] to evaluate the SDO readiness towards Cloud adoption for CBT. For the assessment of influential factors and to find the implementation of various dimensions in the assessment case study, we also incorporate the fuzzy set theory. Seven linguistic scales as illustrated in Table 2 based on the dimension of Motorola evaluation tool design for case companies to rate factors across the three dimensions of Motorola evaluation tool [73] such as approach, deployment, and results as presented in Table 3.

3.3.6. Obtaining the Readiness of the Software Testing Organizations for Cloud Adoption via FMCDM. The procedures for calculating the degree of implementation of the items in particular organizations are described in the following steps:

(i) Step 1: Generate three matrices $\widetilde{B}_A$, $\widetilde{B}_D$, and $\widetilde{B}_R$ for the rate of implementation of the items ($I_b, b = 1, 2, 3, \ldots, n$). A, D, and R denotes the three dimensions of the Motorola evaluation tool as presented in Table 3. The respondents of the survey ($E^a, a = 1, 2, 3, \ldots, m$) are asked to give their expert judgment about the degree of implementation of the items in their respective organizations following the guidelines of Motorola evaluation tool as illustrated in Table 3, by using the linguistic scale as presented in Table 2, e.g.,

$$\widetilde{B}_A = \begin{array}{c} \\ I_1 \\ I_2 \\ I_3 \\ \vdots \\ I_n \end{array} \begin{array}{c} \begin{array}{cccccc} E^1 & E^2 & E^3 & \cdots & E^m \end{array} \\ \begin{bmatrix} \widetilde{\beta}_1^1 & \widetilde{\beta}_1^2 & \widetilde{\beta}_1^3 & \cdots & \widetilde{\beta}_1^m \\ \widetilde{\beta}_2^1 & \widetilde{\beta}_2^2 & \widetilde{\beta}_2^3 & \cdots & \widetilde{\beta}_2^m \\ \widetilde{\beta}_3^1 & \widetilde{\beta}_3^2 & \widetilde{\beta}_3^3 & \cdots & \widetilde{\beta}_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \widetilde{\beta}_n^1 & \widetilde{\beta}_n^2 & \widetilde{\beta}_n^3 & \cdots & \widetilde{\beta}_n^m \end{bmatrix} \end{array}, \quad a = 1, 2, \ldots, m;\ b = 1, 2, \ldots, n,$$

$$\widetilde{B}_D = \begin{array}{c} \\ I_1 \\ I_2 \\ I_3 \\ \vdots \\ I_n \end{array} \begin{array}{c} \begin{array}{cccccc} E^1 & E^2 & E^3 & \cdots & E^m \end{array} \\ \begin{bmatrix} \widetilde{\beta}_1^1 & \widetilde{\beta}_1^2 & \widetilde{\beta}_1^3 & \cdots & \widetilde{\beta}_1^m \\ \widetilde{\beta}_2^1 & \widetilde{\beta}_2^2 & \widetilde{\beta}_2^3 & \cdots & \widetilde{\beta}_2^m \\ \widetilde{\beta}_3^1 & \widetilde{\beta}_3^2 & \widetilde{\beta}_3^3 & \cdots & \widetilde{\beta}_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \widetilde{\beta}_n^1 & \widetilde{\beta}_n^2 & \widetilde{\beta}_n^3 & \cdots & \widetilde{\beta}_n^m \end{bmatrix} \end{array}, \quad a = 1, 2, \ldots, m;\ b = 1, 2, \ldots, n,$$

$$\widetilde{B}_R = \begin{array}{c} \\ I_1 \\ I_2 \\ I_3 \\ \vdots \\ I_n \end{array} \begin{array}{c} \begin{array}{cccccc} E^1 & E^2 & E^3 & \cdots & E^m \end{array} \\ \begin{bmatrix} \widetilde{\beta}_1^1 & \widetilde{\beta}_1^2 & \widetilde{\beta}_1^3 & \cdots & \widetilde{\beta}_1^m \\ \widetilde{\beta}_2^1 & \widetilde{\beta}_2^2 & \widetilde{\beta}_2^3 & \cdots & \widetilde{\beta}_2^m \\ \widetilde{\beta}_3^1 & \widetilde{\beta}_3^2 & \widetilde{\beta}_3^3 & \cdots & \widetilde{\beta}_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \widetilde{\beta}_n^1 & \widetilde{\beta}_n^2 & \widetilde{\beta}_n^3 & \cdots & \widetilde{\beta}_n^m \end{bmatrix} \end{array}, \quad a = 1, 2, \ldots, m;\ b = 1, 2, \ldots, n, \tag{12}$$

where n denotes the total number of items and m denotes the total number of respondents and $\widetilde{\beta}_b^a = (l\beta_b^a, m\beta_b^a, u\beta_b^a)$ denotes the fuzzy implementation score of the items provided by the ath respondent for the bth item. After obtaining the three-dimensional assessment score ($\widetilde{B}_A, \widetilde{B}_D, \widetilde{B}_R$), we calculate the average score $\widetilde{B}_M$ using

$$\widetilde{B}_M = \frac{\widetilde{B}_A + \widetilde{B}_D + \widetilde{B}_R}{3}, \tag{13}$$

where M represents mean, one example is given in Table 7.

Step 2: As each expert judgment for the same items might be different according to their role, expertise, understanding, and perception of the subject matter. Therefore, we used the average score approach to get the cumulative fuzzy implementation score of each item by m experts by equation (14):

$$\widetilde{q}_b = \frac{1}{m}\left[\sum_{a=1}^{m} \widetilde{\beta}_b^a\right], \tag{14}$$

where $\widetilde{q}_b = (lq_b, mq_b, uq_b)$ denotes the aggregate fuzzy importance of the bth item.

Step 3. The accumulated TFN $\widetilde{q}_b$ is obtained to calculate the BNP score. BNP_{Q_b} can be obtained using equation (15):

$$\mathrm{BNP}_{Q_b} = \frac{[(uq_b - lq_b) + (mq_b - lq_b)]}{3} + lq_b, \tag{15}$$

where BNP_{Q_b} denotes the BNP value for the TFN $\widetilde{q}_b$ and Q_b denotes the rating for the bth item in the crisp number format.

Step 4. Since each item measures different aspects of the subcriteria, we calculate the average score to aggregate the best fuzzy importance Q_f of each subcriterion (factor).

$$Q_f = \frac{\sum_{b=1}^{k} Q_b}{k}, \tag{16}$$

where Q_b represents the BNP implementation of the individual items in the corresponding factors (subcriteria), k represents the total number of items in that factor while $\sum_{b=1}^{k} Q_b$ is the sum of the BNP weight of the all items in that factor.

Phase#4: Evaluation via FMCDM. Once we came up with ranking and rating, then it is easy to estimate the possibility of success. The success value will be evaluated with predefined standard, i.e., ≥70% for adoption. The process is explained as follows.

3.3.7. Determining the Possibility of Successful Cloud Adoption for Software Testing. Once we came up with R_b and R_f the weights of the items and factors (subcriteria) and Q_b and Q_f the implementation of the items and factors in

Table 7: An example of the predictor (main criteria) ranking using factor (subcriteria).

Expert	An example of factors ranking (subcriteria)									
	M2S1	M2S2	M2S3	M2S4	M2S5	M2S6 (1)	M2S6 (2)	M2S7 (1)	M2S7 (2)	M2S8
E1	(0.1, 0.3, 0.5)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
E2	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)
E3	(0.1, 0.3, 0.5)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
E4	(0.1, 0.3, 0.5)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)
E5	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.9, 1.0, 1.0)
E6	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)
E7	(0.1, 0.3, 0.5)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)
E8	(0.0, 0.0, 0.1)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
E9	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)
E10	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)
E11	(0.1, 0.3, 0.5)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.9, 1.0, 1.0)	(0.3, 0.5, 0.7)	(0.9, 1.0, 1.0)
E12	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)
E13	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)
E14	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)
E15	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)
E16	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
E17	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
E18	(0.7, 0.9, 1.0)	(0.1, 0.3, 0.5)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)
E19	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)
E20	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
E21	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
E22	(0.1, 0.3, 0.5)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)
E23	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)
E24	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)
E25	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)
E26	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)
E27	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)
E28	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
E29	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)
E30	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
E31	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)
E32	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.1, 0.3, 0.5)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)
E33	(0.5, 0.7, 0.9)	(0.1, 0.3, 0.5)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)
E34	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)
E35	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)
E36	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
E37	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)
E38	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)
E39	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
E40	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)
E41	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)
E42	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.1, 0.3, 0.5)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)

Table 7: Continued.

Expert	An example of factors ranking (subcriteria)									
	M2S1	M2S2	M2S3	M2S4	M2S5	M2S6 (1)	M2S6 (2)	M2S7 (1)	M2S7 (2)	M2S8
E43	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)
E44	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)
E45	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.9, 1.0, 1.0)
E46	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)
E47	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)
E48	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.1, 0.3, 0.5)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)
E49	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)
E50	(0.1, 0.3, 0.5)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)
E51	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.9, 1.0, 1.0)	(0.1, 0.3, 0.5)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)
E52	(0.5, 0.7, 0.9)	(0.9, 1.0, 1.0)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)	(0.9, 1.0, 1.0)
E53	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)	(0.7, 0.9, 1.0)	(0.9, 1.0, 1.0)
wj	(0.41, 0.60, 0.79)	(0.57, 0.76, 0.92)	(0.48, 0.68, 0.86)	(0.54, 0.73, 0.88)	(0.50, 0.70, 0.86)	(0.53, 0.71, 0.86)	(0.62, 0.81, 0.95)	(0.65, 0.82, 0.95)	(0.50, 0.69, 0.85)	(0.65, 0.80, 0.89)
W_b	**0.599916**	**0.747017**	**0.675099**	**0.713857**	**0.685898**	**0.701087**	**0.792826**	**0.804194**	**0.680771**	**0.777891**
Avg	**0.599916**	**0.747017**	**0.675099**	**0.713857**	**0.685898**	**0.746957**		**0.742482**		**0.777891**

the respective organizations, then it is easy to estimate the possibility of success P_{sucess} based on items using equation (17) as well as based on factors by equation (18)

$$P_{\text{sucess}} = R_b \times Q_b, \tag{17}$$

$$P_{\text{sucess}} = R_f \times Q_f. \tag{18}$$

From the possibility of success, we can easily compute the possibility of failure.

Phase#5: Assessment via FMCDM Using Motorola Assessment Tool. In the result of the evaluation, in case of a low outcome (less than 70%), the model will pinpoint weak influential factors and predictors for further improvements while in case of high rate outcomes it will direct adoption and will signpost towards areas for enhancements.

3.3.8. Identify the Weak Area for Future Enhancement. The BNP weights of items are used to compute the BNP weights of factors and predictors. If I_f represents the implementation of factor and I_p represents the implementation of predictor, then it can be found using (19) and (20), respectively:

$$I_f = \frac{Q_f}{k}, \tag{19}$$

where k represents the total number of items in that factor $Q_f = \sum_{b=1}^{k} Q_b$ is the aggregate BNP implementation of all items implements that factor.

$$I_p = \frac{Q_p}{h}, \tag{20}$$

where h represents the total number of items in that predictor and $Q_p = \sum_{b=1}^{h} Q_b$ is the aggregate BNP implementation of all the items that measure that predictor.

4. Results and Analysis

This section demonstrates the outcomes of our study. This study proposes a technology acceptance model for CBT, based on the factors identified through the SLR and questionnaire survey, using the codified knowledge of experts and an intelligence technique for decision-making regarding CBT. The outcomes presented here are obtained based on the analysis of collected data in the first phase, i.e., ranking and rating of these data through the industrial survey and case study using FMCDM.

4.1. Results Obtained via Ranking the Criteria through an Empirical Survey. This section presents the results obtained by employing FMCDM techniques to analyze the data collected through SLR. We have identified 133 items to measure the ten predictors called the main criteria using 70 subfactors via FMCDM.

4.1.1. Weighs of the Factors (subcriteria) Using FMCDM. One hundred thirty-three items as shown in Appendix-A in Supplementary materials, were used in our scale to rank predictors and factors through a questionnaire survey. Ninety-five experts participate in the weighting process through a questionnaire survey. The weight calculation processes are as follows:

(1) These experts are asked to give their subjective judgment about the items using the linguistic scale presented in Table 1.

(2) The linguistic evaluations are then transcribed into the corresponding TFN as shown in Table 7 while taking predictor M2: "perceived performance expectancy" as an example.

(3) Because the perceptions of each expert are different due to their role, industrial experience, qualification, etc., equation (5) is used to get the synthesized aggregate TFN ($\widetilde{\omega}_b$) as listed in Supplementary Table 2 column 4.

(4) Then, defuzzification of the TFN is carried out to obtain BNP in a crisp format using equation (6). The outcomes are shown in Supplementary Table 2 (BNP_WJ). The BNP value is used in the further calculation for global ranking R_b and local ranking as shown in Supplementary Table 2 column 7.

(5) The crisp number obtained in step 4 is normalized and the normalized importance R_b of items are obtained by using equation (7) which are further used to find an overall rank for each item. The outcomes are presented in Supplementary Table 2 column 8 (second last column). R_b is used to obtain a global rank of each item as shown in Supplementary Table 2 column 9 (last column). As some factors and barriers are measured using more than one item, the individual item scores of a factor in BNF format are then converted to an average score in the same BNF format using equation (8) as shown in Supplementary Table 2 (last row). Referring to equation (8), the BNP weight of the factor (subcriteria) W_f is an average of the BNP weight of all the items in that respective factor.

Using W_f, we obtained the local rank of the factors (subcriteria) within predictors (main criteria) as shown in Supplementary Table 2 second last column (W_f).

4.1.2. Weights of the Predictor (Main Criteria) by FMCDM. Referring to equations (8) and (9), BNP weight of the factor (subcriteria), W_f is the average of the BNP weight of all the items in that respective factor while the BNP weight of the predictor and W_p is the sum of the BNP weights of all the items in that predictor. We first calculate W_f by using equation (8), then we find the importance (rank) of factor within predictor (local weight) using (W_f). The overall rank R_f of a factor was obtained by dividing W_f by sum of the weight of all factors, i.e., $\sum_{f=1}^{n} W_f$ refers to equation (10), which are 50.2752 in our study as given in Supplementary Table 2 (last row). For ranking predictors, we used the weight of the normalized importance of each predictor as calculated by equation (11). Ranks of the predictors are presented in

Table 7, where R_p is the normalized weight of the predictor and $W_t = \sum_{b=1}^{n} W_b$ shows the total weight, W_t is the sum of the BNP weight W_b of all the items and is accessible from Supplementary Table 2 (last row).

4.2. Results Obtained via Rating of the Influential Factors. To obtain the actual level of implementation of the influential factors using different items in the SDO organizations, a case study is carried out. The case study is a suitable method for providing enough evidence in the real-world industrial environment.

To be more certain and confident in our assessment, two distinct case studies were carried out at two distinct SDO companies. Companies are chosen because they provided particularly detailed descriptions of different SDO activities. If an item or factor has a strong implementation in the company, then the likelihood of success in the adoption of Cloud for future projects increases. The possible implementation of factor (subcriteria) concerning to each predictor (main-criteria) is calculated as follows:

Step 1: The participants in the case study were requested to provide their independent views about the extent of implementation of each item in their respective organizations from the three dimensions of Motorola by choosing linguistic terms shown in Supplementary Table 3 incorporating the Motorola guidelines given in Table 3.

Step 2: The linguistic terms are then transcribed into corresponding TFN. An example of M2 (PPE): "perceived performance expectancy" are shown in Table 8.

Step 3: Three-dimensional scores in TFN format are then converted to an average score in the same TFN format using equation (13) as shown in Table 8.

Step 4: To aggregate the subjective judgments of the participants on the assessment of factors (because the perception of each expert is different due to their role, experiences, and education level, etc.). Equation (14) is used to get the synthesized TFN as listed in Table 8 (last row).

Step 5: Then defuzzification of the TFN is carried out to obtain BNP in the crisp format using equation (15) as shown in Supplementary Table 3.

4.3. Determining the Success Possibility of Cloud Adoption. Once we have an importance weight R_b and possible implementation Q_j of items, then it is easy to calculate the possibility of successful adoption using (19). The possibility of successful adoption for Company-A is shown in Table 9 (last column) while that of Company-B is shown in Table 10. The overall success value is equal to the sum of the success value of all items. Success value 0.5 indicates a 50% chance of both success and failure. Once we get success value, then the possibility of failure can be calculated using the common probability method.

4.4. Results Obtained through Evaluation of Predictors in the Case Organization. The implementation for company-A is given in Table 9. It is clear from Table 9 that company-A has not implemented any of the predictors (main-criteria). Furthermore, P_{Sucess} is less than 70%. Our model did not suggest company-A to adopt Cloud at this time because there is a 50% chance of both success and failure. To increase the chances of adoption, Company-A must focus on the perceived business concerns which are not implemented. Similarly, based on the results presented in Table 10 Company-B, has implemented seven predictors; however, P_{Sucess} is still less than 70%. Based on the implementation score, Company-B is relatively more secure if moving to the Cloud environment. For complete implementation, Company-B must focus on the weak factors under perceived business concerns (M10), which will further increase trust and will stimulate Company-B to adopt the Cloud.

5. Summary and Discussion

The objective of this study is to prioritize the predictors as well as to rank the overall subcriteria. Firstly, the ten main criteria are ranked as CC adoption "Feasibility planning and risk analysis, 0.11"; "Perceived degree of Cloud resource utilization, 0.1168"; "Perceived effort expectancy, 0.1141"; "Organization competency and capacity, 0.1128"; "Perceived degree of trust, 0.111"; "Perceived business concerns, 0.1020"; "Organizational dynamics and business drivers, 0.0999"; "External stimuli, 0.0924"; "Perceived performance expectancy, 0.0745"; and "Perceived economic benefits of Cloud adoption, 0.057."

According to the rank of predictors, we find that feasibility planning and risk analysis, perceived degree of Cloud resource utilization, and perceived effort expectancy are the top three predictors of Cloud adoption in the context of software testing. It means, currently testing in the Cloud does not apply to all types of software. Furthermore, most of the testing techniques cannot be applied in the Cloud environment. Additionally, currently testing in the Cloud is not fully automated and testers are left with the laborious tasks of writing scripts and configuring the test environment. Utilization of Cloud resources is also limited due to the viability and suitability of different types of software to be tested and types of tests performed in the Cloud environment. Furthermore, human testers are still needed to conduct testing. This might be the reason that "perceived economic" benefit of Cloud adoption has low weight and ranks low. Besides, perceived performance expectancy is the second least significant predictor in our study. This might be the reason that the adoption of CC in the domain of software testing is very slow compared with that in other fields. The following studies support the results:

(i) The first step in moving to the Cloud is the deep understanding of CC resources and how they relate to their needs. It is also imperative that SDO realizes and analyzes the CC value-added features and goes on only if there is a growing need for this approach [56].

Table 8: Corresponding TFNs of evaluation in case organization taking "Perceive performance expectancy" is an example.

M2 (PPE)		Perceived performance expectancy									
Expert	Dimension	Factors/items									
		M2S1	M2S2	M2S3	M2S4	M2S5	M2S6 (1)	M2S6 (2)	M2S7 (1)	M2S7 (2)	M2S8
E1	A	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
	D	(0.1, 0.3, 0.5)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.1, 0.3, 0.5)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
	R	(0.1, 0.3, 0.5)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.1, 0.3, 0.5)	(0.0, 0.1, 0.3)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.1, 0.3, 0.5)
	M	**(0.2, 0.4, 0.6)**	**(0.5, 0.7, 0.9)**	**(0.4, 0.6, 0.8)**	**(0.2, 0.4, 0.6)**	**(0.1, 0.3, 0.5)**	**(0.3, 0.5, 0.7)**	**(0.4, 0.6, 0.8)**	**(0.4, 0.6, 0.8)**	**(0.3, 0.5, 0.7)**	**(0.2, 0.4, 0.6)**
E2	A	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)
	D	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.1, 0.3, 0.5)	(0.3, 0.5, 0.7)
	R	(0.1, 0.3, 0.5)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.1, 0.3, 0.5)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.1, 0.3, 0.5)	(0.1, 0.3, 0.5)
	M	**(0.4, 0.6, 0.8)**	**(0.5, 0.7, 0.9)**	**(0.4, 0.6, 0.8)**	**(0.3, 0.5, 0.7)**	**(0.2, 0.4, 0.6)**	**(0.3, 0.5, 0.7)**	**(0.4, 0.6, 0.8)**	**(0.6, 0.8, 0.9)**	**(0.4, 0.6, 0.8)**	**(0.3, 0.5, 0.7)**
E3	A	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)
	D	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)
	R	(0.1, 0.3, 0.5)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.1, 0.3, 0.5)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
	M	**(0.3, 0.5, 0.7)**	**(0.6, 0.8, 0.9)**	**(0.5, 0.7, 0.9)**	**(0.3, 0.5, 0.7)**	**(0.5, 0.7, 0.9)**	**(0.6, 0.8, 0.9)**	**(0.6, 0.8, 0.9)**	**(0.4, 0.6, 0.8)**	**(0.4, 0.6, 0.8)**	**(0.5, 0.7, 0.9)**
E4	A	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)
	D	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
	R	(0.1, 0.3, 0.5)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.0, 0.0, 0.1)
	M	**(0.3, 0.5, 0.7)**	**(0.6, 0.8, 0.9)**	**(0.7, 0.9, 1.0)**	**(0.5, 0.7, 0.9)**	**(0.4, 0.6, 0.8)**	**(0.4, 0.6, 0.8)**	**(0.4, 0.6, 0.8)**	**(0.5, 0.7, 0.9)**	**(0.4, 0.6, 0.8)**	**(0.2, 0.3, 0.5)**
E5	A	(0.0, 0.1, 0.3)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
	D	(0.0, 0.0, 0.1)	(0.0, 0.1, 0.3)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.1, 0.3, 0.5)
	R	(0.0, 0.0, 0.1)	(0.0, 0.1, 0.3)	(0.3, 0.5, 0.7)	(0.1, 0.3, 0.5)	(0.3, 0.5, 0.7)	(0.1, 0.3, 0.5)	(0.1, 0.3, 0.5)	(0.3, 0.5, 0.7)	(0.1, 0.3, 0.5)	(0.0, 0.1, 0.3)
	M	**(0.0, 0.0, 0.2)**	**(0.2, 0.4, 0.6)**	**(0.5, 0.7, 0.9)**	**(0.2, 0.4, 0.6)**	**(0.3, 0.5, 0.7)**	**(0.2, 0.4, 0.6)**	**(0.2, 0.4, 0.6)**	**(0.4, 0.6, 0.8)**	**(0.2, 0.4, 0.6)**	**(0.1, 0.3, 0.5)**
E6	A	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
	D	(0.1, 0.3, 0.5)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.1, 0.3, 0.5)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)
	R	(0.1, 0.3, 0.5)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.1, 0.3, 0.5)	(0.5, 0.7, 0.9)	(0.1, 0.3, 0.5)	(0.1, 0.3, 0.5)	(0.0, 0.0, 0.1)
	M	**(0.2, 0.4, 0.6)**	**(0.4, 0.6, 0.8)**	**(0.4, 0.6, 0.8)**	**(0.3, 0.5, 0.7)**	**(0.4, 0.6, 0.8)**	**(0.2, 0.4, 0.6)**	**(0.5, 0.7, 0.9)**	**(0.3, 0.5, 0.7)**	**(0.4, 0.6, 0.8)**	**(0.2, 0.3, 0.5)**
E7	A	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)
	D	(0.0, 0.0, 0.1)	(0.1, 0.3, 0.5)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.7, 0.9, 1.0)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)
	R	(0.0, 0.0, 0.1)	(0.1, 0.3, 0.5)	(0.5, 0.7, 0.9)	(0.1, 0.3, 0.5)	(0.5, 0.7, 0.9)	(0.5, 0.7, 0.9)	(0.3, 0.5, 0.7)	(0.1, 0.3, 0.5)	(0.3, 0.5, 0.7)	(0.1, 0.3, 0.5)
	M	**(0.1, 0.2, 0.3)**	**(0.2, 0.4, 0.6)**	**(0.5, 0.7, 0.9)**	**(0.3, 0.5, 0.7)**	**(0.6, 0.8, 1.0)**	**(0.6, 0.8, 0.9)**	**(0.4, 0.6, 0.8)**	**(0.3, 0.5, 0.7)**	**(0.4, 0.6, 0.8)**	**(0.3, 0.5, 0.7)**
$\tilde{q}_b = (l\tilde{q}_b, m\tilde{q}_b, u\tilde{q}_j)$		**(0.18, 0.34, 0.52)**	**(0.40, 0.60, 0.80)**	**(0.49, 0.69, 0.87)**	**(0.31, 0.51, 0.70)**	**(0.37, 0.57, 0.76)**	**(0.37, 0.57, 0.76)**	**(0.41, 0.61, 0.80)**	**(0.41, 0.61, 0.81)**	**(0.31, 0.51, 0.71)**	**(0.27, 0.44, 0.63)**

(ii) To find an answer to the question "when to migrate testing to the CC platform?" one must check the characteristics of SUT, perceived support of the testing methods in the CC environment, preferences, nature, and dependency of test execution [74]. If the majority of these subfactors cannot operate in the Cloud environment, then Cloud adoption is considered feasible; otherwise, a decision to stop the migration can be made [74].

(iii) Before going to adopt the Cloud testing service, you have to look at ways through which testing is conducted and testing services the service providers offer, in order to plan best to maximum utilize Cloud platform for their testing [25].

(iv) TaaS can be effusively embraced by the industry if it supports more types of testing such as agile development via continuous regression testing [56].

(v) Perceived ease of use contributes significantly towards a willingness to adopt CC services [45]. The outcomes show that convenience and ease of use are the key motivators for Cloud services adoption [45].

(vi) Automation could encourage the adoption of a service. On the contrary, if a service requires tremendous human effort, this might negatively affect adoption [75].

(vii) Despite the great importance of Cloud-based testing, it is extremely complex and at present lacks proper tools to support it [76].

(viii) Extra efforts are required to deploy application and setup test environment in the Cloud [77].

(ix) For simple test cases, the use of CC might increase the task cost and system complexity [78].

(x) Automation encourages the adoption of the CBT technique. On the contrary, if a tool or service requires huge efforts, this might limit its adoption [75].

(xi) To improve quality and reduce the cost of test automation is the most important means [79].

(xii) According to a topical survey [80], presently 28 % of test cases are automated whereas the executives demand to raise this amount shortly. Employing test automation mainstream in the software industry, more and more companies will adopt CBT.

(xiii) The authors of [56] suggest that future work on TaaS needs to focus on entirely automatic test cases. They examine the currently available tools and services and find that only BlazeMeter can offer fully automated performance and load testing [56].

The ranks of the subcriteria in connection to "CC adoption" for software testing are given in Supplementary Table 2. Among the 70 subcriteria, top ten criteria are support for multidirectional testing, 0.018547; top management support, 0.018471; interface usability, 0.017637; standardization; 0.016031; reduced employee cost, 0.017257; pricing and deployment models, 0.016830; employee support for CBT, 0.016723; resources availability, 0.016637; perceived IT risks, 0.013693; and service automation, 0.016500.

Ten least significant factors in decreasing order of weights from 61 to 70 are regulation requirements, 0.012336; improve developer-tester collaboration, 0.011930; risk analysis, 0.011784; competitor's pressure, 0.011492; security, 0.011027; incompatibility, 0.008024; performance uncertainty and outage, 0.007983; legal concerns, 0.005898; insufficient QoS guarantee via SLA, 0.004413; and concerns over confidentiality, 0.004020.

Priyadarshinee et al. [71] rank the factors through fuzzy AHP. They identify organizational risks, sharing and collaboration, confidentiality and integrity, and security and privacy as top factors. According to them, security and privacy, sharing and collaboration, confidentiality, and integrity are the top four factors that have a major effect on the adoption of CC. Competitive pressure is ranked in the middle by us (and it is ranked 20th out of 51 by them), but unlike them, we have relatively low weight for security and privacy. The reason might be that testing data is not sensitive; therefore, organizations are not worried about the security of data.

(i) Migrating the testing to the CC environment is anticipated to be a safe decision because the testing will not contain any sensitive business data [2].

(ii) Migration will have a nominal impact on corporate work as usual deeds [26].

(iii) For organizations having no concerns for security, software testing can be one of the suitable areas that can be moved to the Cloud platform [49].

However, the privacy of the personal information and test results still concerns to SDO. Assurance of privacy, disaster recovery, and back up inspire the confidence of the client in Cloud adoption for software testing.

(i) Since testing does not contain sensitive information, the security concerns are related to the privacy of resources and confidentiality of test results [81].

(ii) Before adopting Cloud for software testing, one should decide whether the Cloud is safe for their sensitive information [35].

(iii) An acceptable testing service essentially addresses concerns related to multitenancy as well as to the safety of uploaded material and confidentiality of test results [82].

(iv) All tenants' data in Cloud are stored in a shared infrastructure using a shared single instance, hence making it difficult to ensure data confidentiality [83].

To assess our proposed framework, we have executed two case studies in the SDO planning to adopt Cloud for software testing shortly. At the end of the case study, a focus group session was conducted among the participants to

Table 9: Evaluation of predictor in case organization (Company-A).

Name of predictor	Q_p	# items	I_p	P_{Success}
M01: perceived economic benefits of Cloud adoption	4.7540	9	0.5282	50.3122 approx. 50.3%
M02: perceived performance expectancy	5.4460	10	0.5446	
M03: perceived effort expectancy	8.1333	14	0.5810	
M04: perceived degree of Cloud resource utilization	8.6905	14	0.6208	
M05: organizational dynamics and business drivers	7.4556	12	0.6213	
M06: perceived degree of trust	8.2571	16	0.5161	
M07: external stimuli (pressure + support)	5.0429	12	0.4202	
M08: organizational competency and readiness	8.3413	13	0.6416	
M09: feasibility planning and risk analysis	9.8943	16	0.6184	
M10: perceived business concerns	8.0921	18	0.4496	

Table 10: Evaluation of predictor in case organization (Company-B).

Name of predictor	Q_p	# items	I_p	P_{Success}
M01: perceived economic benefits of Cloud adoption	6.174	09	0.6860	61.26407 approx. 61.3%
M02: perceived performance expectancy	6.996	10	0.6996	
M03: perceived effort expectancy	10.031	14	0.7165	
M04: perceived degree of Cloud resource utilization	10.325	14	0.7375	
M05: organizational dynamics and business drivers	9.4556	12	0.7880	
M06: perceived degree of trust	8.2571	16	0.5161	
M07: external stimuli (pressure + support)	7.0429	12	0.5869	
M08: organizational competency and readiness	9.1930	13	0.7072	
M09: feasibility planning and risk analysis	11.843	16	0.7402	
M10: perceived business concerns	10.921	18	0.6067	

obtain feedback and the model was revised accordingly. Case study method proves the most powerful instrument for validation in empirical software engineering [72].

In this study, the Motorola assessment instrument and the TFN scale are used in a fuzzy environment. Our contribution is the introduction of TFN scores and the updated guidelines for a 7-point Likert scale. The existing Motorola guidelines are suitable for a 5-point Likert scale only; we have added two more rows, one at the start and the other at the end to make it convenient for measurement on a 7-point Likert scale. The same instrument is also used for the evaluation of SOVRM [84], SPIIMM [354], and SOPM [20]. This tool has been used in CMMI to evaluate the organizations' present position comparable to the CMM and to recognize weak areas that require improvements and management attention [73].

Evaluation results regarding factors using the Motorola assessment instrument are presented in Table 3. According to Motorola assessments, factors having an average evaluation or implementation score greater than 0.7 will be considered implemented while scores less than 0.7 will be marked as weakness. For instance, in predictor perceived economic benefits (M1), two factors such as market place establishment (MIS6) and access to global markets (M1S7) are not implemented properly. The reason might be that Company-A, which does not want these two goals to be achieved through CC adoption.

(i) The biggest challenge is to ensure global availability all the time [56].

(ii) Currently, there is no comprehensive model to test the mobile CC applications. Furthermore, they state that the new Cloud testing methods are restricted to the applications with special features and can only support limited types of tests [85].

All factors in perceived performance expectancy (M2) have scores less than 0.7 except better service availability and reliability (M2S3). It means Company-A is optimistic only about Cloud reliability and availability. It can be observed from Table 3 that "improves developer-tester collaboration" (M2S1) gets a very low implementation score, i.e., lower than 35%. It means regarding Company-A, adoption of CC will not improve developer and tester communication to a high level. Additionally, "better usability and easy deployment" gets a score (56%) which is found very high in the other Cloud adoption study such as [45, 86]. The reason might be that as compared with other Cloud services, testing is not fully automated. The following literature proves the claim:

(i) CC adoption is neither feasible and risk-free all the time nor an automatic or easy process [13].

(ii) The use of CC might increase the task cost and system complexity in case of simple tests [78].

(iii) According to a well-established French testing company called Sogeti, merely 28% of the test cases are presently automated, whereas the executives demand to raise this amount soon. Utilizing test automation in the software industry more and more companies will adopt CBT [80].

(iv) Future work on TaaS needs to focus on entirely automatic test cases. They have examined the currently available tools and services and found that

only BlazeMeter can offer fully automated performance and load testing [56].

In connection to the perceived degree of Cloud resource utilization (M4), we got a relatively low implementation of factors "varieties of service utilization" (M4S3), "support for Global software development" (M4S5), and "support for ubiquity and mobile testing" (M4S6).

(i) Not all types of testing are suitable to be performed in the Cloud. For instance, high-volume automated testing, unit testing (particularly large regression suites), and performance testing are ideal choices to be migrated to the Cloud environment [14].

(ii) TaaS can be effusively embraced by the industry if it supports more types of testing such as agile development via continuous regression testing [56].

(iii) The biggest challenge is to ensure global availability all the time [56].

(iv) Currently, there is no comprehensive model to test the Cloud mobile applications. Furthermore, they state that the new Cloud testing methods are restricted to applications with special features and can only be used in some types of tests [85].

6. Conclusion

This study proposes that software testing should be moved to the Cloud. Additionally, this study explores the determinants and predictors of Cloud adoption for software testing. For this purpose, this study explores the technical aspect of software testing in the Cloud through the FMCDM approach and further develops a technology acceptance model. An expert system based on the codified knowledge of an expert and artificial intelligence techniques such as MCDM is applied to help organizations decide Cloud adoption for software testing. The proposed model includes various predictors and determinants that will guide software development organizations (SDO) towards Cloud adoption for software testing. The model is developed using a five-phase analytic approach. Data are collected via SLR and questionnaire-based survey while analysis is made through fuzzy multicriteria decision-making (FMCDM) approach.

The results of this study suggest that Cloud service providers need to provide support to SDO who are presently working in the CBT domain, develop a cooperative environment for the CC adoption, and eliminate any ambiguity regarding this technology. Besides, the CC providers should clarify the advantages of Cloud services over in-house services. Furthermore, they must focus on concerns such as locality and privacy faced by the SDO and resolve those problems by forming an unrelenting relationship with the clients. The Cloud providers have to assure the SDO that their testing data are safe and accessible upon request around the clock. They should increase their reliability to get confidence, faith, and trust of the SDO about their services. A trial chance may be provided to the SDO to try the Cloud services out before its actual implementation. The provider government has to play a significant role in building trust regarding government intrusion among the SDO. The Client needs to implement CBT and it is very off-beamed to think that CBT is meant for larger companies only.

7. Threats to Validity

The empirical survey was based on the results of SLR; this two-phase framework ensures content validity. Construct validity is concerned with measurement scale whether the measurement scales represent the attributes being measured. The attributes of this research study were taken from a considerable amount of previous research [25, 42–48] and experiencing a systematic literature review [20, 87, 88]. The respondents of the survey confirm the relevance of the attributes selected. Further, the inner reliability of survey responses was assessed using Cronbach's alpha coefficient, which is 0.89 (>0.70), which demonstrates the reliability of data and scale. To internal validity, the SLR findings were used as input for the design of the questionnaire.

The empirical study part of this research engaged participants mainly from the Asian countries only. However, to lessen population prejudice, contributors from other countries such as North America were also invited to include diverse perspectives. Fifty experts voluntarily participated in this exploratory study and there were no previous bonds between the participants and researchers. Contributors were informed that their participation is entirely voluntary and they can withdraw at any time during any stage of the survey if they want. However, to ensure external validity and to diminish any possible bias, the ninety-five contributors were chosen from 20 different countries. Besides, most of the participants had worked in a range of small, medium, and large multinational organizations. Although we cannot claim that all the contributors from these 20 countries would agree with us, we believe that they provide a descriptive sample. In an empirical survey-based research, it is hard to obtain a fully representative sample and to deal with them in an entirely objective fashion [89]. To overcome these limitations, only those participants were included who are involved in test automation. The claim/relevant expertise of the participants were verified by inculcating some open-ended questions in the questionnaire, which were difficult to answer by an ordinary tester or manager etc. This situation might create difficulties when contributors' judgments may be inaccurate or factors supposed to have a significant interrelation for adoption may not be significant at all. However, similar to other opinion-based empirical research studies [25, 43–48], we have full confidence that the findings of this research are based on the data that have been collected from the relevant participants who have been involved and have vastly diversified experience in SDO.

Data Availability

The data used to support the study are provided in the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by National Key Research and Development Program of China (No. 2018YFB1003800), National Natural Science Foundation of China (No. 61972414), Beijing Natural Science Foundation (No. 4202066), and Fundamental Research Funds for Central Universities (Nos. 2462020YJRC001 and 2462018YJRC040).

Supplementary Materials

All the necessary data are provided in Supplementary Tables 1 to 3. (*Supplementary Materials*)

References

[1] M. Madan, M. Dave, and A. Tandon, "Challenges in testing of cloud based application," *International Journal of Advanced Research in Computer Science and Electronics Engineering*, vol. 5, no. 1, pp. 28–31, 2016.

[2] M. S. N. Murthy and V. Suma, "Software testing and its scope in CLOUD: a detailed survey,," in *Proceedings of the 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pp. 269–273, Bengaluru, India, February 2017.

[3] O. Taipale, J. Kasurinen, K. Karhu, and K. Smolander, "Trade-off between automated and manual software testing," *International Journal of System Assurance Engineering and Management*, vol. 2, no. 2, pp. 114–125, 2011.

[4] V. Garousi, M. V. Mäntylä, A. Betin-Can, and S. Mirshokraie, "When and what to automate in software testing? A multi-vocal literature review," *Information and Software Technology*, vol. 76, pp. 92–117, 2016.

[5] G. N. Iyer, J. Pasimuthu, and R. Loganathan, "PCTF: an integrated, extensible cloud test framework for testing cloud platforms and applications," in *Proceedings of the 13th International Conference on Quality Software (ICSQ)*, pp. 135–138, Dallas, TX, USA, May 2013.

[6] S. Marston, Z. Li, S. Bandyopadhyay, and A. Ghalsasi, "Cloud computing-the business perspective," in *Proceedings of the 44th Hawaii International Conference on System Sciences*, pp. 1–11, Kauai, HI, USA, January 2011.

[7] M. Armbrust, I. Stoica, M. Zaharia et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[8] J. Luftman, H. S. Zadeh, B. Derksen, M. Santana, E. H. Rigoni, and Z. Huang, "Key information technology and management issues 2011-2012: an international study," *Journal of Information Technology*, vol. 27, no. 3, pp. 198–212, 2012.

[9] L. A. Kappelman, E. R. Mclean, J. N. Luftman, and V. Johnson, "Key issues of IT organizations and their leadership: the 2013 SIM IT trends study," *Mis Quarterly Executive*, vol. 12, no. 4, p. 7, 2013.

[10] A. Elumalai, I. Starikova, and S. Tandon, *IT as a Service: From Build to Consume*, McKinsey and Company, New York, NY, USA, 2018.

[11] J. Avrane-Chopard, T. Bourgault, A. Dubey, and L. Moodley, "Big business in small business: cloud services for SMBs," *Telecommunications, Media, and Technology Practice*, McKinsey and Company, New York, NY, USA, 2014.

[12] J. Gao, X. Bai, W. Tsai, and T. Uehara, "Testing as a service (TaaS) on clouds,," in *Proceedings of the IEEE Seventh International Symposium on Service-Oriented System Engineering (SOSE)*, pp. 212–223, San Francisco Bay, CA, USA, March 2013.

[13] H. Srikanth, M. Cashman, and M. B. Cohen, "Test case prioritization of build acceptance tests for an enterprise cloud application: an industrial case study," *Journal of Systems and Software*, vol. 119, pp. 122–135, 2016.

[14] T. Parveen and S. Tilley, "When to migrate software testing to the cloud?" in *Proceedings of the Third International Conference on Software Testing, Verification, and Validation Workshops (ICSTW)*, pp. 424–427, Paris, France, April 2010.

[15] Z. Ganon and I. E. Zilbershtein, "Cloud-based performance testing of network management systems," in *Proceedings of the 14th IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 1–6, Pisa, Italy, June 2009.

[16] L. Riungu-Kalliosaari, "Empirical study on the adoption, use and effects of cloud-based testing," Doctoral Dissertation, Lappeenranta University of Technology, Lappeenranta, Finland, 2014.

[17] D. Kumar and K. K. Mishra, "The impacts of test automation on software's cost, quality and time to market," *Procedia Computer Science*, vol. 79, pp. 8–15, 2016.

[18] S. Ali and N. Ullah, "Critical influential factors for software testing-as-a-service adoption: preliminary findings from systematic literature review," in *Proceedings of the International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pp. 1–6, Swat, Pakistan, July 2019.

[19] M. Diaz and J. Sligo, "How software process improvement helped Motorola," *IEEE Software*, vol. 14, no. 5, pp. 75–81, 1997.

[20] S. Ali and S. U. Khan, "Software outsourcing partnership model: an evaluation framework for vendor organizations," *Journal of Systems and Software*, vol. 117, pp. 402–425, 2016.

[21] S. Ali, H. Li, S. U. Khan, Y. Zhao, and L. Li, "Fuzzy multi attribute assessment model for software outsourcing partnership formation," *IEEE Access*, vol. 6, pp. 55431–55461, 2018.

[22] S. Ali and H. Li, "Moving software testing to the cloud: an adoption assessment model based on fuzzy multi-attribute decision making algorithm," in *Proceedings of the IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA)*, pp. 382–386, Tokyo, Japan, April 2019.

[23] R. A. Khan, M. Y. Idris, S. U. Khan et al., "An evaluation framework for communication and coordination processes in offshore software development outsourcing relationship: using fuzzy methods," *IEEE Access*, vol. 7, pp. 112879–112906, 2019.

[24] X. Bai, M. Li, B. Chen, W. T. Tsai, and J. Gao, "Cloud testing tools," in *Proceedings of the IEEE 6th International Symposium on Service Oriented System Engineering (SOSE)*, pp. 1–12, Irvine, CA, USA, December 2011.

[25] L. Riungu-Kalliosaari, O. Taipale, K. Smolander, and I. Richardson, "Adoption and use of cloud-based testing in practice," *Software Quality Journal*, vol. 24, no. 2, pp. 337–364, 2016.

[26] I. C. Priyanka, I. Chana, and A. Rana, "Empirical evaluation of cloud-based testing techniques," *ACM Sigsoft Software Engineering Notes*, vol. 37, no. 3, pp. 1–9, 2012.

[27] C. Jia, Y. Cai, Y. T. Yu, and T. H. Tse, "5W + 1H pattern: a perspective of systematic mapping studies and a case study on cloud software testing," *Journal of Systems and Software*, vol. 116, pp. 206–219, 2016.

[28] W.-T. Tsai and G. Qi, "Integrated fault detection and test algebra for combinatorial testing in TaaS (Testing-as-a-Service)," *Simulation Modelling Practice and Theory*, vol. 68, pp. 108–124, 2016.

[29] S. Zein, N. Salleh, and J. Grundy, "A systematic mapping study of mobile application testing techniques," *Journal of Systems and Software*, vol. 117, pp. 334–356, 2016.

[30] T. Banzai, H. Koizumi, R. Kanbayashi et al., "Design of a software testing environment for reliable distributed systems using cloud computing technology," in *Proceedings of the IEEE/ACM International Symposium Cluster Computing and the Grid (CCGRID)*, pp. 631–636, Melbourne, Australia, May 2010.

[31] R. M. Ashfaq, "A new testing framework for cloud-based application," *Journal of Software Engineering & Intelligent Systems*, vol. 2, no. 3, pp. 157–163, 2017.

[32] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud migration research: a systematic review," *IEEE Transactions on Cloud Computing*, vol. 1, no. 2, pp. 142–157, 2013.

[33] P. Harikrishna and A. Amuthan, "A survey of testing as a service in cloud computing," in *Proceedings of the International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–5, Coimbatore, India, 2016.

[34] A. A. Ahmad, P. Brereton, and P. Andras, "A systematic mapping study of empirical studies on software cloud testing methods,," in *Proceedings of the IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pp. 555–562, Prague, Czech Republic, July 2017.

[35] A. F. Mohammad and H. McHeick, "Cloud services testing: an understanding," *Procedia Computer Science*, vol. 5, pp. 513–520, 2011.

[36] J. Fernandes and F. Gemmer, "Application testing on the cloud: smart testing for agile enterprises," 2020, https://www.oracle.com/technetwork/oem/s317292-app-testing-in-cloud-181639.pdf.

[37] A. Bertolino, L. Nautiyal, and Preeti, "Annotated buzzwords and key references for software testing in the cloud,," in *Proceedings of the International Conference on Computing, Communication and Automation (ICCCA)*, pp. 893–900, Noida, India, May 2017.

[38] S. Nachiyappan and S. Justus, "Cloud testing tools and its challenges: a comparative study," *Procedia Computer Science*, vol. 50, pp. 482–489, 2015.

[39] M. Tiitinen, "Key factors for selecting software testing tools," Master Degree in Business Informatics Master Business Informatics, Helsinki Metropolia University of Applied Sciences, Helsinki, Finland, 2013.

[40] M. S. N. Murthy and V. Suma, "A study on cloud computing testing tools," in *ICT and Critical Infrastructure: In Proceedings of the 48th Annual Convention of Computer Society of India*, vol. I, pp. 605–612, Springer, Cham, Switzerland, 2014.

[41] M. S. Geetha Devasena, V. Krishna Kumar, and R. Kingsy Grace, "LTTC: a load testing tool for cloud," in *Proceedings of International Conference on Communication and Networks*, pp. 689–698, Singapore, 2017.

[42] M. O. Alassafi, A. Alharthi, R. J. Walters, and G. B. Wills, "A framework for critical security factors that influence the decision of cloud adoption by Saudi government agencies," *Telematics and Informatics*, vol. 34, no. 7, pp. 996–1010, 2017.

[43] N. Alkhater, R. Walters, and G. Wills, "An empirical study of factors influencing cloud adoption among private sector organizations," *Telematics and Informatics*, vol. 35, no. 1, 2018.

[44] Y. Alshamaila, S. Papagiannidis, and F. Li, "Cloud computing adoption by SMEs in the north east of England," *Journal of Enterprise Information Management*, vol. 26, no. 3, pp. 250–275, 2013.

[45] P. Gupta, A. Seetharaman, and J. R. Raj, "The usage and adoption of cloud computing by small and medium businesses," *International Journal of Information Management*, vol. 33, no. 5, pp. 861–874, 2013.

[46] P.-F. Hsu, S. Ray, and Y.-Y. Li-Hsieh, "Examining cloud computing adoption intention, pricing mechanism, and deployment model," *International Journal of Information Management*, vol. 34, no. 4, pp. 474–488, 2014.

[47] S. K. Sharma, A. H. Al-Badi, S. M. Govindaluri, and M. H. Al-Kharusi, "Predicting motivators of cloud computing adoption: a developing country perspective," *Computers in Human Behavior*, vol. 62, pp. 61–69, 2016.

[48] R. D. Raut, B. B. Gardas, M. K. Jha, and P. Priyadarshinee, "Examining the critical success factors of cloud computing adoption in the MSMEs by using ISM model," *The Journal of High Technology Management Research*, vol. 28, no. 2, pp. 125–141, 2017.

[49] K. Incki, B. Ari, and H. Sozer, "A survey of software testing in the cloud,," in *Proceedings of the IEEE Sixth International Conference on Software Security and Reliability (SESR)*, pp. 18–23, Gaithersburg, MD, USA, August 2012.

[50] R. R. Kumar and C. Kumar, "A multi criteria decision making method for cloud service selection and ranking," *International Journal of Ambient Computing and Intelligence*, vol. 9, no. 3, pp. 1–14, 2018.

[51] P. Priyadarshinee, R. D. Raut, M. K. Jha, and B. B. Gardas, "Understanding and predicting the determinants of cloud computing adoption: a two staged hybrid SEM - neural networks approach," *Computers in Human Behavior*, vol. 76, pp. 341–362, 2017.

[52] S. Nazir, S. Shahzad, M. Nazir, and H. U. Rehman, "Evaluating security of software components using analytic network process," in *Proceedings of the 11th International Conference on Frontiers of Information Technology (FIT)*, pp. 183–188, Islamabad, Pakistan, December 2013.

[53] S. Nazir, S. Anwar, S. A. Khan et al., "Software component selection based on quality criteria using the analytic network process," *Abstract and Applied Analysis*, vol. 2014, Article ID 535970, 12 pages, 2014.

[54] S. Nazir, M. A. Khan, S. Anwar, H. Khan, and M. Nazir, "A novel fuzzy logic based software component selection modeling," in *Proceedings of the 2012 International Conference on Information Science and Applications (ICISA)*, pp. 1–6, Suwon, South Korea, May 2012.

[55] S. Nazir, S. Shahzad, S. Mahfooz, and M. Nazir, "Fuzzy logic based decision support system for component security evaluation," *The International Arab Journal of Information Technology*, vol. 15, no. 2, pp. 224–231, 2018.

[56] S. Tilley and B. Floss, "Hard problems in software testing solutions using testing as a service (TaaS)," vol. 2, no. 1, pp. 1–103, 2014.

[57] W. Kim, S. D. Kim, E. Lee, and S. Lee, "Adoption issues for cloud computing," in *Proceedings of the 11th International Conference on Information Integration and Web-Based Applications and Services*, pp. 3–6, Kuala Lumpur, Malaysia, December 2009.

[58] P. Prodanovic, *Fuzzy Set Ranking Methods and Multiple Expert Decision Making*, Department of Civil and Environmental Engineering Facility for Intelligent Decision Support, University of Western, Ontario, Canada, 2001.

[59] T. H. Chang and T. C. Wang, "Using the fuzzy multi-criteria decision making approach for measuring the possibility of successful knowledge management," *Information Sciences*, vol. 179, no. 13, pp. 355–370, 2009.

[60] A. Sangaiah and A. Thangavelu, "An exploration of FMCDM approach for evaluating the outcome/success of GSD projects," *Central European Journal of Engineering*, vol. 3, pp. 419–435, 2013.

[61] G. Li, G. Kou, C. Lin, L. Xu, and Y. Liao, "Multi-attribute decision making with generalized fuzzy numbers," *Journal of the Operational Research Society*, vol. 66, pp. 1793–1803, 2015.

[62] S. Guo and H. Zhao, "Fuzzy best-worst multi-criteria decision-making method and its applications," *Knowledge-Based Systems*, vol. 121, pp. 23–31, 2017.

[63] G. Li, G. Kou, and Y. Peng, "A group decision making model for integrating heterogeneous information," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 48, no. 6, pp. 1–11, 2016.

[64] G. Kou, Y. Peng, and G. Wang, "Evaluation of clustering algorithms for financial risk analysis using MCDM methods," *Information Sciences*, vol. 275, pp. 1–12, 2014.

[65] M.-S. Kuo and G.-S. Liang, "A soft computing method of performance evaluation with MCDM based on interval-valued fuzzy numbers," *Applied Soft Computing*, vol. 12, no. 1, pp. 476–485, 2012.

[66] Y.-J. Wang, "Applying FMCDM to evaluate financial performance of domestic airlines in Taiwan," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1837–1845, 2008.

[67] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[68] C. C. Kitchenham, "Guidelines for performing systematic literature reviews in software engineering," Ver. 2.3 EBSE Technical Report, Durham University, Durham, UK, 2007.

[69] T. C. Lethbridge, S. E. Sim, and J. Singer, "Studying software engineers: data collection techniques for software field studies," *Empirical Software Engineering*, vol. 10, no. 3, pp. 311–341, 2005.

[70] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, Sage Publications, Thousand Oaks, CA, USA, 2013.

[71] P. Priyadarshinee, R. D. Raut, M. K. Jha, and S. S. Kamble, "A cloud computing adoption in Indian SMEs: scale development and validation approach," *The Journal of High Technology Management Research*, vol. 28, no. 2, pp. 221–245, 2017.

[72] R. K. Yin, *Case Study Research: Design and Methods*, Sage Publications, Thousand Oaks, CA, USA, 2009.

[73] M. K. Daskalantonakis, "Achieving higher SEI levels," *IEEE Software*, vol. 11, no. 4, pp. 17–24, 1994.

[74] S. Tilley and T. Parveen, "SMART-T: migrating testing to the cloud," *Software Testing in the Cloud: Migration and Execution, Springer Briefs in Computer Science*, pp. 19–35, Springer, Berlin, Germany, 2012.

[75] A. O. Portillo-Dominguez, M. Wang, J. Murphy, and D. Magoni, "Automated WAIT for cloud-based application testing," in *Proceedings of the IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pp. 370–375, Cleveland, OH, USA, March 2014.

[76] A. Gambi, W. Hummer, and S. Dustdar, "Automated testing of cloud-based elastic systems with AUToCLES," in *Proceedings of the 28th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 714–717, Palo Alto, CA, USA, November 2013.

[77] M. Arslan, U. Qamar, S. Hassan, and S. Ayub, "Automatic performance analysis of cloud based load testing of web-application & its comparison with traditional load testing," in *Proceedings of the 37th International Conference on Software Engineering (ICSE)*, pp. 140–144, Firenze, Italy, May 2015.

[78] Z. Yang, M. Xiao, X. Zhao, D. Feng, L. Zhang, and H. Song, "Cloud computing and exploration of its application to test field and PHM," in *Proceedings of the Prognostics and System Health Management Conference*, pp. 1–4, Chengdu, China, October 2016.

[79] H. Muccini, A. D. Francesco, and P. Esposito, "Software testing of mobile applications: challenges and future research directions," in *Proceedings of the 7th International Workshop on Automation of Software Test (AST)*, pp. 29–35, Zurich, Switzerland, June 2012.

[80] Capgemini and Sogetti, "World quality report 2014-2015," 2020, https://www.sogeti.com/explore/reports/world-quality-report-2019.

[81] J. Zhou, S. Li, Z. Zhang, and Z. Ye, "Position paper: cloud-based performance testing: issues and challenges," in *Proceedings of the 2013 International Workshop on Hot Topics in Cloud Services*, pp. 55–62, Prague, Czech Republic, April 2013.

[82] L. Ciortea, C. Zamfir, S. Bucur, V. Chipounov, and G. Candea, "Cloud 9," *ACM SIGOPS Operating Systems Review*, vol. 43, no. 4, pp. 5–10, 2010.

[83] Q. Gao, W. Wang, G. Wu, X. Li, J. Wei, and H. Zhong, "Migrating load testing to the cloud: a case study," in *Proceedings of the IEEE Seventh International Symposium on Service-Oriented System Engineering (SOSE)*, pp. 429–434, Redwood City, CA, USA, March 2013.

[84] S. U. Khan, M. Niazi, and R. Ahmad, "Systematic literature review protocol for a software outsourcing vendors readiness model (SOVRM)," in *Proceedings of the International Conference on Information and Communication Technology (ICICT)*, pp. 1–8, Bannu, Pakistan, 2008.

[85] A. S. Al-Ahmad, S. A. Aljunid, and A. S. A. Sani, "Mobile cloud computing testing review," in *Proceedings of the 2013 International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*, pp. 176–180, Kuching, Malaysia, 2013.

[86] R. D. Raut, P. Priyadarshinee, B. B. Gardas, and M. K. Jha, "Analyzing the factors influencing cloud computing adoption using three stage hybrid SEM-ANN-ISM (SEANIS) approach," *Technological Forecasting and Social Change*, vol. 134, pp. 98–123, 2018.

[87] S. Ali, L. Hongqi, S. U. Khan, Y. Zhongguo, and Z. Liping, "Success factors for software outsourcing partnership management: an exploratory study using systematic literature review," *IEEE Access*, vol. 5, pp. 23589–23612, 2017.

[88] S. Ali and S. U. khan, "Critical success factors for software outsourcing partnership (SOP): a systematic literature review," in *Proceedings of the 14th International Conference on Global Software Engineering (ICGSE)*, pp. 153–162, Shangai, China, August 2014.

[89] H. Coolican, *Research Methods and Statistics in Psychology*, Routledge, London, UK, 7th edition, 2018.

Hindawi
Scientific Programming
Volume 2020, Article ID 5763837, 12 pages
https://doi.org/10.1155/2020/5763837

Research Article

Image Interpolation via Gradient Correlation-Based Edge Direction Estimation

Sajid Khan,[1] Dong-Ho Lee,[2] Muhammad Asif Khan,[3] Muhammad Faisal Siddiqui,[4] Raja Fawad Zafar,[5] Kashif Hussain Memon,[2] and Ghulam Mujtaba[1]

[1]*Center of Excellence for Robotics, Artificial Intelligence, and Blockchain, Sukkur IBA University, Sukkur, Pakistan*
[2]*School of Electrical Engineering, Hanyang University ERICA Campus, Ansan, Republic of Korea*
[3]*Department of Electrical Engineering, Sukkur IBA University, Sukkur, Pakistan*
[4]*Faculty of Engineering, Department of Electrical Engineering, University of Malaya, Kuala Lumpur 50603, Malaysia*
[5]*Department of Mathematics and Social Sciences, Sukkur IBA University, Sukkur, Pakistan*

Correspondence should be addressed to Dong-Ho Lee; dhlee77@hanyang.ac.kr

Received 3 January 2020; Accepted 17 February 2020; Published 21 April 2020

Guest Editor: Rahman Ali

This paper introduces an image interpolation method that provides performance superior to that of the state-of-the-art algorithms. The simple linear method, if used for interpolation, provides interpolation at the cost of blurring, jagging, and other artifacts; however, applying complex methods provides better interpolation results, but sometimes they fail to preserve some specific edge patterns or results in oversmoothing of the edges due to postprocessing of the initial interpolation process. The proposed method uses a new gradient-based approach that makes an intelligent decision based on the edge direction using the edge map and gradient map of an image and interpolates unknown pixels in the predicted direction using known intensity pixels. The input image is subjected to the efficient hysteresis thresholding-based edge map calculation, followed by interpolation of low-resolution edge map to obtain a high-resolution edge map. Edge map interpolation is followed by classification of unknown pixels into obvious edges, uniform regions, and transitional edges using the decision support system. Coefficient-based interpolation that involves gradient coefficient and distance coefficient is applied to obvious edge pixels in the high-resolution image, whereas transitional edges in the neighborhood of an obvious edge are interpolated in the same direction to provide uniform interpolation. Simple line averaging is applied to pixels that are not detected as an edge to decrease the complexity of the proposed method. Applying line averaging to smooth pixels helps to control the complexity of the algorithm, whereas applying gradient-based interpolation preserves edges and hence results in better performance at reasonable complexity.

1. Introduction

Image interpolation is used to produce a high-resolution image from a low-resolution one. Image interpolation is used in many applications and fields of image processing [1–3]. The increase in the resolution of modern-day display screens demands artifact-free upscaling of videos and images [4]. The TV industry has and is still evolving through the introduction of advanced and high-resolution technologies. The evolution of the TV industry from analog to digital and then from HDTV to SUHDTV increases the significance of image/video interpolation. Video interpolation is needed to produce a pleasant experience of viewing a low-resolution video on high-resolution displays. A video essentially consists of a sequence of frames, and each frame is represented by an image. Upscaling a video sequence typically amounts to upscaling a series of frames. An image interpolation method should interpolate a low-resolution image in such a way that it produces no artifacts or few artifacts to provide a pleasant viewing experience.

Image interpolation is also used in medical image processing [5, 6] to enhance magnetic resonance images (MRI), picture archiving and communication systems (PACS), computer-assisted surgery (CAS), digital

subtraction angiography (DSA), and computer-aided diagnosis. After the acquisition of medical images, the zooming and rotation of images are required for the purpose of diagnosis and treatment, and interpolation methods are required to perform such operations.

An image contains two regions: uniform regions and edges. Interpolating an edge correctly is the most important requirement of an interpolating algorithm since human eyes are very sensitive to image edges. The production of artifacts along an edge results in an unpleasant viewing experience. Most of the interpolation algorithms that are prominent in the field [6–18] have focused on the interpolation of edges. These algorithms range from simple linear algorithms to complex edge-preserving methods. Linear algorithms fail to preserve most of the edges, whereas complex algorithms preserve edges very well compared to linear algorithms. However, these complex algorithms fail to detect certain edge patterns such as thin edges and sometimes produce artifacts along edges.

Nearest neighbor, bilinear, and bicubic [6] are the most prominent simple methods, providing time-efficient interpolations at the cost of blurring, jagging, and smoothing along the edges. Paik et al. [10] used cross correlations using horizontal and vertical direction vectors that fail in restoring edges that are not correlated with predefined horizontal and vertical vectors. Allebach and Wong [11] interpolated smooth regions using bilinear interpolation and applied data correction along edges. The problem with EDI is its sensitivity to the texture that can spuriously produce artifacts along the edges. Dong et al. [12] combined a nonlocal aggressive model with a sparse representation model. The sparse representation-based method provides reasonable results, but they are always time-inefficient. Sparse-based methods are dictionary-based, and they change local properties such as image intensity and edge properties, as well as introduce artifacts along the edges. Shi and Shan [14] proposed a method that correlates neighbors of the unknown pixel with six modes and interpolates unknown pixels along the mode that has the closest correlation. The problem with this method is that it uses sixteen pixels for each mode, and the major variation of only one pixel along an appropriate mode might result in the rejection of that mode.

Some researchers have provided particularly promising contributions in the field of image interpolation. Li and Orchard [15] proposed the interpolation method new edge-directed interpolation (NEDI). NEDI calculates local covariance coefficients from a low-resolution input image and then uses these coefficients to a produce high-resolution image on the basis of geometric duality between the coefficients of both images. However, NEDI fails to preserve thin edges or high-frequency regions.

Giachetti and Asuni [17] proposed the iterative curvature-based interpolation (ICBI) method that provides better results compared to other conventional algorithms and preserves edges well. ICBI is a two-step interpolation that uses a second-order derivative in the first step and minimization of energy functions in the second step. Minimization of the energy function refines edges, which can result in oversmoothing of edges. Yang et al. [13] and Timofte et al. [18] worked on a dictionary learning-based interpolation algorithm. Timofte et al. proposed an anchored neighborhood regression (ANR) approach that provides a much faster dictionary learning-based interpolation compared to the method proposed by Yang et al. The problem with a dictionary-based learning approach is that information in an image is replaced using a limited size dictionary that sometimes changes image properties such as intensity and edge shapes.

In this paper, a new interpolation algorithm, named directional gradient-based edge interpolation (DGEI), is proposed. The proposed method categorizes the input image into obvious edges, transitional regions, and uniform regions. Uniform regions are interpolated using simple line averaging, whereas DGEI is applied to obvious edge pixels. DGEI predicts the direction of an edge and interpolates unknown edge pixels in the direction with a high gradient correlation. Transitional edges are interpolated in the direction of neighboring pixels. The proposed method preserves edges with no or minimum artifacts using all possible predefined edge patterns. Simulation results show that the proposed algorithm provides better results compared to bicubic, NEDI, ICBI, and ANR.

The rest of this paper is organized as follows. Section 2 describes some of the conventional methods. Section 3 describes the proposed algorithm. Section 4 describes the performance analysis. Section 5 is the conclusion.

2. Conventional Methods

Before describing the proposed algorithm, NEDI, ICBI, and ANR are briefly discussed since they are state-of-the-art methods and provide good results.

NEDI assumes that the input low-resolution image of size $H \times W$ directly comes from high-resolution output image of size c with pixel to pixel relation of $I_{2i,2j} = X_{i,j}$. Hence, unknown pixel can be calculated using the following equation:

$$I_{2i+1,2j+1} = \sum_{k=0}^{1} \sum_{l=0}^{1} \propto_{2k+1} I_{2(i+k),2(j+l)}. \tag{1}$$

Generalized equation for $\propto$ is given as

$$\overrightarrow{\propto} = R^{-1}\overrightarrow{r}, \tag{2}$$

where $R = [R_{kl}]$ and $\overrightarrow{r} = [r_k]$ are high-resolution covariance. Both R_{kl} and r_k can be obtained from Figure 1.

The problem with NEDI is that is assuming that low-resolution images and high-resolution images are symmetrical. This assumption works well for nonedges and smooth edges, but it fails to interpolate when there is an abrupt change in edge, thin edges, or region with texture.

ICBI is a two-step algorithm that first interpolates image in the main diagonal or auxiliary diagonal direction based on the minimum response of second-order derivative. It then refines the initially interpolated values by minimizing three energy terms U_c, U_e, and U_i which are edge sharpness, artifact removal, and preserving isolevel curves. The energy function that requires minimization is given as

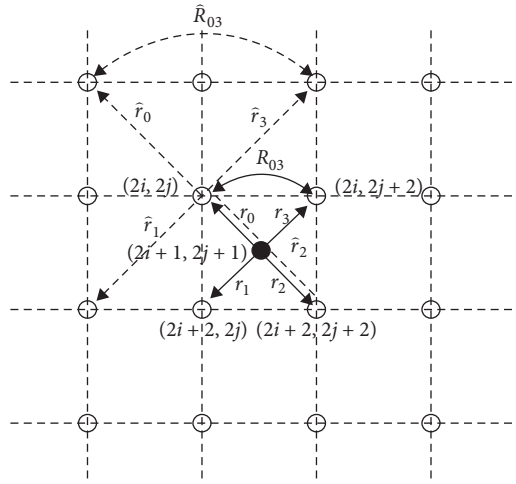


Figure 1: Correspondence between low- and high-resolution covariance.

$$U(2i+1,2j+1) = aUc(2i+1,2j+1) + bUe(2i+1,2j+1) + cUi(2i+1,2j+1). \tag{3}$$

U_c, U_e, and U_i are given in the following equations, respectively.

$$U_c(2i+1,2j+1) = w_1D_1 + w_2D_2 + w_3D_3 + w_4D_4, \tag{4}$$

$$U_e(2i+2j) = \left|I_{11}(2i+1,2j+1)\right| + \left|I_{22}(2i+1,2j+1)\right|, \tag{5}$$

$$U_i = f(I)|_{2i+1,2j+1} I(2i+1,2j+1), \tag{6}$$

where

$$\begin{aligned} D_1 &= \left|\left((I_{11}2i+1,2j+1) - I_{11}(2i+2,2j+2)\right)\right| + \left|\left(I_{22}(2i+1,2j+1) - I_{22}(2i+2,2j+2)\right)\right|, \\ D_2 &= \left|\left(I_{11}(2i+1,2j+1) - I_{11}(2i+2,2j)\right)\right| + \left|\left(I_{22}(2i+1,2j+1) - I_{22}(2i+2,2j)\right)\right|, \\ D_3 &= \left|I_{11}(2i+1,2j+1) - I_{11}(2i,2j+2)\right| + \left|I_{22}(2i+1,2j+1) - I_{22}(2i,2j+2)\right|, \\ D_4 &= \left|\left(I_{11}(2i+1,2j+1) - I_{11}(2i,2j)\right)\right| + \left|\left(I_{22}(2i+1,2j+1) - I_{22}(2i,2j)\right)\right|. \end{aligned} \tag{7}$$

The problem with ICBI is that the minimization of energy terms does not consider the properties of the known pixel and that is why it sometimes produces artifacts for low-level upscaling (2x) and most of the time for high-level upscaling (4x or 8x). Production of false isolevel curves because of the energy function U_i is an issue with ICBI.

ANR [18] is a patch-based dictionary learning approach based on minimization of the following function:

$$\min_{\beta} \left\| y_F - N_l\beta \right\|_2^2 + \gamma\|\beta\|_2, \tag{8}$$

where N_l is set that consists of K nearest neighbors of the input feature image and γ. allows to alleviate the singularity of ill-posed problems and stabilize the coefficient β.

The problem with ANR is that it completely changes some properties of an image such as intensity, production of artifacts in the transitional region, and unnecessary sharpness of edges.

3. The Proposed Method

The objective of this work is to propose a method that provides artifact-free upscaling and preserves the natural properties of the image. Figure 2 shows the flowchart of the algorithm. The proposed algorithm initially calculates the edge map of the low-resolution image by first calculating gradient G_{low} of the input image of size $M \times N$ and then applies hysteresis thresholding to G_{low} to obtain a low-resolution edge map E_{low}. E_{low} contains both obvious and

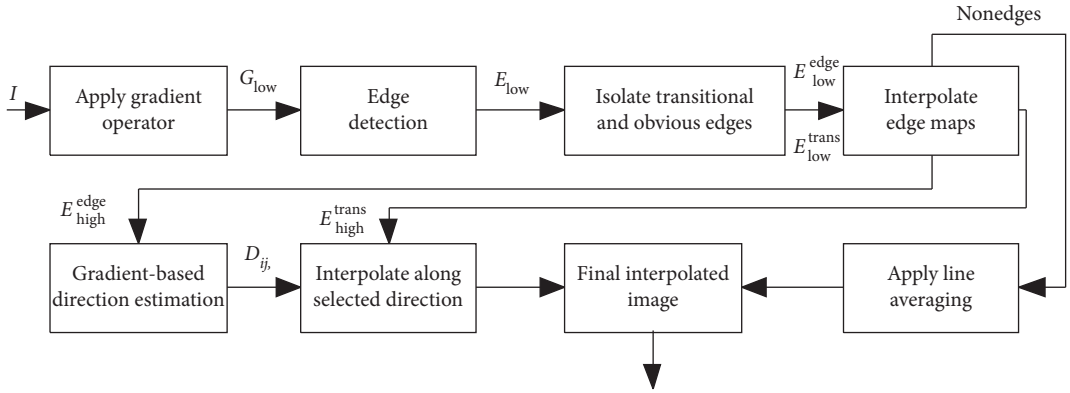


Figure 2: Flowchart of the proposed method.

transition regions since transition regions also have strong gradient values. Transition regions and obvious edge are isolated by applying nonmaximal suppression to produce E^{trans}_{low} and E^{edge}_{low}, respectively. High-resolution edge maps E^{edge}_{high} and E^{trans}_{high} of sizes $2M \times 2N$ are obtained by processing E^{edge}_{low} and E^{trans}_{low}, respectively. Smooth pixels are subjected to the application of simple line averaging, whereas obvious edges in E^{edge}_{high} are processed by directional gradient-based interpolation. Transition regions are interpolated in the direction estimated for their neighboring obvious edge pixel to provide uniformity along edge and transition regions.

3.1. Edge Detection. Hysteresis thresholding-based edge detection is used for edge detection. Hysteresis thresholding [19–24] involves the calculation of two thresholds T_{low} and T_{high} and uses edge-map linking for better true detection and noise reduction compared to single thresholding [25, 26]. The method proposed by Medina-Carnicer et al. [21] was used for the calculation of T_{low} and T_{high} as it provides reasonable pair of low and high thresholds.

The input of the Medina-Carnicer method is the gradient image G_{low} obtained by applying horizontal and vertical Sobel operators [27].

After calculation of T_L and T_H, low- and high-thresholded edge maps E_L and E_H are obtained. E_L contains all the obvious edges along with noisy edges, whereas E_H contains incomplete true edges. For each incomplete edge in E_H, the corresponding connected edge pixels from E_L were obtained and added to the final low-resolution edge map E_{low}. In that way, all the true edges were taken from E_L while ignoring noisy edges, as E_H does not have the representation of noisy edges.

3.2. Isolating Transitions and Obvious Edges. Edges of the edge map E_{low} are thick, as transition regions also have some reasonable gradient values and are detected in the low threshold edge map. To separate transition edges from obvious edges, nonmaximal suppression is applied to gradients at detected edge locations. Nonmaximal suppression is a technique used by Canny [22]. This method is used to thin edges by selecting the edge that corresponds to a maximum gradient value. Canny applied nonmaximal suppression to the gradient image before thresholding. The proposed method, however, applies nonmaximal suppression only after the application of edge detection. Nonmaximal suppression is applied to the image $G_{low} \times E_{low}$ to obtain an edge map E^{edge}_{low} that has edges with a thickness of only one pixel. These edges are considered as obvious edges because they have a maximum gradient value in a thick detected edge. The transition map that corresponds to the transition regions can be calculated using

$$E^{trans}_{low} = E_{low} - E^{edge}_{low}. \quad (9)$$

Figure 3 shows the results of the isolation of the edge map and transition map. It can be seen from Figure 3 that the edge map obtained using nonmaximal suppression has edges of a thickness of one pixel. The direction obtained using E^{edge}_{low} contains a small chance of error because the direction of an edge can be predicted more precisely if the thickness of the edge is one pixel. E^{trans}_{low}, on the other hand, contains thick transition regions, and the probability of producing an error is high if the direction is obtained separately for transition regions. However, it is clear from Figure 3 that transitional edges in E^{trans}_{low} have the same direction as those of E^{edge}_{low}. Therefore, if an accurate direction for an edge pixel of E^{edge}_{low} is predicted, an interpolation in the same direction of its neighboring transition pixels in E^{trans}_{low} can be performed to produce artifact-free image upscaling.

3.3. Interpolating Edge/Transition Maps. Both E^{trans}_{low} and E^{edge}_{low} are subjected to simple edge map interpolation. The proposed algorithm upscales an edge/transition map by first interpolating along the row direction and then along the column direction. Both row and column interpolation follow the same procedure. The edge/transition map of size $M \times N$ is used as the input edge/transition map for performing edge/transition interpolation along the row direction in order to obtain the row-interpolated edge/transition map E^{Row} of size $2M \times N$. The transpose of E^{Row} is then used as the input image for column interpolation to obtain an interpolated image of size $2N \times 2M$. The transpose of interpolated $2N \times 2M$ image is needed to obtain a final interpolated image E^{edge}_{high} (or E^{trans}_{high} in the case of transition map interpolation) of size $2M \times 2N$.

(a) (b) (c)

Figure 3: Result of edge and transition region isolation; (a) E_{low}; (b) $E_{\text{low}}^{\text{edge}}$; and (c) $E_{\text{low}}^{\text{trans}}$.

A 3×3 window is moved across every unknown row. The pixel that has edge/transition pixels in any of the three directions (main diagonal, vertical, and auxiliary diagonal) shown in Figure 4 is selected as an edge/transition pixel. Figure 5 shows the results of the application of edge interpolation on $E_{\text{low}}^{\text{edge}}$.

3.4. Gradient-Based Direction Estimation. All the unknown pixels that are not detected as edge pixels or transitional regions in $E_{\text{high}}^{\text{edge}}$ and $E_{\text{high}}^{\text{trans}}$ are interpolated using simple line averaging. Information for known edge pixels in $E_{\text{high}}^{\text{edge}}$ is used to estimate the direction of unknown edge pixels. Unknown edge pixels are then interpolated along the estimated direction. Unknown transitional pixels in $E_{\text{high}}^{\text{trans}}$ are interpolated in the estimated direction of their neighboring edge pixel in $E_{\text{high}}^{\text{edge}}$. This section provides the direction calculation methodology for the unknown edges pixels in $E_{\text{high}}^{\text{edge}}$.

Figure 6 shows all possible directions that an edge might have. These directions include horizontal, vertical, main diagonal, auxiliary diagonal, left concave, right concave, upward concave, and downward concave. The direction for every unknown edge pixel is determined by using eight available directions.

Every unknown pixel has some cases that show the positions of unknown and known pixels around it. For evaluating the direction, only the properties of the known pixels can be used. Given that known pixels belong to an odd row and odd column, three cases show the positions of known and unknown pixels.

(1) Unknown pixel is at an even row and even column

(2) Unknown pixel is at an odd row and even column

(3) Unknown pixel is at an even row odd column

Figure 7 shows the three cases for the positions for known and unknown pixels. The red circle in each case shows the center unknown pixel, and blue circles show the positions of the known pixels, whereas black pixels show the positions of the surrounding unknown pixels. The case of even rows and even columns does not have known pixels along with adjacent vertical and horizontal directions, and the case of the odd row and even columns does not have known pixels along the vertical direction, whereas the case of even rows and odd columns do not have known pixels along the horizontal direction. Therefore, the number of possible directions for the first case is six, whereas in the second and third cases, the number of possible directions is seven.

The proposed direction estimation-based method uses edge map $E_{\text{high}}^{\text{edge}}$ to see if a certain direction is a candidate direction or not and then uses gradient correlation to determine the pixels that can be used for interpolation. A certain direction cannot be selected as a candidate direction if it does not follow the crossing condition, for example, if one wants to see whether or not the main diagonal is a candidate for an unknown edge pixel that is on an even row and even column. The pixels C31 and C51 should be edge pixels in order to satisfy the crossing condition. If any of these two pixels are not edge pixels, then the edge is certainly not moving along the main diagonal. Also, for gradient correlation, a reference gradient value is needed because gradient correlation takes the absolute difference between the reference gradient with respective gradients of possible pixels in that direction. These differences are finally compared with a threshold value to see whether the correlation is in the wanted range or not. Every direction has its own possible reference gradient position, its own "must be edges" condition to satisfy the crossing condition, and its own positions of possible directional members.

Table 1 shows all the information needed to estimate the direction for an unknown pixel that has a position of an even row and even column. Steps for estimating the direction of the unknown edge pixel are

(1) Take a 9×9 patch around the unknown edge pixel.

(2) For all directions, if possible references contain an edge pixel, move to step 3 and select the gradient value of that possible reference as a reference gradient. If none of the possible references for a direction has an edge, ignore that direction as the candidate direction.

(3) If the direction follows the "must be edges" condition given in Table 1, select this direction as the candidate direction and move to step 4. If the "must be edges"

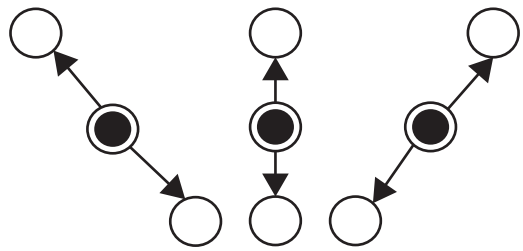


Figure 4: Directions selected for edge interpolation. (a) Main diagonal; (b) vertical; and (c) auxiliary diagonal.

Figure 5: Results of application of edge map interpolation. (a, c) Patches of $E_{\text{low}}^{\text{edge}}$; (b, d) patches of $E_{\text{high}}^{\text{edge}}$.

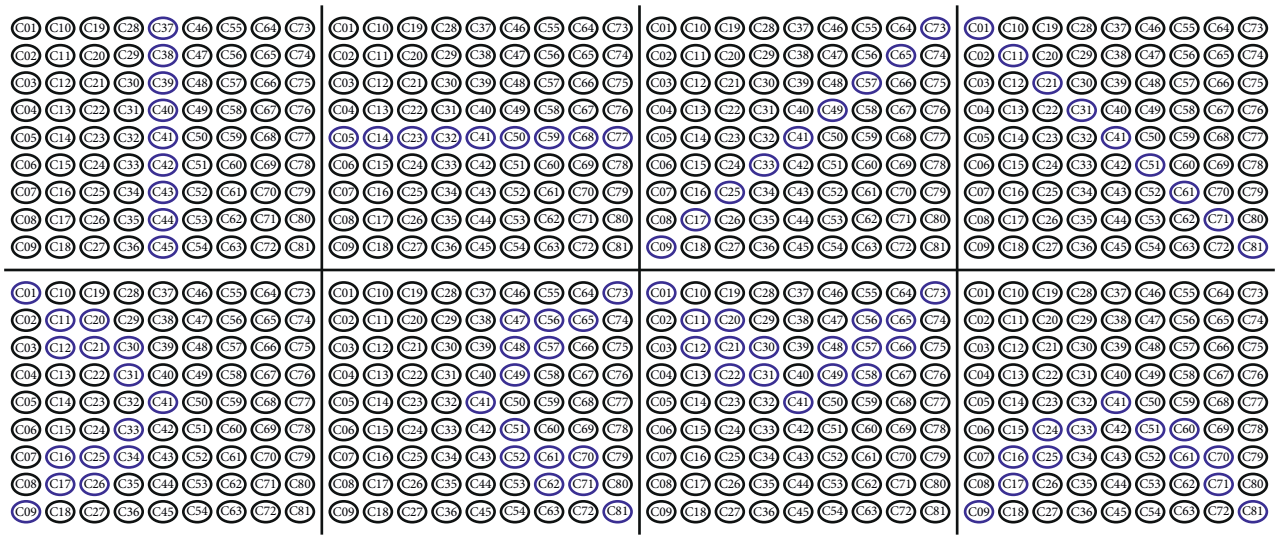


Figure 6: All possible primary configurations of an edge.

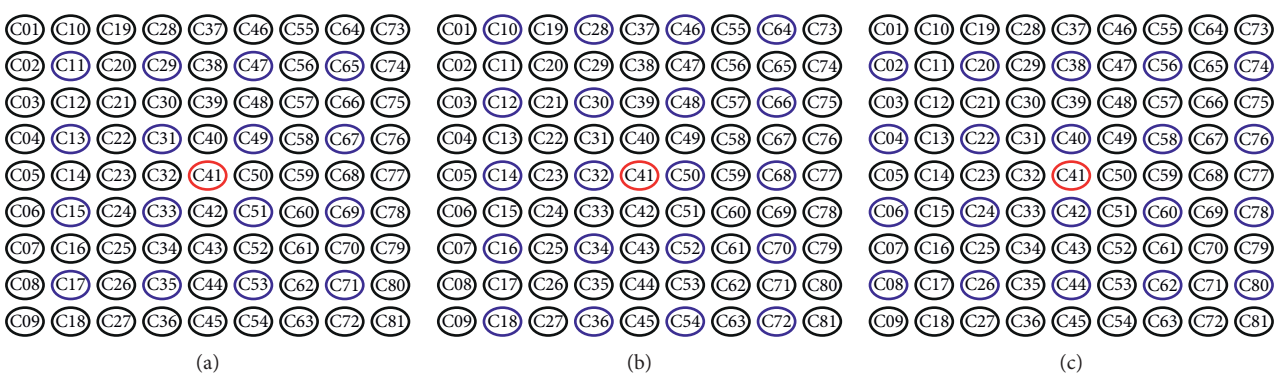


Figure 7: Three cases for positions of known and unknown pixels: (a) even row even column; (b) odd row even column; (c) even row odd column.

Table 1: Information for evaluating edge directions for the case of even rows and even columns.

Direction	Possible directional members	Possible references	Pixels that must be edges
Main diagonal	C11, C29, C13, C31, C51, C69, C53, C71	C31, C51	(C31 & C51)
Left concave	C11, C29, C31, C49, C51, C33, C35, C17	C31, C33	(C49 | C51) & (C33 & C31)
Auxiliary diagonal	C65, C47, C67, C49, C33, C15, C35, C17	C49, C33	(C49 & C33)
Right concave	C65, C47, C49, C31, C33, C51, C53, C71	C49, C51	(C31 | C33) & (C49 & C51)
Upward concave	C11, C13, C31, C33, C49, C51, C65, C67	C31, C49	(C33 | C51) & (C31 & C49)
Downward concave	C15, C17, C31, C33, C49, C51, C69, C71	C33, C51	(C31 | C49) (C33 & C51)
Horizontal	No horizontal direction available		
Vertical	No vertical direction available		

Table 2: Information for evaluating edge directions for the case of even rows and even columns.

Direction	Possible directional members	Possible references	Pixels that must be edges
Main diagonal	C10, C12, C30, C32, C50, C52, C70, C72	C32, C50, C30, C52	(C30 | C32) & (C50 | C52)
Left concave	C64, C66, C48, C50, C32, C34, C16, C18	C32, C50, C48	(C48 | C50) & (C32 | C34)
Auxiliary diagonal	C10, C12, C30, C32, C48, C50, C16, C34, C52, C18	C32, C30, C34	(C48 | C50 | C52) & (any two of C30, C32, & C34)
Right concave	C30, C32, C34, C48, C50, C52, C64, C66, C70, C72	C50, C48, C52	(C30 | C32 | C34) & (any two of C48, C50, & C52)
Upward concave	C10, C12, C30, C32, C48, C50, C64, C66	C32, C50	(C30 | C32) & (C50 | C48)
Downward concave	C16, C18, C32, C34, C50, C52, C70, C72	C32, C50	(C34 | C32) & (C50 | C52)
Horizontal	C14, C32, C50, C68	C32	(C32 & C50)
Vertical	No vertical direction available		

Table 3: Information for evaluating edge directions for the case of even rows and even columns.

Direction	Possible directional members	Possible references	Pixels that must be edges
Main diagonal	C2, C4, C20, C22, C40, C42, C60, C62, C78, C80	C22, C60	(C22 | C60) & (C40 | C42)
Left concave	C2, C8, C20, C22, C24, C26, C40, C42	C40, C42	(C20 | C22) & (C40 | C42) & (C24 | C26)
Auxiliary diagonal	C6, C8, C24, C26, C40, C42, C56, C58, C74, C76	C24, C58	(C24 | C58) & (C40 | C42)
Right concave	C40, C42, C56, C58, C60, C62, C74, C80	C40, C42	(C56 | C58) & (C40 | C42) & (C60 | C62)
Upward concave	C2, C20, C22, C40, C42, C56, C58, C74	C22, C40, C58	(C24 | C42 | C60) & (any two of C22, C40 & C58)
Downward concave	C8, C24, C26, C40, C42, C60, C62, C80	C24, C42, C60	(C22 | C40 | C58) & (any two of C24, C42 & C60)
Horizontal	No horizontal direction available		
Vertical	C38, C40, C42, C44	C40	(C40 & C42)

condition is not justified, ignore that direction as the candidate direction.

(4) Take the absolute difference of the gradient of "possible directional members" with the reference gradient to calculate the gradient correlation. Select all those pixels as interpolating pixels that have a gradient correlation less than $T = 21$.

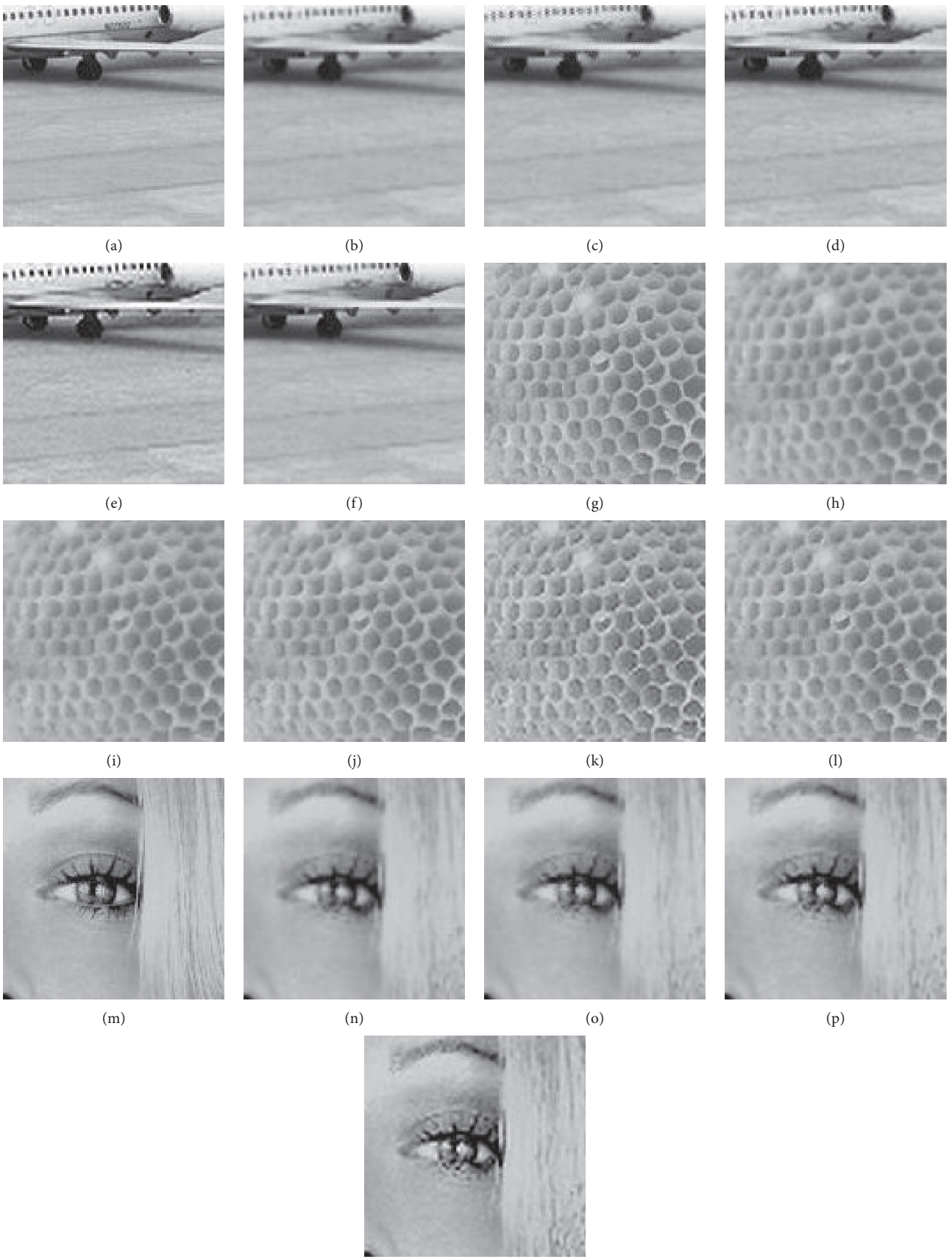
(5) Among all candidate directions, select the direction that has more pixels with a gradient correlation value of less than T. Store their gradient correlation in an array D_G. D_G will then be used in the coefficient calculation for interpolation.

Directions estimated for unknown edge pixels that have positions of odd rows and even columns or even rows and odd columns can be calculated in a similar way using Tables 2 and 3, respectively.

3.5. Interpolation Using the Estimated Direction. An edge pixel can be interpolated using

$$I_{\text{high}}^{\text{edge}} = \sum_{r=0}^{N_G-1} I_r^{\text{GC}} \times C_{G(r)}^{\text{GC}} * C_{D(r)}^{\text{GC}}, \tag{10}$$

where I_r^{GC}, $C_{G(r)}^{\text{GC}}$, and $C_{D(r)}^{\text{GC}}$ are the image intensity, the gradient coefficient, and the distance coefficient of the rth pixel with a gradient correlation less than T. C_G and C_D can be calculated using



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q)

Figure 8: Cropped regions after upscaling (×2); (a, g, m) original; (b, h, n) bicubic; (c, i, o) NEDI; (d, j, p) ICBI; (e, k, q) ANR; (f, l, q) proposed.

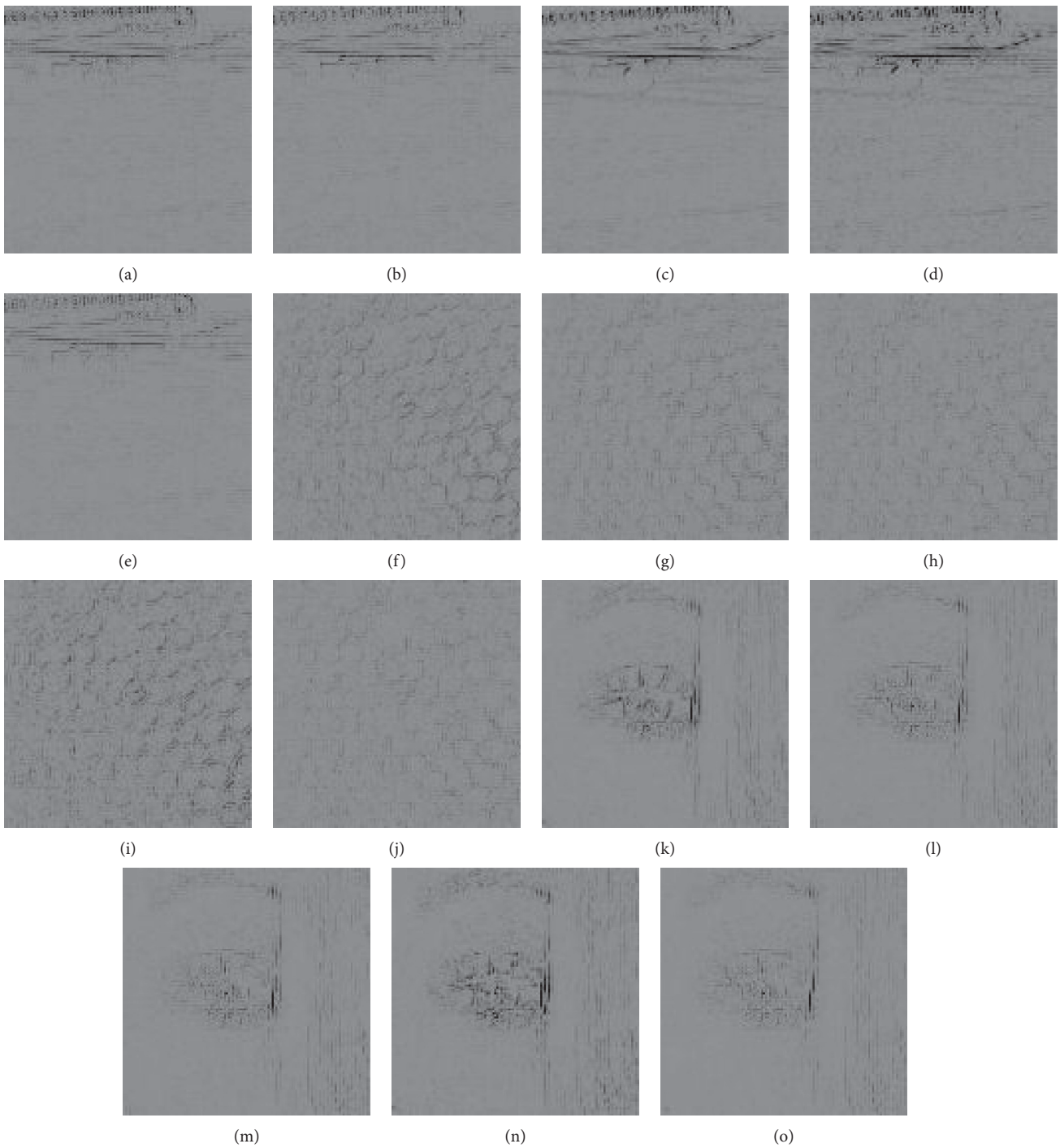


Figure 9: Error patches for cropped regions after upscaling (×2); (a, f, k) bicubic; (b, g, l) NEDI; (c, h, m) ICBI; (d, i, n) ANR; (e, j, o) proposed.

$$C_G = e^{-\sqrt[N]{D_G}}, \tag{11}$$

$$\widehat{C}_D = e^{-\left(x^2+y^2\right)/2\times\sigma^2}, \tag{12}$$

$$C_D = \frac{\widehat{C}_D}{\max\left(\widehat{C}_D\right)}. \tag{13}$$

Here, $N = 4$ and $\sigma = 1.5$ are parameters chosen as a result of simulations. The variables x and y in equation (12) are the x-directional and y-directional distances of a pixel from the center pixel. There is no need to dynamically calculate C_D for every pixel. This is because each pixel in a 9×9 patch is located at a fixed distance from the center pixel. Therefore, a 9×9 matrix of coefficients C_D can be calculated only once and used for all pixels. C_G, however, requires

dynamic calculation for every pixel because the gradient correlation is different for all pixels. Before applying C_D and C_G for interpolation, both coefficients should be normalized in such a way that it follows the given condition:

$$\sum_{r=0}^{N_G-1} C_{G(r)}^{GC} * C_{D(r)}^{GC} = 1. \quad (14)$$

After interpolating an edge pixel, its neighboring region needs to be scanned for the interpolation of transitional pixels. If the edge is moving in the vertical direction with zero slope or some slope, all of the transitional pixels that are connected to the left and right are scanned and interpolated in the same direction as the edge pixel. Unknown transitional pixels on the left or right direction of the interpolated edge pixel are checked for whether they are on an even row and even column, even row and odd column, or odd row and even column and are interpolated in that direction using information available in the corresponding Tables 1–3. A transitional pixel is interpolated by selecting a reference gradient and then following steps 4 and 5 of the previous subsection by calculating the gradient correlation D_G, using the gradient of the selected transitional pixel as the reference. After calculating D_G and indices for members of the array D_G, interpolation can be applied to the transitional pixel using equations (10)–(14). However, if an edge is moving in the horizontal direction, transitional pixels connected to its upward and downward directions are interpolated similarly. A few exceptional cases might occur when the direction estimated for an edge pixel is vertical or horizontal; however, some neighboring transitional pixels are located, where they do not have known vertical or horizontal pixels. In that case, the transitional pixel is interpolated by using the mean of the two closest pixels in that direction using line averaging.

4. Results

4.1. Subjective Evaluation. The proposed method is compared with the bicubic, NEDI, ICBI, and ANR approaches by applying algorithms on grayscale images provided in [28]. Both subjective and objective evaluations of all algorithms are performed in order to quantitatively compare the quality of the images created by different methods and the related computational costs. The tic toc function in Matlab 2018b was used to calculate the execution time. Bicubic interpolation was applied using the MATLAB built-in function named interp2, whereas the source codes for NEDI, ICBI, and ANR were kindly provided by Prof. Xin Li, Prof. Andrea Giacheti, and Dr. Yang, respectively.

Figure 8 shows some cropped regions of some grayscale test images that are upscaled by a factor of two. The input image was first downscaled by a factor of two, and then all algorithms were applied to the downscaled images to see whether or not they preserve distorted edges. It can be seen that the bicubic and NEDI algorithms failed to preserve edges. ANR provided some sharp edges, but changed the edge structures and intensity of those patches. ICBI preserved edges well; however, it oversmoothed the edges with the postprocessing step of refinement, as seen in Figure 8(j). The proposed method preserved edges well compared to other algorithms and also effectively preserved the contrast and shapes of the edges.

Table 4: Error indices comparison of all algorithms on the patches in Figure 9.

Image	Bicubic	NEDI	ICBI	ANR	Proposed
Airplane	646.07	394.29	377.38	606.04	336.86
Beehive	573.86	446.54	402.01	683.11	362.58
Woman	388.10	334.40	332.03	468.28	255.62

Table 5: PSNR (dB) comparison of all algorithms.

Image	Bicubic	NEDI	ICBI	ANR	Proposed
Airplane	25.762	27.509	27.873	20.075	28.679
Banana	34.23	36.124	36.853	22.739	36.931
Barbara	24.126	25.526	24.96	21.552	25.594
Brush	29.818	31.065	31.226	21.309	31.641
Beehive	31.643	33.245	33.664	20.723	33.921
Bridge	23.88	25.12	25.246	20.326	25.857
Brief case	27.98	29.532	30.254	22.894	30.644
Cofee maker	29.365	31.546	32.353	24.267	32.912
Elephant	26.051	27.577	27.611	19.845	28.164
Feather	25.703	27.14	27.208	21.599	27.953
Golf cart	27.529	29.011	29.306	21.173	30.075
Grater	30.47	31.772	32.126	20.626	32.618
Lena	30.028	32.619	33.523	22.764	32.958
Mail box	26.383	27.986	28.504	22.143	29.134
Pond	23.203	24.453	24.107	18.731	25.172
Shopping cart	24.412	25.622	25.52	19.362	26.705
Tire	25.873	26.859	27.025	19.566	27.901
Trafficone	29.102	30.683	31.319	23.5	32.379
Trash can	26.076	27.137	27.739	21.748	28.728
Woman	36.964	39.25	40.662	21.365	40.991
Average	27.923	29.488	29.851	21.315	30.4478

The error index (EI) defines an algorithm's cumulative magnitude of error for a patch and is calculated for all patches of size $M \times N$, given as

$$\mathrm{EI} = \sum_{r=1}^{M\times N} \left|\mathrm{Patch}_{\mathrm{orig}} - \mathrm{Patch}_{\mathrm{algo}}\right|. \quad (15)$$

Error indices calculated for patches in Figure 9 are given in Table 4. Error indices for the ICBI method and the proposed method are the lowest with the proposed method showing superior results. ANR and bicubic have the highest EI, whereas NEDI provides EI that is better than ANR and Bicubic.

4.2. Objective Evaluation. The proposed algorithm outperformed conventional algorithms based on not only subjective evaluations but also objective evaluations. Table 5 shows the peak signal to noise ratio (PSNR) comparison of all algorithms for 20 images. On average, the proposed algorithm provides a PSNR that is 2.5181 dB higher than bicubic, 0.959 dB higher than NEDI, 0.594 dB higher than ICBI, and 9.13 dB higher than ANR.

Table 6: Elapsed CPU time (sec) comparison of all algorithms.

Image	Bicubic	NEDI	ICBI	ANR	Proposed
Airplane	0.0449	23.654	7.1067	3.2999	2.063
Banana	0.0042	21.548	6.1403	2.8692	2.2415
Barbara	0.0052	37.441	10.76	4.5643	5.2492
Brush	0.0052	25.22	7.4608	3.373	2.4569
Beehive	0.0037	24.726	6.8956	3.2409	2.3253
Bridge	0.0038	23.011	6.7087	3.0958	3.5507
Brief case	0.0037	21.374	6.2375	2.8979	2.1482
Cofee maker	0.004	27.333	7.8577	3.5475	2.4451
Elephant	0.0031	20.428	5.9387	2.8511	2.3893
Feather	0.0037	23.063	6.6781	3.1025	2.9475
Golf cart	0.0038	24.581	7.1438	3.2469	2.6199
Grater	0.0034	19.554	5.6954	2.738	1.7863
Lena	0.0036	23.046	6.7263	3.0986	2.3076
Mail box	0.0041	22.974	6.6943	3.1225	3.1428
Pond	0.0035	22.799	6.6565	3.0779	3.0704
Shopping cart	0.0036	22.479	6.6835	3.1652	2.6862
Tire	0.0042	22.555	6.5826	3.0227	2.0995
Trafficone	0.0036	22.765	6.6323	3.0833	2.8033
Trash can	0.003	20.275	5.9118	2.7633	1.9244
Woman	0.0037	22.635	6.6033	3.0823	2.4294
Average	0.00588	23.572	6.8557	3.1621	2.6343

Table 6 shows the CPU elapsed time for all algorithms. CPU elapsed time for bicubic is the lowest compared to other methods because it is a very basic interpolation method and the function interp2 interpolates an image using the mex compiler of MATLAB. On average, the proposed method is almost 9 times faster than NEDI, 2.61 times faster than ICBI, and 1.2 times faster than ANR.

5. Conclusions

The preservation of edges is considered to be the most important parameter in the field of image interpolation. Linear interpolation methods produce blurring effects while complex algorithms result in oversmoothed edges or add artifacts to the image. These artifacts are a direct consequence of post-processing conducted on the interpolated images to either enhance or refine edges. In this paper, a gradient-based edge-preserving interpolation method is proposed. The proposed method uses a combination of low-complexity and efficient hysteresis thresholding based edge detection, the isolation of edges and transitional regions, simple edge map interpolation, direction estimation using the edge map and gradient map, followed by gradient and distance coefficient based interpolation. Simulation results show that the proposed method provides better performance with regard to both subjective and objective evaluations, while accurately preserving edges and image details when compared to leading alternative methods.

Abbreviations

ICBI: Iterative curvature-based interpolation
ANR: Anchored neighborhood regression
NEDI: New edge-directed interpolation
EI: Error index
PSNR: Peak signal to noise ratio.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. Khan and D. Lee, "Efficient deinterlacing method using simple edge slope tracing," *Optical Engineering*, vol. 54, no. 10, Article ID 103108, 2015.

[2] S. Ng, "A brief history of entertainment technologies," *Proceedings of the IEEE*, vol. 100, pp. 1386–1390, 2012.

[3] A. Amanatiadis and I. Andreadis, "A survey on evaluation methods for image interpolation," *Measurement Science and Technology*, vol. 20, no. 10, pp. 316–324, 2009.

[4] R. W. Kubey, "Television and aging: past, present, and future," *The Gerontologist*, vol. 20, no. 1, pp. 998–1004, 1980.

[5] T. M. Lehmann, C. Gonner, and K. Spitzer, "Survey: interpolation methods in medical image processing," *IEEE Transactions on Medical Imaging*, vol. 18, no. 11, pp. 1049–1075, 1999.

[6] R. C. Gonzalez, *Digital Image Processing*, Prentice-Hall, Upper Saddle River, NJ, USA, 2009.

[7] B. Jin, J. G. Kuk, and N. I. Cho, "A gradient guided deinterlacing algorithm," in *Proceedings of the 2012 19th IEEE International Conference on Image Processing*, pp. 853–856, Orlando, FL, USA, October 2012.

[8] D. H. Lee, "A simple, high performance edge-adaptive deinterlacing algorithm with very low complexity," in *Proceedings of the 2012 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 636-637, Las Vegas, NV, USA, January 2012.

[9] X. Chen, G. Jeon, and J. Jeong, "Filter switching interpolation method for deinterlacing," *Optical Engineering*, vol. 51, no. 10, 2012.

[10] J. H. Paik, K. P. Hong, J. K. Paik, and J. H. Hwang, "Image sequence interpolation for improving the resolution of the magnified image," in *Proceedings of the APCCAS'96-Asia Pacific Conference on Circuits and Systems*, pp. 544–547, Seoul, Republic of Korea, November 1996.

[11] J. Allebach and P. W. Wong, "Edge-directed interpolation," in *Proceedings of the 3rd IEEE International Conference on Image Processing*, pp. 707–710, Lausanne, Switzerland, September 1996.

[12] W. Dong, L. Zhang, R. Lukac, and G. Shi, "Sparse representation based image interpolation with nonlocal autoregressive modeling," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1382–1394, 2013.

[13] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[14] J. Shi and Z. Shan, "Image interpolation using a variation-based approach," in *Proceedings of the 2011 8th International Conference on Information, Communications & Signal Processing*, pp. 1–4, Singapore, December 2011.

[15] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, vol. 10, no. 10, pp. 1521–1527, 2001.

[16] N. Asuni and A. Giachetti, "Accuracy improvements and artifacts removal in edge based image interpolation," *International Conference on Computer Vision Theory and Applications*, vol. 8, pp. 352–355, 2008.
[17] A. Giachetti and N. Asuni, "Real-time artifact-free image upscaling," *IEEE Transactions on Image Processing*, vol. 20, no. 10, pp. 2760–2768, 2011.
[18] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1920–1927, Sydney, Australia, December 2013.
[19] E. R. Hancock and J. Kittler, "Adaptive estimation of hysteresis thresholds," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 196–201, Maui, HI, USA, June 1991.
[20] R. Medina-Carnicer, A. Carmona-Poyato, R. Muñoz-Salinas, and F. J. Madrid-Cuevas, "Determining hysteresis thresholds for edge detection by combining the advantages and disadvantages of thresholding methods," *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 165–173, 2009.
[21] R. Medina-Carnicer, F. J. Madrid-Cuevas, A. Carmona-Poyato, and R. Muñoz-Salinas, "On candidates selection for hysteresis thresholds in edge detection," *Pattern Recognition*, vol. 42, no. 7, pp. 1284–1296, 2009.
[22] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
[23] W. Rong, Z. Li, W. Zhang, and L. Sun, "An improved CANNY edge detection algorithm," in *Proceedings of the IEEE International Conference on Mechatronics and Automation*, pp. 577–582, Tianjin, China, August 2014.
[24] Y. Yu, Z. Li, B. Liu, and X. Liu, *An Adaptive Unimodal and Hysteresis Thresholding Method, Bio-Inspired Computing-Theories and Applications*, Springer, Berlin, Germany, 2014.
[25] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
[26] P. L. Rosin, "Unimodal thresholding," *Pattern Recognition*, vol. 34, no. 11, pp. 2083–2096, 2001.
[27] W. Burger and M. J. Burge, *Digital Image Processing: An Algorithmic Introduction Using Java*, Springer, Berlin, Germany, 2016.
[28] Computer Vision and Pattern Recognition Group, http://marathon.csee.usf.edu/edge/edge_detection.html, 2019.

Hindawi
Scientific Programming
Volume 2020, Article ID 6349342, 14 pages
https://doi.org/10.1155/2020/6349342

Research Article

Partial Observer Decision Process Model for Crane-Robot Action

Asif Khan,[1] **Jian Ping Li**,[1] **Amin ul Haq**,[1] **Shah Nazir**,[2] **Naeem Ahmad**,[3] **Naushad Varish**,[4] **Asad Malik**,[5] **and Sarosh H. Patel**[6]

[1]*School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu 611731, China*
[2]*Department of Computer Science, University of Sawabi, Swabi, KPK, Pakistan*
[3]*School of Computer Applications, Madanapalle Institute of Technology and Science, Madanapalle, India*
[4]*Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, KL University, Guntur, India*
[5]*School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China*
[6]*Interdisciplinary Robotics, Intelligent Sensing & Control (RISC) Lab, Department of Computer Science & Engineering, School of Engineering University of Bridgeport, Bridgeport, CT, USA*

Correspondence should be addressed to Asif Khan; asifkhan@uestc.edu.cn and Jian Ping Li; jpli2222@uestc.edu.cn

Received 1 December 2019; Accepted 27 December 2019; Published 28 February 2020

Guest Editor: Rahman Ali

The most common use of robots is to effectively decrease the human's effort with desirable output. In the human-robot interaction, it is essential for both parties to predict subsequent actions based on their present actions so as to well complete the cooperative work. A lot of effort has been devoted in order to attain cooperative work between human and robot precisely. In case of decision making , it is observed from the previous studies that short-term or midterm forecasting have long time horizon to adjust and react. To address this problem, we suggested a new vision-based interaction model. The suggested model reduces the error amplification problem by applying the prior inputs through their features, which are repossessed by a deep belief network (DBN) though Boltzmann machine (BM) mechanism. Additionally, we present a mechanism to decide the possible outcome (accept or reject). The said mechanism evaluates the model on several datasets. Hence, the systems would be able to capture the related information using the motion of the objects. And it updates this information for verification, tracking, acquisition, and extractions of images in order to adapt the situation. Furthermore, we have suggested an intelligent purifier filter (IPF) and learning algorithm based on vision theories in order to make the proposed approach stronger. Experiments show the higher performance of the proposed model compared to the state-of-the-art methods.

1. Introduction

Environmental perception and object recognition is an important part of the image processing. It can be widely used in robot visual perception, video surveillance, exception handling, intelligent early warning and rapid retrieval and efficient image storage, camera, and other fields. Humans can perceive the complex scenes easily and respond to get the location and type of the target object correctly, but currently this is a challenging problem for robot visual understanding.

Human eyes also have best capability to capture where neurons help in filtering the scenes. Human motion forecasting is the ability to predict subsequent motion series according to a given sequence of motions. By observing the motion behavior of the target object, the motion features are extracted and then motion forecasting is finally realized. Until now, processing of software as pirated or not pirated still becomes a challenging task [1]. Human beings can realize such forecasting through observation, which embodies the more intelligent reasoning ability of human beings (Figure 1). In some sensitive scenarios where the object is completely unknown (encrypted) from the observer and that object has to be identified in the encrypted form, ideas need to be added in the future work to improve the observer's capability in the encrypted domain [2, 3].

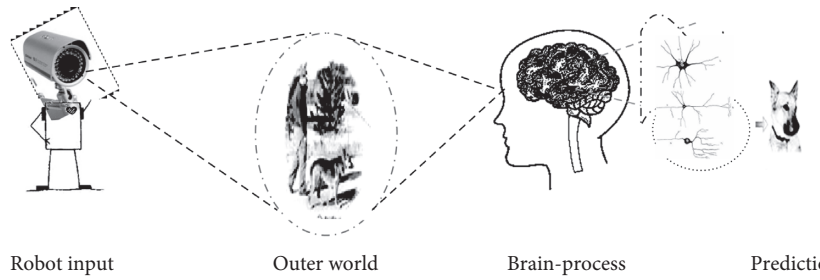


Figure 1: Robot Vs human real-world observation.

The human-robot or robot-robot interactions ability becomes particularly crucial for observing the motion of the object. In this context, an object may vary in orientation and scale or may even be partially obstructed. But, this is not always impairing our ability to recognize it. Due to the lack of this ideal platform, judging a complex scene, obtaining the location, and targeting the object accurately are complex tasks for machines or robots as compared with humans. Computer-assisted predictive system plays an important role to assist any observer recognition [4]. To assist any observer, selected features might be redundant variables which must be handled [5]. Selecting the most appropriate components is crucial for the success of the entire machine. However, decisions regarding software component reusability are often made in an ad hoc manner, which ultimately results in schedule delay and lowers the entire quality system [6].

Recognition by the robot vision is a tough and challenging problem to predict a significant part of the complex, unstructured, and arbitrary scenes; it is also very difficult to balance and place the output of the algorithm and the effect of recognition for already known targets. Visual scenes interact with each other in various topography combinations, i.e., the arrangement of the physical characteristics of a region, and adaptive system design is difficult to enhance understanding of the impact of natural scenes in complex environments.

Therefore, the development of the natural environment, image processing, and computer vision is focused on visual perception and faces enormous challenges. The visual perception system is highly nonlinear dynamic system level neural information collection. For storage and intelligible process, visual attention structure plays an important role in the visual perception. Software for decision making like birthmark is a unique quality to detect software theft [7]. Local visually interpreted information and available computing resources are concentrated on the most essential evidence that makes the visual perception possible in real-time, which can be customized to the dynamic perception of the real world.

Every living thing has patterns of action as per their nature, but for machines there is need to program them to work accordingly. From our home to big industries, there is lot of application of robotics that can be found like vacuum cleaner, self-driving vehicles, and different types of industrial robots. In such types of robots, the working of brain and vision is very similar to human beings' brain and eye control.

A lot of effort is being made in the research and development sector throughout the world to find solutions for this problem [8]. Vision capability towards perception and sensing is a real phenomenon for understanding mobility and manipulation of real world's random situation. In many situations, it is difficult to get sufficient images of an object, which makes the object recognition and the identity authentication difficult. Small sample with high dimension problem is a hot project recently. In a conventional object database, the number of images is limited.

The objectives of this research are encouraged by the neural network cognitive intelligence and the adaptive nature scene recognition technology, which can enhance the understanding of natural scenes, targeting the object, and solving the diversity, randomness, complication of natural scenes and other problems, making the real-time visual system highly flexible. It is to provide a stable foundation for the practical application of mining. Natural complex environment with complex scenes diversification shown in Figure 2 demonstrated how to overcome the lack of randomness in the visual processing system.

For example, the Biological Vision Model (BVM) is devoted to providing a new technological approach on behalf of merging new cognitive visual futures with inspired nerve cells cognitive intelligence cortex, which try to relate with real-world object recognition. To perceive arbitrary natural scene from complex environment perception and sensing in robotic mobility and manipulation on unstructured random natural scene understanding is a challenging problem in visual imaging and processing [9].

Neural network is a map of "neuron like" nodes; in this paper, we are taking neural network (NN) as just an example, committed to making a contribution a new technical concept for scene comprehension and acknowledgment by reorganizing new visual intellectual characteristics into scene expression, which can be very essential and provide robot vision with perceptual intelligence. This approach not only let the system proceed but also provide learning in natural scene with complex environmental perception and understanding. Through the study of perception ability of the natural scene image from complex environment, robot vision is enhanced with the integration of cognitive visual feature and the scene expression [10].

Our contributions summarized as follows:

(i) We enhance the efficiency of capturing, representing the target features of visual images and improving the features representation of the natural environment, so that the system can intelligently observe the unorganized nature scenes.

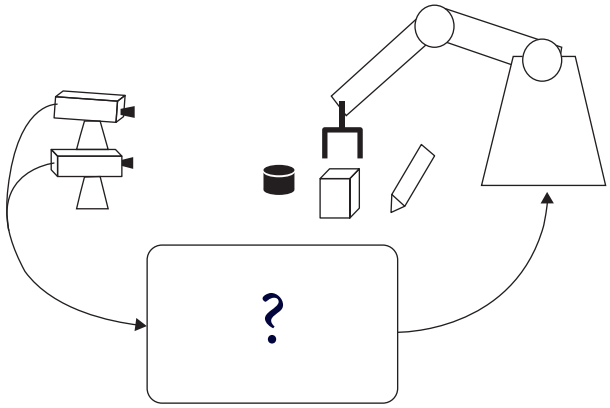


Figure 2: Real-time visual system.

(ii) We proposed a model that can go through essential, generally measured capacity skill for intelligent approach of vision-based information retrieval system, analyzing and refining as a breakthrough to provide better intelligence to the visual information.

(iii) Our proposed model inherits a new intelligent purifier filter processing scheme, that is, upgradation of bio-inspired image processing.

(iv) The proposed model is essentially inspired by complex BM (Boltzmann Machine) mechanism that is scene prediction for visual information processing, which is expert in obtaining better perception for decision performance with deep belief network. We provide considerable empirical observations on the chosen datasets to support obtained results.

The remainder of this paper is organized as follows. In Section 2, we outline the related works with cause of motivation for our work. Section 3 covers the proposed partial observer decision process model which is further described with two subsections: first is to obtain the possible perception for making next step decision with deep belief network, and second is to learn decision for further action by its filter analysis that is included with learning algorithm. In Section 4, the remarkable performance is demonstrated by the experimental simulation and their outcomes. Finally, in Section 5, we conclude our proposed analysis with future accepts.

2. Related Work

Studies have shown that [11] the factors affecting visual attention from two aspects, i.e., top-down prior knowledge and input signal, make the sensor stimulus from bottom to up. Among them, the top-down prior knowledge and applications are highly correlated, which is very tough for modeling analysis. Therefore, there are lot of sensor stimulations only for the Bottom-up visual attention model. The paradigm of bottom-up visual attention can be classified into two categories [12]. One is to use the eye tracker eye to glance at the image's location and use statistical methods to make the eye zone appear longer and as a significant area of human interest. Another category is defined by multichannel multiscale analysis of the input image, statistically significant interest on the extent of each pixel in image depending on the extraction distribution.

First visual attention model based on significant distribution map had been proposed by Koch and Ullman [13]. Before that, there have been many visual attention models based on significant distribution in [13–17]. But there is no model work with a compatible system for complex domain knowledge like human eyes. The same does not involve the human eye gaze input image and gaze time which statically show the number of repetitive tests that are pleasant to human beings like us. There is many field space where target detection [11, 18, 19], video compression and coding [20, 21], image analysis [22, 23] and scene understanding [24]. These models will be applied. And other fields can use the limited memory computing resources to process the input video image or the region of most interest of human vision.

Therefore, without decreasing the intervention efficiency of the concept, the system not only reduces the overhead space but also increases system performance in many ways, such as the effect of processing individual vision needs is more, a stronger noise robustness has increased stability in complex backgrounds, and so on [11, 18, 19, 22–25]. These models, moreover, need to calculate multiscale and multichannel features of the Gaussian pyramid input image and calculate these considerable sample dividend payments into a globally significant distribution, using the Winner-Take-All (WTA) mechanism to select the most significant area independently [24].

The entire process requires a large number of intermediate results that can be stored and has a larger amount of computation, making it even more difficult to implement the limited computing resources in embedded systems. Biological science experiments confirmed that in the primate temporal cortex of the brain, nerve cell activity and animal identification of objects are closely linked [25]. When contrasting with the generic image model stored in the brain, the reorganization of the particular object can be understood. The researchers therefore conclude that a viable approach is to simulate the visual cortex structure in order to construct the object recognition.

Related to the earliest primate visual system model is the neocognitron model [26], which is based on self-organizing feed-forward neural networks. British Wallis and Rolls Experimental Psychology Department of the University of Oxford promoted the constant target identification VisNet primate model [27] and the improved version called Visnet2 [28]. It is a four-level feed-forward, convergence, and competitive nature of the network, where each layer brings together the former cell layer in a small portion of the input field (called filters). With this aggregation law, primate visual cortex cells increase the size of the receptive field characteristics by simulating the advance from junior to senior level. Mel made the SEEMORE model in 1997, and it is also a feed-forward hierarchical structure model which uses the color, shape, and texture combination to achieve visual object recognition. SEEMORE used multiclass combination of features to improve the recognition robustness. Serre et al. in 2005 and 2007 applied HMAX model to object

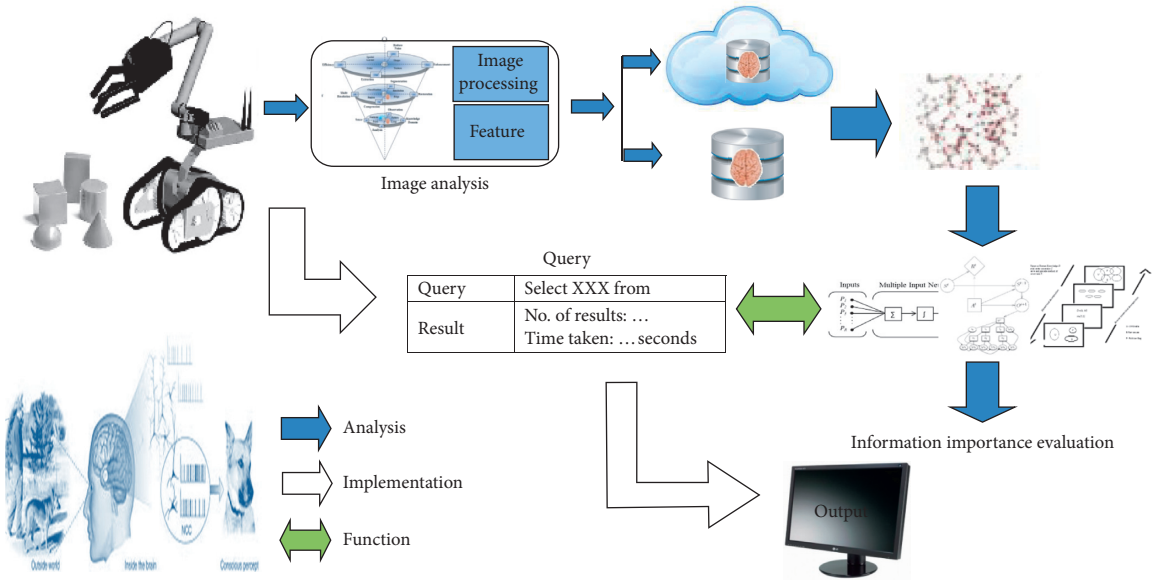


Figure 3: Observer prediction and decision process.

recognition; the improved model constructed high-level simulation of biological visual features. Visual features of the hierarchy template is a operation of matching and merging along the object recognition, where continuous simulation is used for invariant scale, translation, and rotation in the visual cortex. A lot of researchers have made outstanding contributions in this field.

Our proposed model has two aspects. (a) It provides an intelligent platform for integration of features and preprocesses to predict future prediction. For this we analyze Boltzmann machine mechanism [29], whose outcomes go through the second phase of purifier intelligent filter, which is inspired by the biological vision model to purify, segment, and identify the object, which makes the proposed model simple and efficient. (b) Second aspect covers the decision-based model that is incorporated based on accurate perception results, and the partners can cooperate with each other better. This requires the observer to possess the ability to identify and estimate motion sequences [30, 31].

3. Partial Observer Decision Process

To provide vision intelligence for action, the robot needs to go though the learning of the steps for task, while new associated algorithms are put forward to settle range of demanding theoretical problems in visual information processing system. To explore the inherent characteristics that provide new visibility for perception, such as diversity, randomness, and complexity in real-time complex natural environment, where adapting Network perception ability of the natural scene image is improved with the combination of cognitive visual features and scene expression.

The perception hierarchical model outcomes directly incorporate and participate for sensing object decision model. Action model can concurrently remember more than one objective, not only for the common goal of better classification, but also on the texture, non-rigid targets classification. This model is based mainly on visual computing simulation to calculate a cortical action (set of task) network hierarchy. The process model of observation can be easily understood by Figure 3.

To predict information C about dynamic object in complex environment by using visual information and predict about, typical approach form input $(A, \widehat{X})$/output $(B, \widehat{Y})$ relationships as follows,

$$f_0\left(C_{xy}\right) = \begin{cases} A, & \text{for} \quad f_i(\widehat{x}, \widehat{y}) \le t, \\ B, & \text{for} \quad f_i(\widehat{x}, \widehat{y}) > t, \end{cases} \tag{1}$$

where t is the threshold value and f is input/output image functions, respectively. It is ample to implement apparent contrast target and its background. The vision source included in the system makes a complex environment captured to be processed. During this process, the visual feedback is constantly followed to see the template matching for each frame of object's information and forecasting the position extraction dynamically. When the sum squared error between two objects is less than the captured image and BM results in predetermined threshold, then we can say that the object has found the one we were looking for [32].

3.1. Perception for Decision. Boltzmann machine (BM) mechanism is a variant of forecasting outcomes for sensing and perception. BM is a nonlinear generative model for time series that uses an undirected model with binary latent variables, h, connected to a collection of visible variables, v. At each time step t, v and h receive directed connections from the visible variables at the last N time steps, where N is the size of the temporal window considered. The "history" vector or knowledge datasets are concatenated by the data at $t-1, t-2, \ldots, t-N$, which we call $v_{<t}$. In MB, the model

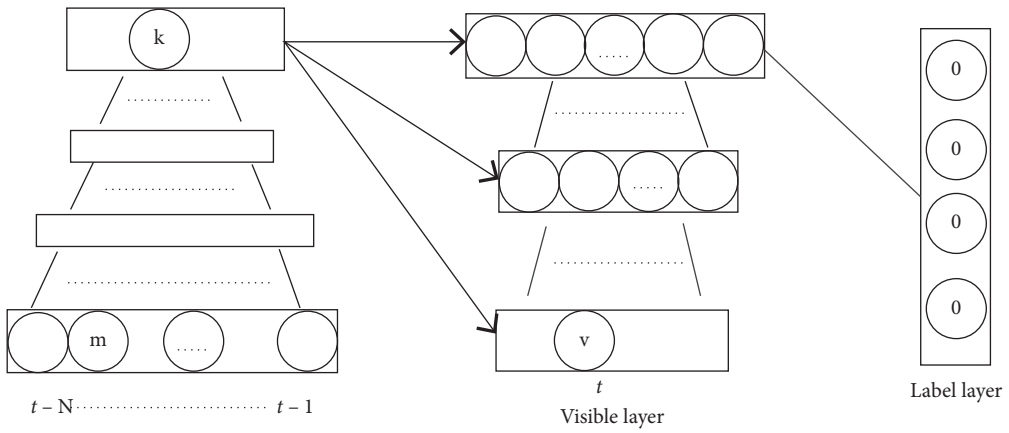


Figure 4: Deep belief network.

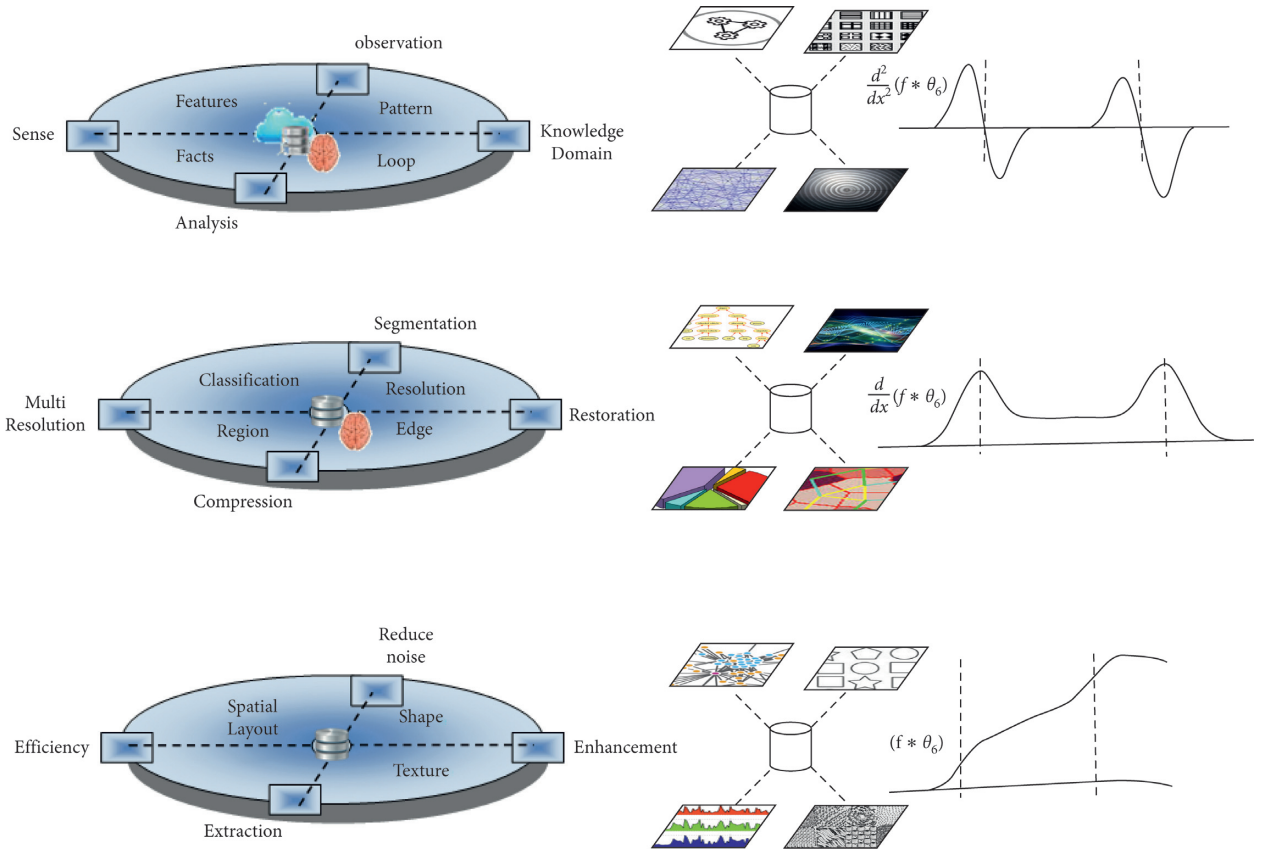


Figure 5: Primely integrated intelligent purifier filter.

defines a joint probability distribution over v_t and h_t, as shown in equation (3) conditioned on $t \geq v$:

$$Z(v_t, h_t \mid v_{<t}) = -\sum_j v_{j,t}\hat{X}_{j,t} - \sum_i h_{i,t}\hat{Y}_{i,t} - \rho, \tag{2}$$

where

$$\begin{aligned} \hat{X}_{j,t} &= bv_j + \sum_k X_{kj}v_{k,<t}, \\ \hat{Y}_{i,t} &= bh_i + \sum_k Y_{ki}v_{k,<t}, \end{aligned} \tag{3}$$

where

$$P(v_t, h_t \mid v_{<t}, \theta) = \frac{\exp\left(-Z(v_t, h_t \mid v_{<t}, \theta)\right)}{E(v_{<t})}, \tag{4}$$

where $E(v_{<t})$ is a constant called the partition function and $\hat{X}_t$ and $\hat{Y}_t$ are the dynamic biases on time t, which express the input from the past to the visible and hidden units of vision resource (equations (2) and (4)).

Boltzmann machine can also be classified as trained with a generative learning objective, where internal entity

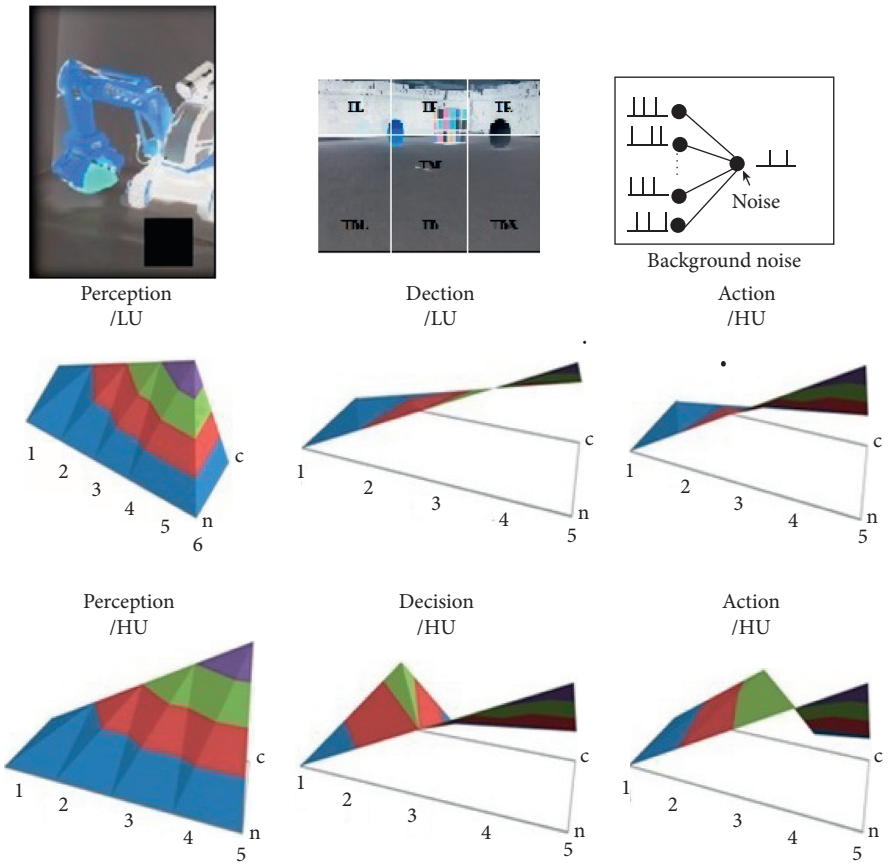


Figure 6: Observer decision process and its partial task analysis.

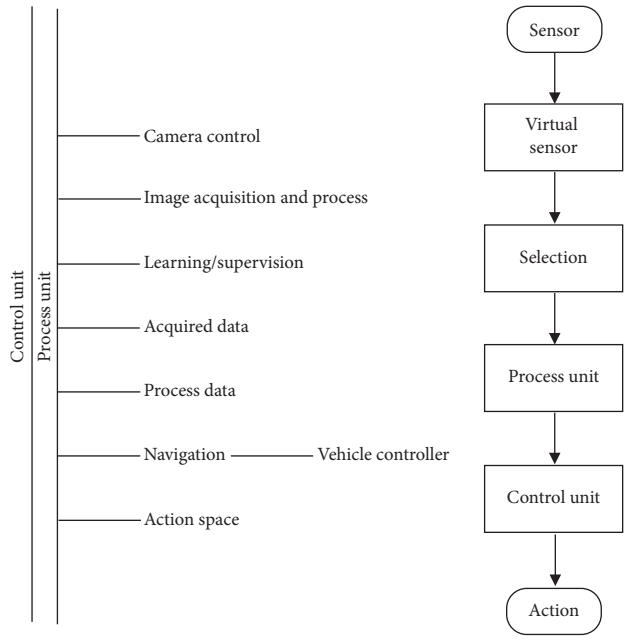


Figure 7: Perception and decision model for next action.

like camera follow this to track, motion, and control. For that learning of the joint distribution $p(v, y)$ of the input vector v and the target class y, and/or a discriminate learning objective and learning of the conditional distribution $p(y|v)$ directly are necessary. It requires no additional training phase for a classifier like the traditional Boltzmann Machine. The energy function of the BM is shown in the following equation:

Input: dynamic input coordinates x_t, y_t;
Hidden representation of time $t + 1$: X_{t+1};
Class label for the time t: Y_t;
Output: the prediction on time $t + 1$: x_{t+1};
(1) Step 1: initialization
(2) **do**
(3) Calculate the hidden representation on time $t + 1$:
(4) Calculate the visible representation on time $t + 1$:
(5) Calculate the class label of the prediction on time $t + 1$:
(6) **while** $Y_{t+1} == Y_t$;
(7) Adjust the initial input value x_t randomly.
(8) **return** Y_{t+1};

Algorithm 1: Observer parameter learning.

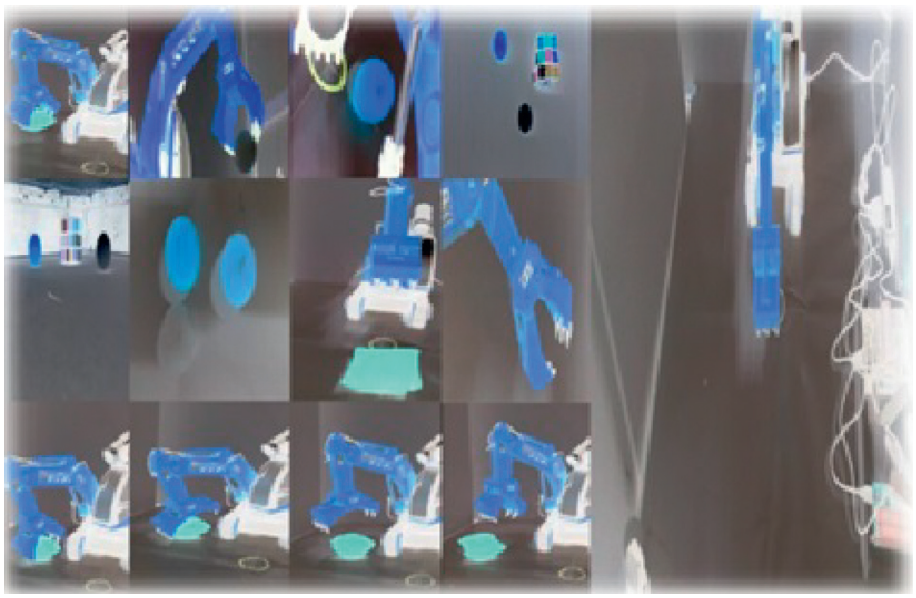


Figure 8: Experimental setup.

Table 1: Forward walk table.

Steps/count (10)	Steps/count (50)
8.5236	31.3214
8.0325	28.8976
7.3178	26.7849
7.4592	24.8934
8.1562	21.2123

$$Z(y, v, h) = -b^T v - c^T h - d^T e_y - h^T\left(Ue_y - Wv\right), \quad (5)$$

with parameter $\rho = (b, c, d, W, U)$ and where v is the input vector and y is the first process of the class label. To achieve the discriminate objective, the posterior probability in the BM can be inferred from the following equation:

$$P(y|v) = \frac{\exp\left(-Z(v, y)\right)}{\sum \exp\left(-Z(v, y^*)\right)}. \quad (6)$$

The denominator sums over all labels y^* to make $P(y - v)$ a probability distribution. BM can only do the classification task with the independent relation between samples except time series; for the time series, the samples between each other are dependent with each other and can be affected by previous and succeeding samples.

The main purpose of our model, which is primarily based on BM, is to suppress the error amplification problem and prolong the perception length. From the analysis in this section area, we can conclude that the problem that makes BM inefficient during perception primarily arises from two aspects: the first is the previous past result as input data directly and the second is that there is no constraint on the present result. In our work, we should avoid the past prediction result directly being the input data; meanwhile, we should also lower the perception ratio [9].

We retrieve the feature for decision using BN for the last N time steps and discriminate the class label of the

Forward walk table

Steps/count (10)	Steps/count (50)
8.5236	31.3214
8.0325	28.8976
7.3178	26.7849
7.4592	24.8934
8.1562	21.2123

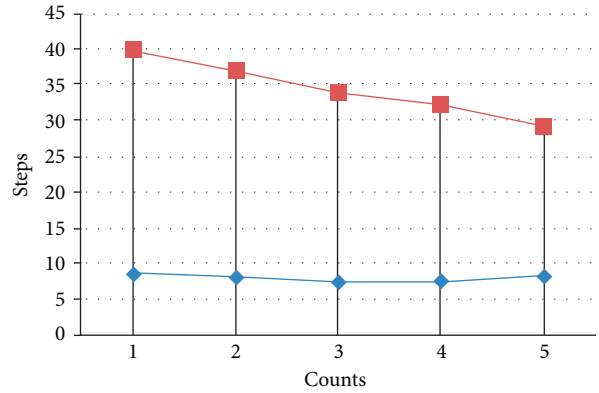


Figure 9: Ratio of walking (forward) table.

Table 2: Backward walk table.

Steps/count (10)	Steps/count (50)
8.8756	36.2543
8.9546	29.6435
7.1289	28.7543
8.1136	26.1235
8.0128	22.8456

forecasting result to decide whether to accept it accordingly. The structure of the model is shown in Figure 3.

3.1.1. Deep Belief Network. After we have trained the model, we can add layers as in a DBN (Figure 4). The previous steps are kept and connected to each hidden layer with an independent weight matrix. The next level will take the previous hidden state vector as the "observed or predicted" data. A two-level model is shown in Figure 4.

3.2. Decision for Action. Intelligent purifier filter (IPF), vision paradigm understanding, and interpretation totally depend on an intelligent relationship between processing of visual real world as inputs and processing output, which make the machine capable to see and understand. Primitive layer and comparative layer integrate all visual processing features and preprocesses by which analysis of image becomes easier. Nowadays different work environments also maintain a huge database of complex world's objects, so by this model, we try to analyze objects on behalf of their preprocess and features as in Figure 5. Models goal layer capable to perform an intelligence of identification, observation with their artificial approach, cause of this possibility second phase of this proposed model reduce complexity of vision for robots to predict accurate for next steps. Every layer consists of different unit type and uses the previous output as the input. (a) The first layer uses the fundamental scale image as the input, and the last layer output is the characteristic value which can be applied to class recognition. Along with the time, field's size increment and complexity become progressive receptively. (b) The complexity of the top visual area is simply built up by the lower-layer steps and has some redundancy. (c) In this proposed model, the purifying filter with the Gaussian pyramid based on the input dynamic objects or real world surrounding calculated brightness, color, direction, and multi-scale characteristics of the channel. It leads to a plenty of calculation and storage of the next process of random sampling. Some others replace across decision for the combinations and normalization with the local extremism method, iterative method, or a prior knowledge method.

DOG filter function can be described as

$$D_o G_s, I_c(I) = G_{\sigma(s)}(I - I_c)) - G_{s\cdot\sigma(s)}(I - I_c), \tag{7}$$

$$G_{\sigma(s)}(I) = \frac{1}{2\lambda \cdot \sigma(s)^2} \cdot e\left(\frac{1}{2 \cdot \sigma(s)}\right). \tag{8}$$

As per equation (8), it is clear that in 2D for Gaussian function with equation (8) variance $\sigma(\&)$ that depends on the scale as S_1 is the position for center position I_c of filter with photoreceptor.

Then, cell activation is computed as dot product as shown in the following equation:

Backward walk table

Steps/count (10)	Steps/count (50)
8.8756	36.2543
8.9546	29.6435
7.1289	28.7543
8.1136	26.1235
8.0128	22.8456

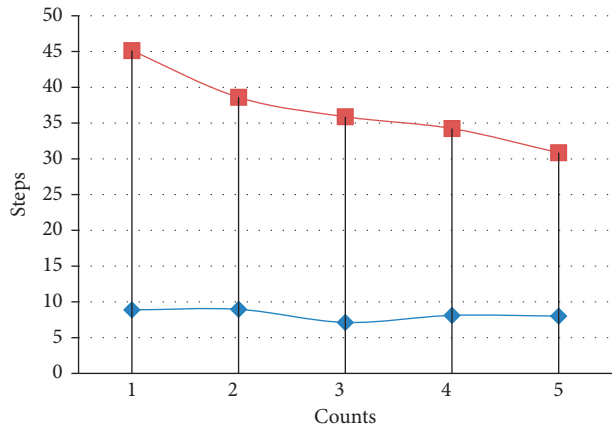


Figure 10: Ratio of walking (backward) table.

$$\langle I, \phi_1 \rangle = x = \sum_{i=R} I(l).\phi_1(l). \tag{9}$$

Here, ϕ_1 is the filter weight, I is the neuron from the respective region R for intensity $I(l)$. So, after the MAX operation, the response of a complex unit for C_1 is

$$r = \max\left(x_j\right), \quad j = 1, 2, 3, \ldots, m. \tag{10}$$

So, precisely timed action potential through intensity is expressed as

$$T_i = f(s_i) = T_{\max} - \log(\beta s_i + 1). \tag{11}$$

That show for one pixel S_1, with scaly factor β maximum time of esocdy windows is $T_{\max}$.

There exists each cell count with different oscillations (subthreshold membrane oscillations). So, it is described as

$$\mathrm{OSC}_i = (\cos(\omega T + \phi)). \tag{12}$$

The number of cycle W and initial phase for I pixel are

$$\phi_1 = \phi_0 + (i - 1) \cdot \mathrm{d}\phi. \tag{13}$$

After converting the intensity value, equation (13) into a line action, the algorithmic operation is implemented as all steps of performed relational information.

So, retrieval for learning perform correlation measure is adopted to measure the similar degree between desired (d) and actual o/p. So, matrix epochs are

$$C = \frac{\vec{V}_d \cdot \vec{V}_a}{\left|\vec{V}_d\right| \cdot \left|\vec{V}_a\right|}. \tag{14}$$

After information to learning, we make a decision by correlation C between the desired o/p and actual o/p, so target pattern considers as much closed to C.

The authors believe that due to probabilistic prediction and sensing technique, the ideal (Figure 6) solution provides basic means for extending the capacity of hardware system beyond the boundaries provided by currently used observation process methods (as shown in Figure 7). In our opinion, the application of introduced methods (in particular, the new DBN, purifier filter, and obtained dataset representation of Algorithm 1) leads to effective improvement outcomes, at least for the case of using the following.

4. Experimental Setup and Result

Following, motion, and control are essential for a settled verbalized robot like crane with a portable device like mobile sensor to arrange the objects accordingly with arms and onboard camera, with visible device that is able to position controllable world stage. So these genuine exploratory methodologies consolidate with an algorithmic coding of MATLAB [32] and that signal activity executed as control and process (Figure 8). The action-3D database is a type of motion-action behavior dataset. This dataset was captured by a strength camera. There are 23 types of behavior in the dataset, namely, move-backward, move-forward, move-jump, move-up, move-down, high arm, horizontal arm, hammer, hand catch, throw, draw, circle, hand two hand move, side move, back move, role, left side move, right side move serve, pick up, and throw. Each movement was repeated two times by 10 subjects; thus, there were 30 sequences of each action in the dataset, and there were 600

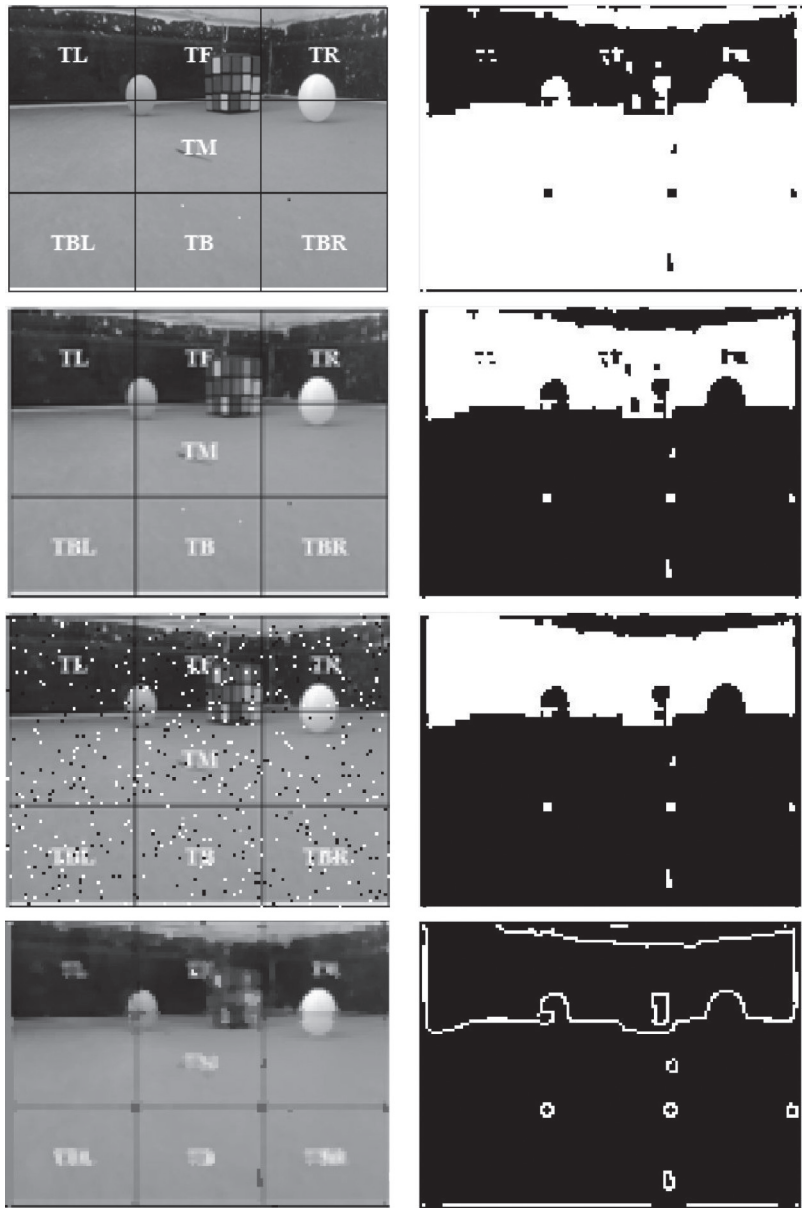


Figure 11: Partial observation analysis and outcomes after LR/HU.

Table 3: Pickup and through table.

Steps/count (10)	Steps/count (50)
9.0273	33.3423
8.4236	30.8745
8.2684	27.6715
7.9214	26.9452
7.1029	21.1579

sequences in total. The sampling frequency is 15 times per second, and the resolution of each frame is 640∗480.

4.1. Simulation Result. We validated the performance of our model on the dynamic observer dataset in IPF. The dynamic observer dataset is retrieved from a video camera. There are two categories in this dataset (as shown in Table 1 and Figure 9). One is "forward walking," where the performer points to somewhere with nothing in their hand, and the other is "backward walking," where the performer steps try to holds the object (as shown in Table 2 and Figure 10). There are a total of 200 time series in the data. We chose 150 series as training data, and the remainder are testing data. Each series contains 150 frames, and each frame is univariate. We represented the whole series in a matrix, and each row stands for a single motion. The preprocessing of data is a very necessary step for good representation of data and machine learning [33]. We show the curve graphs of the two types for output (Figure 11). The left one is the "forward walking" class, and the right one is the "backward walking" class. Then, we incorporate the whole system to grasp the target object and replace at particular destination in final action (as

Pickup and through table

Steps/count (10)	Steps/count (50)
9.0273	33.3423
8.4236	30.8745
8.2684	27.6715
7.9214	26.9452
7.1029	21.1579

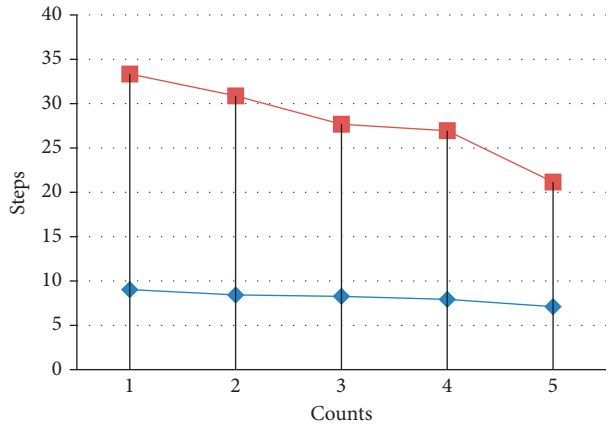


Figure 12: Ratio (pickup and through) table.

LR/HU table

Learning rate	Hidden unit
0.1	100
0.0712	0.0801
10.0618	11.0398
0.1322	0.5721
0.01	200
0.0765	0.0712
9.9589	9.0286
0.1159	0.4678

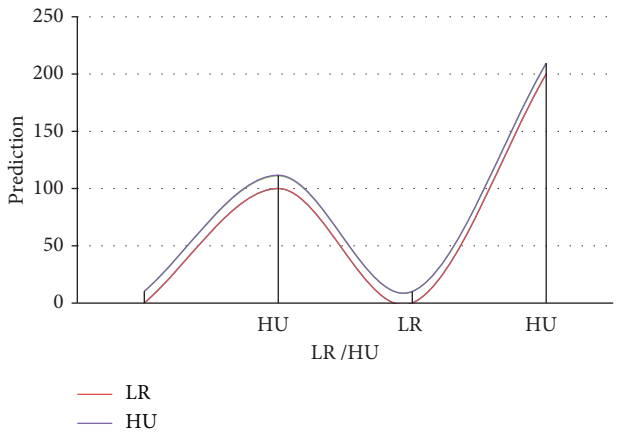


Figure 13: LR/HU for observer prediction.

shown in Table 3 and Figure 12). We took each single time series as a batch, which means that there were a total of 150 batches when training. We first verified the BM with a shallow structure. We show the results of different hidden unit numbers, different prestep numbers, and different learning rates in Figure 13 and Table 4. Because there is no theory on how to set the hidden unit number, prestep

Table 4: LR/HU table.

Learning rate	Hidden unit
0.1	100
0.0712	0.0801
10.0618	11.0398
0.1322	0.5721
0.01	**200**
0.0765	0.0712
9.9589	9.0286
0.1159	0.4678

number, and learning rate, we chose the root mean squared error (RMSE), mean absolute percent error (MAPE), and mean relative error (MRE) as the criteria. We found that when the prestep number is 5 and the learning rate is 0.01, after 200 iterations, the hidden unit number is 200, which can obtain relatively small value on RMSE, MAE, and MRE. In addition to the hidden unit number, we conducted several experiments with different prestep numbers, and we found that increasing the prestep number cannot improve the prediction performance much further.

The values in Table 4 show that when the number of hidden units is 200, relatively small values of the three criteria can be obtained. Therefore, in the following experiments, we set the number of hidden units as 200, the learning rate as 0.01, and the previous step number as 5.

4.2. Result Analysis. Because the original data sets are depth images that have high noise, the image is too vague and has other shortcomings; thus, this paper uses the real-time tracking algorithm to extract the image in 3D joint positions and finally combine the 3D dataset vector. Because the motion of the subjects in the dataset is actually 3D stereo motion, we transform the three-dimensional vector into a two-dimensional vector to express the original motion.

4.2.1. Comparison. Based on the results of the base work experiment (Figures 14 and 15) [32], and according to the proposed model outcomes as (in Figure 11, Table 4, and Figure 13) set the layer 2, the previous input step is 5, the hidden unit numbers are 200 for layer 1 and 100 for layer 2, and the learning rate is 0.01. We trained the model for 500 epochs. We divide the dataset into batches. Each batch contains 100 samples. The parameters are updated after each batch. To depict the affection of proposed model, we randomly chose one sequence from the forward-move-action and input the first 5 frames into our model to generate the following 25 frames, hoping that the model can generate the remaining motions correctly. From analyzing the graph, the first predictions of these models are all very close to the targets.

5. Conclusion

This technique provides a simple and efficient approach for vision-based decision and action in comparison with the conventional or traditional one. However, the performance may be influenced by the limitations of the hardware such as

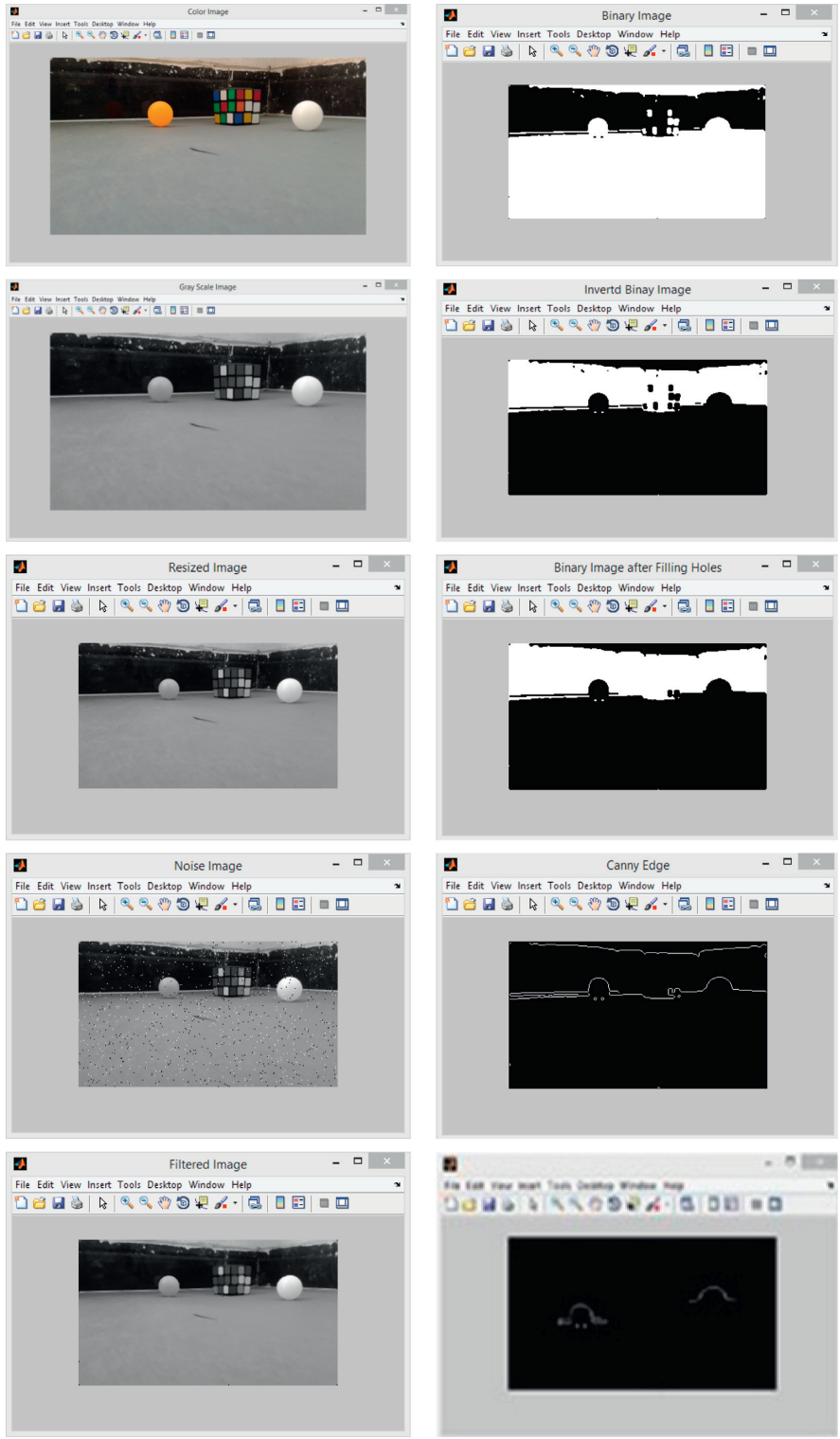


Figure 14: Partial observation analysis and outcomes.

model architecture and decision processing required. Acceptable empirical results have been obtained using the proposed strategy. As per obtained results, the number of preceding inputs and the contemporary decision output considerably have an effect on the performance. The next step in our research will be how to change the number of steps of previous inputs and how many units should be in the hidden layer to produce a high quality result. We will also continue to refine the algorithm to improve prediction and accuracy of practice. We are taking into consideration the speedy action processing and motion estimation as our future work.

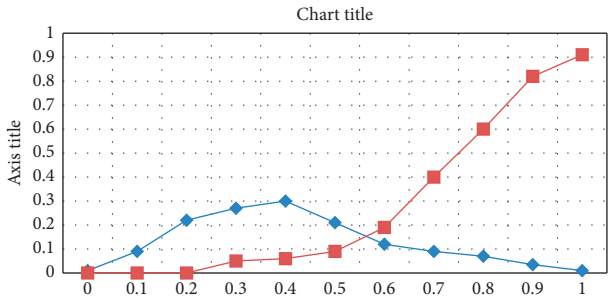


Figure 15: PDF outcome ratio.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant no. 61370073), the National High Technology Research and Development Program of China (grant no. 2007AA01Z423), and the project of the Science and Technology Department of Sichuan Province. All resources used for experiment support were given by Key Laboratory of Wavelet Active Media Technology School of Computer Science, University of Electronic Science and Technology of China (UESTC), No. 2006, Xiyuan Avenue, West Hi-Tech Zone, Chengdu, Sichuan, 611731, P. R. China.

References

[1] S. Nazir, S. Shahzad, and L. S. Riza, "Birthmark-based software classification using rough sets," *Arabian Journal for Science and Engineering*, vol. 42, no. 2, pp. 859–871, 2017.

[2] A. Malik, H. Wang, H. Wu, and S. M. Abdullahi, "Reversible data hiding with multiple data for multiple users in an encrypted image," *International Journal of Digital Crime and Forensics*, vol. 11, no. 1, pp. 46–61, 2019.

[3] A. Malik, H. Wang, T. Chen et al., "Reversible data hiding in homomorphically encrypted image using interpolation technique," *Journal of Information Security and Applications*, vol. 48, Article ID 102374, 2019.

[4] A. U. Haq, J. P. Li, M. H. Memon et al., "Feature selection based on L1-norm support vector machine and effective recognition system for Parkinson's disease using voice recordings," *IEEE Access*, vol. 7, pp. 37718–37734, 2019.

[5] A. Ul Haq, J. Li, M. H. Memon, J. Khan, and S. Ud Din, "A novel integrated diagnosis method for breast cancer detection," *Journal of Intelligent & Fuzzy Systems*, pp. 1–16, 2019.

[6] S. Nazir, S. Anwar, S. A. Khan et al., "Software component selection based on quality criteria using the analytic network process," *Abstract and Applied Analysis*, vol. 2014, Article ID 535970, 12 pages, 2014.

[7] S. Nazir, S. Shahzad, S. A. Khan, N. Binti Alias, and S. Anwar, "A novel rules based approach for estimating software birthmark," *The Scientific World Journal*, vol. 2015, Article ID 579390, 8 pages, 2015.

[8] S. Nazir, S. Shahzad, R. B. Atan, and H. Farman, "Estimation of software features based birthmark," *Cluster Computing*, vol. 21, no. 1, pp. 333–346, 2018.

[9] L. Xia, J. Lv, and D. Liu, "A motion classification model with improved robustness through deformation code integration," *Neural Computing and Applications*, vol. 31, no. 12, pp. 8519–8532, 2019.

[10] A. Khan, S. Deep, J.-P. Li, K. Kumar, R. A. Shaikh, and F. Hasan, "Vision prehension with cbir for cloud robo," in *Proceedings of the 2014 11th International Computer Conference on Wavelet Actiev Media Technology and Information Processing (ICCWAMTIP)*, pp. 293–296, IEEE, Chengdu, China, December 2014.

[11] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vision Research*, vol. 40, no. 10–12, pp. 1489–1506, 2000.

[12] O. Le Meur, P. Le Callet, D. Barba, and D. Thoreau, "A coherent computational approach to model bottom-up visual attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 802–817, 2006.

[13] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," in *Matters of intelligence*, pp. 115–141, Springer, Berlin, Germany, 1987.

[14] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[15] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, no. 9, pp. 1395–1407, 2006.

[16] T. Kohonen, "A computational model of visual attention," in *Proceedings of the International Joint Conference on Neural Networks, 2003*, vol. 4, pp. 3238–3243, IEEE, Portland, OR, USA, July 2003.

[17] K. Lee, H. Buxton, and J. Feng, "Cue-guided search: a computational model of selective attention," *IEEE Transactions on Neural Networks*, vol. 16, no. 4, pp. 910–924, 2005.

[18] J. Han, K. N. Ngan, M. Li, and H.-J. Zhang, "Unsupervised extraction of visual attention objects in color images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 1, pp. 141–145, 2006.

[19] Z. Chen, J. Han, and K. N. Ngan, "Dynamic bit allocation for multiple video object coding," *IEEE Transactions on Multimedia*, vol. 8, no. 6, pp. 1117–1124, 2006.

[20] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1304–1318, 2004.

[21] P. Zhang and R.-S. Wang, "Detecting salient regions based on location shift and extent trace," *Journal of Software*, vol. 15, no. 6, pp. 891–898, 2004.

[22] C. M. Privitera and L. W. Stark, "Algorithms for defining visual regions-of-interest: comparison with eye fixations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 9, pp. 970–982, 2000.

[23] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 300–312, 2007.

[24] P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.

[25] D. A. Leopold, I. V. Bondar, and M. A. Giese, "Norm-based face encoding by single neurons in the monkey inferotemporal cortex," *Nature*, vol. 442, no. 7102, pp. 572–575, 2006.

[26] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 411–426, 2007.

[27] K. K. Evans and A. Treisman, "Perception of objects in natural scenes: is it really attention free?" *Journal of Experimental Psychology: Human Perception and Performance*, vol. 31, no. 6, pp. 1476–1492, 2005.

[28] M. Carrasco, B. McElree, K. Denisova, and A. M. Giordano, "Speed of visual processing increases with eccentricity," *Nature Neuroscience*, vol. 6, no. 7, pp. 699-700, 2003.

[29] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[30] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, pp. 568–576, Montreal, Canada, December 2014.

[31] J. Donahue, L. Anne Hendricks, S. Guadarrama et al., "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2625–2634, Boston, MA, USA, June 2015.

[32] A. Khan, J.-P. Li, A. Malik, and M. Yusuf Khan, "Vision-based inceptive integration for robotic control," in *Soft Computing and Signal Processing*, pp. 95–105, Springer, Berlin, Germany, 2019.

[33] A. U. Haq, J. Li, M. H. Memon et al., "Comparative analysis of the classification performance of machine learning classifiers and deep neural network classifier for prediction of Parkinson disease," in *Proceedings of the 2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 101–106, IEEE, Chengdu, China, December 2018.